



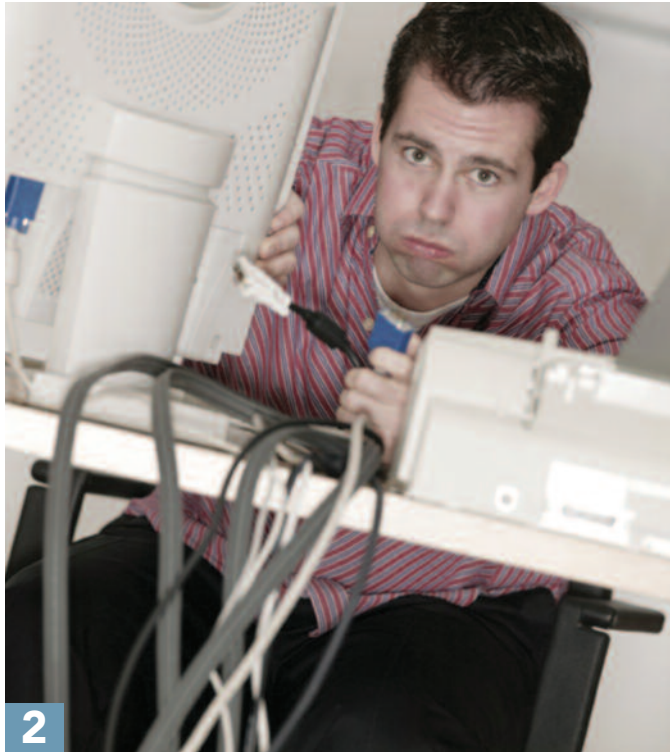
# The Do-It-Yourself Security Audit

10100101011011010010101101010110110010101001  
0100100110011001010100011100101010010  
0100101110010010010101001001001  
11101110010101001010101101010101  
10100101011011010010101101010110110  
01001001100110010101000111001010100100001010101

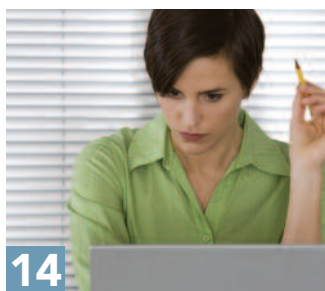
an **internet.com** Security eBook

# contents

## [ The Do-It-Yourself Security Audit ]



*Paul Rubens is an IT consultant based in Marlow, England, and has been writing about business technology for leading US and UK publications for almost 20 years.*



### **2** Introduction

### **4** Carrying Out Your Own Penetration Tests

### **7** Network Discovery: Scanning with Nmap

### **8** Sniffing Your Network with Wireshark

### **10** Checking Password Security with Hydra

### **12** Spotting Weak Passwords Using Offline Attacks

### **16** Checking Wireless Security with aircrack-ng

# The Do-It-Yourself Security Audit

By Paul Rubens

Keeping the servers, laptops and desktop PCs in your organization secure is a vital job, as a breach in security can lead to valuable data being destroyed or altered; confidential data being leaked; loss of customer confidence (leading to lost business); and the inability to use computing resources (and therefore lost productivity).

The cost of a serious security breach can be very high indeed, so most organizations devote significant resources to keeping malware and malicious hackers from getting on to the corporate network and getting access to data.

Typical defenses against these threats include:

- A firewall to separate the corporate network from the Internet
- An intrusion prevention/detection system (IPS/IDS) to detect when typical hacker activities, such as port scans, occur and to take steps to prevent them from successfully penetrating the network
- Malware scanners to prevent malicious software

getting on to the network hidden in e-mail, instant messaging or Web traffic

- The use of passwords to prevent unauthorized access to networks, computers, or data stored on them.



Jupiterimages

Every organization should have these defenses in place, but this leaves a very important question to be answered: How effective are these measures? It's a deceptively simple question, but it's essential that you know the answer to it. That's because if you don't it may turn out that:

- Holes in your firewall leave your network vulnerable
- Your IPS/IDS is not configured correctly and will not protect your network effectively
- The passwords used to protect your resources are not sufficiently strong to provide the protection you require
- Your IT infrastructure has other vulnerabilities you are not aware of, such as an unauthorized and insecure wireless access point, set up by an employee.

“

The cost of a serious security breach can be very high indeed, so most organizations devote significant resources to keeping malware and malicious hackers from getting on to the corporate network and getting access to data.

”

## The Do-It-Yourself Security Audit

### Penetration Testing

Penetration testing seeks to find out how effective the security measures you have in place to protect your corporate IT infrastructure really are by putting them to the test. It may involve a number of stages including:

- **Information gathering:** using Google and other resources to find out as much as possible about a company, its employees, their names, and so on
- **Port scanning:** to establish what machines are connected to a network and what services they have running that may be vulnerable to attack
- **Reconnaissance:** contacting particular servers that an organization may be running and getting information from them (like the usernames of employees, or the applications that are running on a server)
- **Network sniffing:** to find usernames and passwords as they travel over the network
- **Password attacks:** to decrypt passwords found in encrypted form, or to guess passwords to get access to computers or services

Defending a network and attacking a network are two different disciplines that require different mindsets, so it follows that the people best qualified to carry out a penetration test are not corporate security staff – who

are experts at defending a network – but hackers, who are experts at attacking them.

The best penetration tests involve using the services of "ethical hackers" who are engaged to attempt to break in to the network and discover as much information and get access to as many computers as possible.

A cheaper option is to use penetration-testing software, which searches for vulnerabilities, and in some

cases even carries out attacks automatically. A skilled human is more likely to be successful than any software tool, but using penetration-testing software to carry out your own penetration tests is still a good idea.

The software allows you to carry out these tests yourself on a monthly or even weekly basis, or whenever you make significant infrastructure changes, without

incurring the costs associated with repeated tests carried out by a consultant. If you use many of the free penetration testing tools that are available you will almost certainly be using the same ones that many hackers use as hacking tools. If you can successfully compromise your organization's security with these tools then so can hackers – even relatively unskilled hackers who know how to use the software. ■

A skilled human is more likely to be successful than any software tool, but using penetration-testing software to carry out your own penetration tests is still a good idea.



# Carrying Out Your Own Penetration Tests

**T**he more skills and knowledge you have, the more effective your penetration tests will be. A complete guide to penetration testing is beyond the scope of this eBook, but with some very basic hardware and free or low-cost software it's still possible to carry out some important checks to see how effective your security systems are. Any vulnerability you spot and correct raises the bar for anyone wanting to break in to your network and harm your organization.

## What You Will Need

### Hardware

To carry out your penetration tests you'll need a light, portable computer with wireless and Ethernet networking capability.

Although just about any reasonably new laptop will suffice, "netbooks" such as Acer's Aspire One or Asus' Eee PC make ideal penetration testing machines because they are extremely lightweight and portable, making it easy to carry around office buildings. Costing about \$350 they are inexpensive, yet powerful enough for the job, and they can run operating systems booted from a USB stick.

*Note: The instructions in this eBook have been tested with Acer's Aspire One but should work with the Eee PC or any other laptop with little or no modification.*

### Software

Most of the software needed is open-source and available free to download, compile, install, and run on

Linux. But by far the easiest way to get hold of all the software covered in this eBook (plus plenty more to experiment with) is by downloading a "live" Linux security distribution CD image and burning it on to a CD, or copying the contents on to a USB drive (since most netbooks lack an optical drive.) The benefit of a "live" distribution is that the entire operating system and all the software can be run from the removable media without the need for hard disk installation.

*Note: The instructions in this eBook assume that the reader is using a security Linux distribution called BackTrack 3, which can be downloaded from [www.remote-exploit.org/backtrack\\_download.html](http://www.remote-exploit.org/backtrack_download.html) and run from an CD or USB stick.*



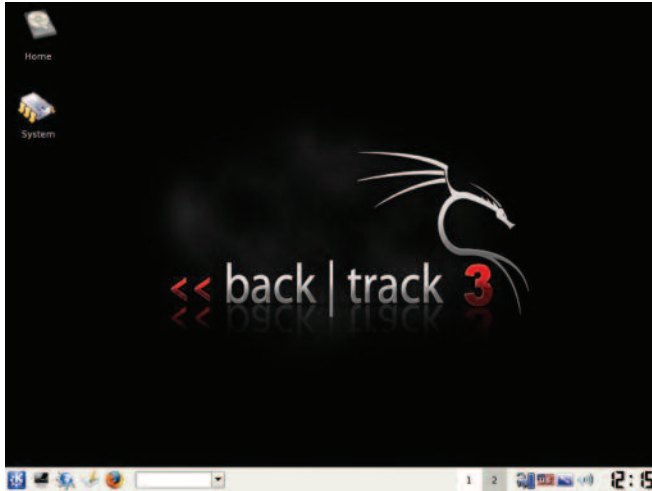
Jupiterimages

“

Although just about any reasonably new laptop will suffice, "netbooks" such as Acer's Aspire One or Asus' Eee PC make ideal penetration testing machines because they are extremely lightweight and portable, making it easy to carry around office buildings.

”

To start BackTrack3, simply insert the CD or USB into your penetration-testing machine, start it up, and boot from the removable media. Once the boot sequence is complete you will be greeted with the standard BackTrack 3 desktop:



*The BackTrack 3 desktop.*

### Automated Penetration Testing with db\_autopwn

db\_autopwn is an automated penetration testing tool that can test large numbers of Windows, Linux, and Unix computers on a network for vulnerabilities at the push of a few buttons. It is part of a suite of software popular with both penetration testers and hackers known as the Metasploit Framework.

To use db\_autopwn you first need to scan your network using a tool called Nmap to discover computers on the network and to establish which ports each of these has open.

Using this information, db-autopwn matches any known vulnerabilities in services that usually run on those ports with exploits in the Metasploit exploit library which use those vulnerabilities, and attacks the machines by running those exploits. If any of the servers on your network are successfully compromised (or "pwn"ed), you will be presented with a command shell giving you control over the compromised machine.

db\_autopwn has a number of benefits. First of all, it's free. It's also a popular tool with hackers. Using it will reveal if a hacker could easily compromise your network by using it. And if you do find that any of your computers can be compromised, it is easy to identify

## Creating a Backtrack 3 "Live" CD or USB Stick

To create a bootable BackTrack CD, download the BackTrack 3 CD image from [www.remote-exploit.org/backtrack\\_download.html](http://www.remote-exploit.org/backtrack_download.html) and burn it to a CD.

To create a bootable BackTrack 3 USB stick, follow these steps:

1. Download the extended USB version of Backtrack 3 from [http://www.remote-exploit.org/backtrack\\_download.html](http://www.remote-exploit.org/backtrack_download.html)
2. Open the downloaded .iso file using an application such as MagicIso or WinRAR (on Windows) or unrar (Linux).
3. Copy the "boot" and "bt3" folders on to a memory stick (minimum 1Gb)
4. Make the USB stick bootable.
  - In Windows, open a command prompt and navigate to the "boot" folder on your memory stick. If your memory stick is drive F:\ then type:

```
cd f:\boot
bootinst.bat
```

- In Linux, open a terminal window, and change directory to your memory stick, probably:

```
cd /media/disk
```

and execute the script bootinst.sh by typing:

```
bootinst.sh
```

the weakness, patch or update the relevant software, and then re-run the test to ensure the problem has been corrected.

On the other hand, db\_autopwn generally does not find vulnerabilities in services running on non-default ports (although hackers using the tool generally won't either). There is also the possibility that running the tool could

cause "collateral damage," i.e., you might crash servers on your network. A hacker running the tool would also do this, so arguably it is better to crash the machines

yourself when you are prepared for it than for a hacker to do so unannounced ■

### Automated Penetration Test Using db\_autopwn

1. Open a terminal window and move to the Metasploit Framework folder:

```
cd /pentest/exploits/framework3
```

2. Start Metasploit:

```
./msfconsole
```

3. Create a database to store the results of your Nmap scan:

```
load db_sqlite3
```

```
db_create Nmapresults
```

4. Scan your network and place the results in the database:

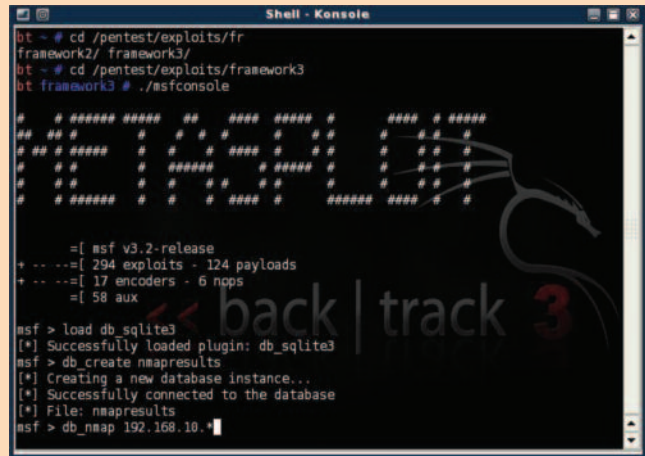
```
db_Nmap [target] (Replace the [target] string with the network block of your local subnet or the IP address of a target system that you want to test, e.g. 192.168.1.*)
```

5. Try to exploit the known vulnerabilities in any services running on the default ports on any of the machines:

```
db_autopwn -t -p -e
```

6. Once the auto\_pwn process is over, check to see if you managed to compromise any machines with the command:

```
sessions -l
```



Preparing to run db\_autopwn in BackTrack3

7. A numbered list of compromised computers will be displayed. To take control of one of these computers, type:

```
sessions -i 1 (replacing 1 with the number of the computer you want to control)
```

This will result in the command shell of the compromised computer, looking something like this:

```
[*] Starting interaction with 1...
```

```
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
C:\WINDOWS\system32>
```

## Network Discovery: Scanning with Nmap

**d**b\_autopwn is often used by relatively unskilled "script kiddies," and if it fails to find any vulnerable machines this doesn't mean that all the systems on the network are secure. That's because a skilled hacker may use other, more labor-intensive methods, plus knowledge and creativity, to try to find a way into machines on the targeted network.

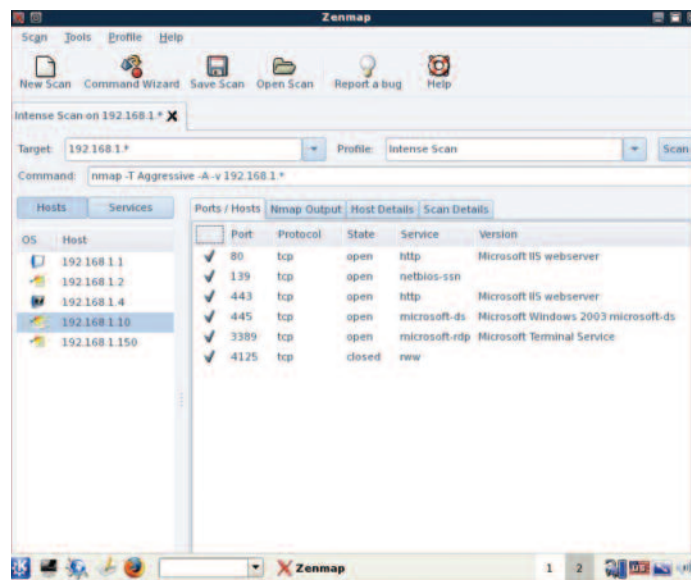
One of the first things an intruder is likely to do is scan the network to find out what machines are connected, and what ports they have open, possibly using Nmap, (the same scanner used to find machines to exploit using db\_autopwn.) Scanning your own network with this scanning tool can reveal what a hacker could discover, the devices connected to your network, and the ports they have open and the services they are (probably) running

This should alert you if unauthorized machines are attached to your network, or if any users are running unauthorized services. Nmap is a command line tool, but it can be operated more easily using a graphical front end such as Zenmap, which is included in BackTrack3.

### Scanning Your Network with ZeNmap

1. Start Zenmap by typing "zenmap" into the text box on the bottom panel on the BackTrack3 desktop.
2. Type in the network block of your local subnet or the IP address of a target system that you want to test in the Target box, e.g. 192.168.1.\*, choose a scan profile (or leave the default "intense scan") and click on scan.

After some minutes you'll be presented with the results:



**Zenmap displaying the results of a scan.**

On the left you can see a list of the hosts attached to the network and an icon representing the operating systems they are running. On the right is displayed a list of open ports and corresponding services on the host 192.168.1.10, a Windows Server 2003 machine.

In this example you can see that the server is running Windows IIS Web server, and also has port

3389 open for remote desktop sessions. Both of these have potential vulnerabilities, and present you with the opportunity to close these ports if these services are not required.

Zenmap is an extremely powerful scanning tool, and for complete instructions and usage example visit: <http://nmap.org/book/zenmap.html>. ■



# Sniffing Your Network with Wireshark

**N**map can give you a clear picture of the hosts connected to your network and which ports they are exposing, but it gives you no insight into the packets running over your network and the sensitive information these packets could reveal to an intruder. To discover this you need to make use of Wireshark (formerly known as Ethereal) an open source network protocol analyzer or packet sniffer. Many people describe using Wireshark as a revelation – the difference between getting a feel for the network they have responsibility for and turning on the lights and looking at what's going over it.

## Choosing a Point to Plug in to Your Network

Before using Wireshark it is vital to consider where you are going to plug your penetration-testing machine in to the network. That's because switches only send packets to ports leading to the destination machine, so if you plug your machine in to certain ports in your network infrastructure some packets won't reach your network interface card at all.

And some hubs (which should send traffic to all ports) are actually switched, so again you will miss out on some traffic.

But if you take time to understand your network topology and your hardware, you should be able to work out the best place (or places) to connect Wireshark to the

network to capture all the packets you are interested in.

To make things easier, some switches have a special monitoring port that replicates traffic to all other ports: plugging your penetration-testing machine into this port will enable you to see all traffic passing through that switch.

Why is Wireshark useful for a hacker?

- Sniffing a username and password pair provides the hacker with access to the user's e-mail box, which could contain sensitive or confidential corporate information
- Many organizations give users the same username for many different purposes, and many people use the same password for many different purposes. So gaining a username and password can help a hacker access other systems on your network, potentially causing far more damage than would be possible with access only to an e-mail account.

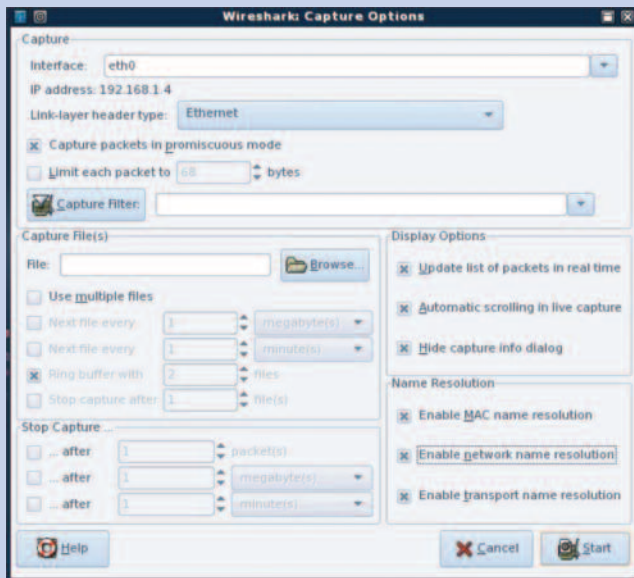


Wireshark can be put to a wide range of uses, including sniffing your network for traffic using protocols that have been banned for security reasons (such as MSN traffic.)

You can find a complete user guide at:  
[www.wireshark.org/download/docs/user-guide-a4.pdf](http://www.wireshark.org/download/docs/user-guide-a4.pdf)

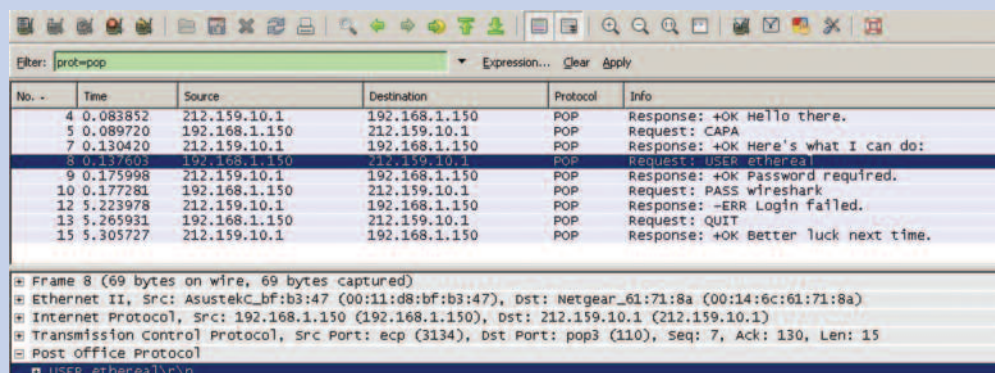
## Sniffing Your Network with Wireshark

1. Start Wireshark by typing "wireshark" into the text box on the bottom panel on the BackTrack3 desktop.
2. Click on "Capture – Interfaces ..." to select the network interface you want to use to monitor traffic, and then "Options" to set up the interface for traffic monitoring.

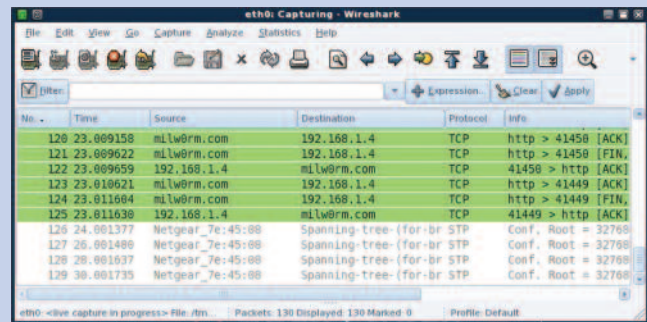


**Wireshark capture options.**

3. Check the "Capture packets in promiscuous mode" box to ensure your network interface captures and sniffs all packets on the network segment, not just those relating to your own network interface.



**Wireshark sniffing pop packets, revealing username ethereal and password wireshark.**



**Wireshark sniffing TCP packets containing a webpage from milw0rm.com.**

4. Click start to begin sniffing. The picture below shows Wireshark sniffing TCP traffic as segments of a page from the website at metasploit.com downloads.

One way that hackers can steal information is by sniffing passwords as they travel across the network. For example, they may sniff pop traffic to discover e-mail usernames and passwords, which are often unencrypted.

5. Type "pop" into Wireshark's filter text box (in some versions type "prot=pop"). Next time a user checks their e-mail on a pop server using an unencrypted connection, their username and password will be sniffed by Wireshark.

In this example a user has attempted to log in to a pop server with the username "ethereal" and password "wireshark".

# Checking Password Security with Hydra

There are many ways a hacker might get the e-mail usernames of people working at your organization. These range from simple techniques, such as looking at your company's Web page and searching for e-mail addresses to relatively difficult techniques such as sniffing your network traffic or interrogating your mail server (which could have been discovered during an Nmap scan.)

If your organization has an obvious e-mail username policy, such as firstname.lastname (john.smith) or intital-lastname (jsmith) then by searching your company Web

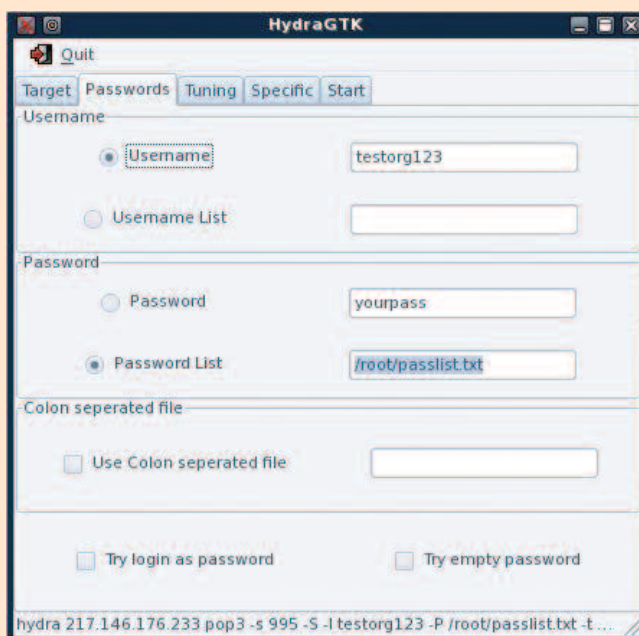
site or other material for the names of employees it would be very easy to compile a list of e-mail usernames.

Many people use the same username for many different uses, so if a hacker gets a hold of a few e-mail usernames they could well be valid on other servers such as ftp or smtp.

To discover the passwords that match these usernames on other servers, the hacker would probably carry out

## Carrying Out an Online Password Attack

1. Launch Hydra by typing `xhydra` in the text box on the BackTrack 3 desktop



Hydra

2. Choose a protocol to test from the Protocol dropdown box: Hydra can handle about 40 common protocols, including Pop3, telnet, ftp, VNC, SMTP, Cisco auth.

3. Choose a target – either the name or IP address of a single server, or a text file with a list of them.

4. Click on the Passwords tab, then either enter a single username to test – in this case `testorg123`, and specify a Password list that you want to test. This is a simple text file containing a list of possible passwords. You can either compile your own, or use Google to find and download lists of hundreds or even thousands of commonly used passwords. You can also add words that are relevant to your organization – the office name or current product or project names, for example.

5. Click on the Tuning tab and selecting the number of login attempts that are submitted simultaneously. This number is important to a hacker because the higher this number is set the higher the chance of being detected or locked out of the system are greater, but the faster the attack will proceed.

*continued*

## The Do-It-Yourself Security Audit

an online password attack. Essentially this involves attempting to log on to the relevant server using a likely username and a guessed password, and repeating this process with different passwords until a successful login occurs.

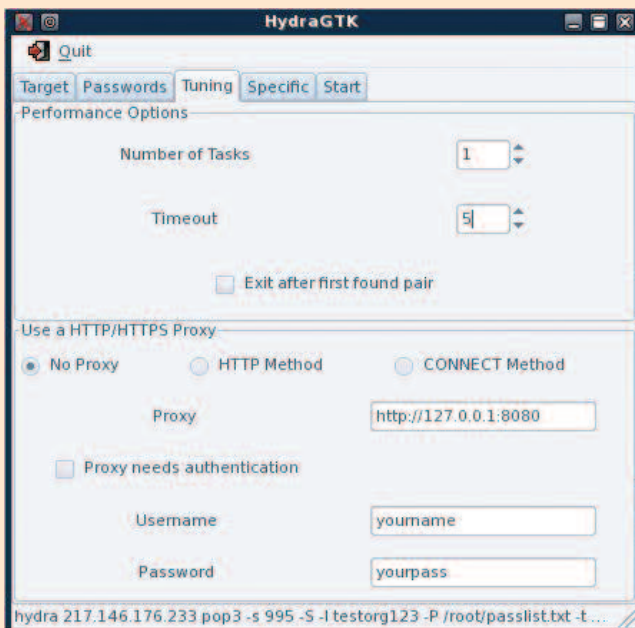
If the correct password is an "obvious" one like `qwerty` or `123456` then the hacker is likely to find it quickly – before the server spots that anything is amiss and prevents the hacker from submitting more login requests. (A well-configured server will limit the number of failed password attempts that are allowed before the account is suspended, the hacker's IP address is blocked or the period before a new login attempt can be made is extended. It should also log where failed attempts are coming from and alert administrators.)

The best way to check that your users have selected strong, difficult-to-guess passwords, and that your

servers are configured to spot a hacker submitting multiple guesses, is to carry out an online attack yourself, using the popular open-source password guessing tool called Hydra.

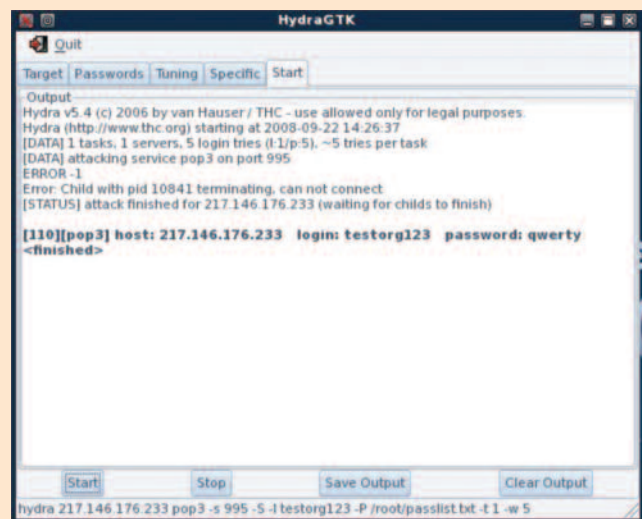
If a server is configured well, it should spot excessive failed login attempts and block Hydra before it manages to find any passwords (unless the hacker gets lucky in his first few guesses.) If you find you can work through a long list of passwords fairly quickly then it is well worth reconfiguring the security settings on your server to block access after fewer failed login attempts: legitimate users may misspell their passwords a couple of times, but there is no reason why anyone should be entering their password incorrectly more than a handful of times consecutively. ■

### Carrying Out an Online Password Attack *continued*



Hydra's Tuning tab.

6. Click the Start tab and click Start to launch the attack. In the example below, the correct password was found in a few moments – the user had picked the very simple password "qwerty." Working through a list of all your users and a long pass-



Hydra successfully guesses the password "qwerty" for the username "testorg123."

word list will reveal all of the users in your organization that have passwords which can be guessed by a hacker in a reasonable amount of time using Hydra – if the server hasn't been configured to prevent online password attacks.



## Spotting Weak Passwords Using Offline Attacks

When a user logs on to a server, he or she first has to submit their password. This password is passed through a hashing function, a mathematical process that converts it into a completely different string of characters, known as the password hash. The server consults a list that contains passwords hashes of all its users, and checks that the one it has received from the user matches the one in its password list.

The beauty of this system is that since hash functions are one-way (meaning that it is not possible to convert a password hash back to the original password) a hacker that gets access to the list of password hashes by breaking in to a server has no direct way to get at the passwords themselves: all they have is a list of password hashes, which have no instant value in themselves.

The only way to use the password hashes to get at the original passwords is by feeding different guesses into the hashing function and waiting until a password hash comes out that matches one of the hashes in the password list.

Since the hacker has the password hash list in their possession, there is no need to submit guesses to the server (using a tool like Hydra) to see if they are correct. Instead, they can run the whole process of passing guesses through a hashing function and comparing the results with the password hashes on the stolen password list on their own computer – a so-called "offline attack."

An offline attack is many times faster than an online attack, limited by the power of the computer carrying out the attack, not the server under attack. The server can't detect an offline attack itself, as it is being carried out on a completely unconnected system.

As well as using a list of guesses to try, it is also possible to attempt to "brute-force" the password hashes. This involves trying every combination of one, two, three, four (and so on) character passwords. Given enough time brute-force attacks are bound to be successful. A brute-force attack will find short passwords very quickly indeed, but a password made up of eight random characters could take hundreds of years to brute-force, and



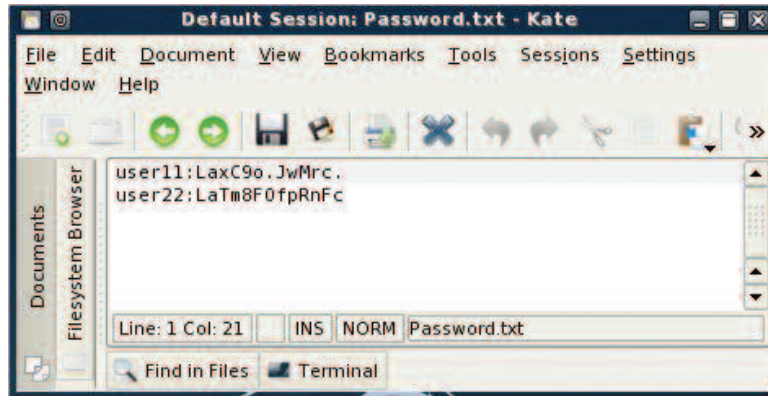
Jupiterimages

“

An offline attack is many times faster than an online attack, limited by the power of the computer carrying out the attack, not the server under attack.

”

## The Do-It-Yourself Security Audit



A password hash file with two usernames and their corresponding password hashes.

a nine character password could take thousands of years.

To test whether any of your users are using easily guessable or short passwords, you need an offline password-cracking tool like John the Ripper – known simply as John - which is included in BackTrack 3. Unlike many of the open source tools in BackTrack3, John has no built-in GUI, but fortunately it is very simple to use.

### Getting a Password Hash List from a Linux or Windows Server

From a Linux server:

1. Copy `/etc/passwd` and `/etc/shadow` from a server onto a memory stick, and transfer these files onto your penetration testing machine to the folder `/usr/local/john-1.7.2/`

2. Combine these two files into a file called `passwordlist.txt` using John's `unshadow` command:

```
cd /usr/local/john-1.7.2/  
unshadow /etc/passwd /etc/shadow > Password.txt
```

From a Windows server:

Since Windows protects the file in question, it is necessary to overcome this by booting your server into BackTrack3 using either a CD or USB. (Note: if this is not possible because they have been disabled then this means that a hacker trying to do this will also be thwarted.)

1. Once BackTrack 3 has booted, run `bkhive` on `SYSTEM` to get the system key:

`bkhive (path to)/SYSTEM systemkey.txt`  
An example of the path is:  
`mnt/sda1/Windows/System32/config`

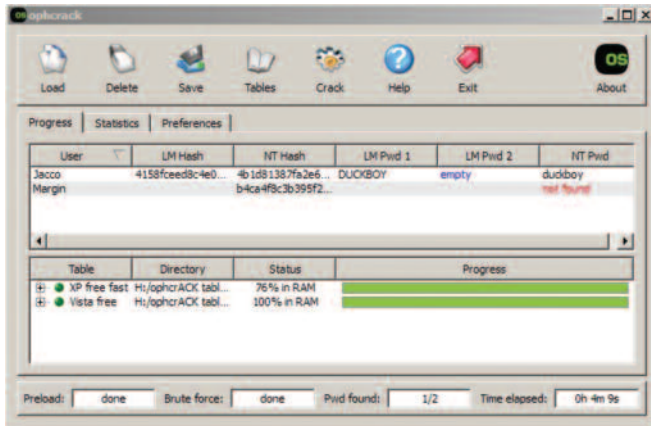
2. Use `samdump2` to get at the account names and password hashes from the SAM:  
`samdump2 (path to)/SAM systemkey.txt>Password.txt`

The result in either case is a file that might look like this one, which has two users, `user11` and `user22`, and two corresponding password hashes.

### LM and NT hashes

Most Windows systems store a type of password hash called an LM hash, and unfortunately, LM hashes are flawed. Passwords longer than seven characters are split into two chunks of seven characters (with zeros added if necessary to make exactly 14 characters.) Both sections of seven characters can usually be guessed or brute-forced in a matter of hours.

All modern versions of Windows software also use a more secure hashing function called the NT hash, so it is wise to ensure that none of the computers in your organization use the LM hash. The only time you might need LM hashes is if you have Windows 95 or 98 clients or Apple Macintosh clients on your network. You can find out how to disable LM hash usage on your computers at:  
<http://support.microsoft.com/kb/299656>



**Ophcrack successfully cracks user Jacco's password "DUCKBOY."**

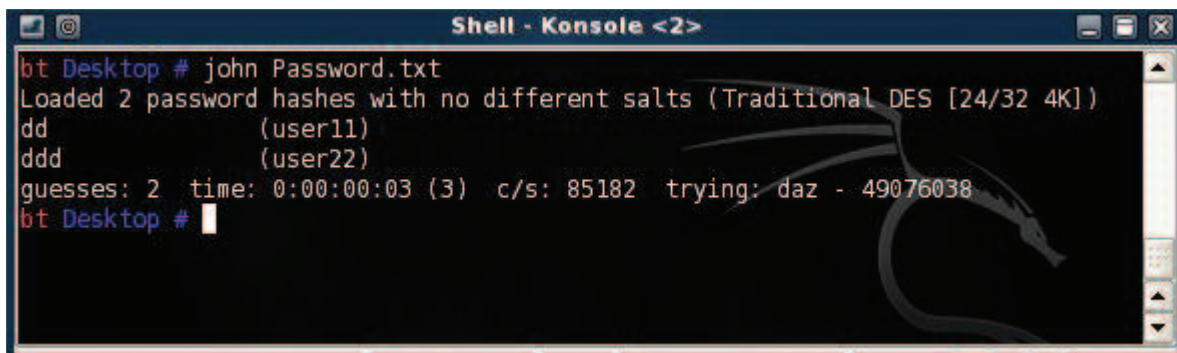
*Note: Any passwords longer than 14 characters will always be stored using an NT hash only – an LM hash will not be used even if LM hashing has not been disabled.*

Just about any LM hash and some NT hashes can actually be cracked to reveal the original password in a matter of minutes using rainbow tables. These are pre-computed lists of (almost) all possible passwords and their resulting LM hashes. Cracking a given hash is thus a simple matter of using a program like Ophcrack to look it up in the table to find the corresponding password.

## Carrying Out an Offline Attack Using John

1. To run a test on the list of hashes, simply type

```
john Password.txt
```



**John brute-forces passwords for user11 (dd) and user22 (ddd).**

## Cracking LM Hashes Using Rainbow Tables with Ophcrack

1. Download and install Ophcrack for Windows of Linux from:

<http://ophcrack.sourceforge.net/download.php>

and a set of rainbow tables from:

<http://ophcrack.sourceforge.net/tables.php>

2. Click Load and select the password hashes that you got using one of the techniques outlines above

3. Click Crack to find the passwords

In this case, the user Jacco's password DUCKBOY was looked up and found in 58 seconds. The user Margin's password (which in fact was longer than 14 characters and therefore only stored as an NT hash) could not be found, after searching all hashes held in the "XP free fast" and "Vista free" rainbow tables.

When run with no options, John gets to work on the passwordlist.txt file, first attempting a single attack, using login information from the password file. It then carries out a wordlist attack using the default wordlist supplied with John, followed by a brute force attack.

In the example above, John finds the simple passwords dd (user1) and ddd (for user2) in a fraction of a second. This illustrates the point that short passwords can be found easily, and also shows the power of hashing functions. The two passwords dd and ddd differ only by one "d," yet the DES hashes they produce are identical in length and completely different. Very similar input leads to completely different output, and input of dif-

## The Do-It-Yourself Security Audit

ferent lengths produces output that is always the same length – in this case 13 characters.

John stores any passwords it cracks in a results files called `john.pot`, and you can view these passwords and their associated usernames by typing

```
john --show Password.txt
```

You can try to crack passwords in more than one list at once simply by adding the names of the extra lists:

```
john passwordlist.txt passwordlist1.txt passwordlist3.txt
```

There are many, many other options you can use to refine how john runs. One of the most useful is

```
john --users=0
```

which only attempts to crack root user (UID=0) passwords.

For a complete list of options and examples go to: <http://www.openwall.com/john/doc>

Running John regularly on all the password hashes on your system will give you an idea of the proportion of your users' passwords that are insecure. You could then:

- Consider changing your password policies to reduce that proportion (perhaps by increasing the minimum length.)
- Contact users with weak passwords and ask them to change them.
- Consider a user education program to help them select more secure passwords. ■



# Checking Wireless Security with aircrack-ng

Wireless networking can be a very big security risk for your organization for at least two reasons:

- A hacker could get access to your network if they can get within range of a wireless access point that is not protected by encryption (an "open" network) or if the encryption key can easily be guessed
- A hacker could set up a so-called "rogue" open access point, and then trick users into connecting to that rather than the company's real wireless access point. Once connected the hacker can usually gain access to the user's computer and any data sent to the rogue access point.

### Wireless Encryption

To connect to your wireless network, a hacker first needs to authenticate himself by providing an encryption key (unless it is an "open" access, which is an open invitation to hackers). Most organizations use a system called WPA Enterprise, which is generally very secure, but smaller organizations may use alternative systems called WPA, WPA2, or WEP.

WEP is a very insecure system, and a hacker can often

get access to a network that uses WEP encryption in less than two minutes. (For more information about this, see

<http://www.enterprisenetworkingplanet.com/netsecur/article.php/3670961>).

WPA and WPA2, on the other hand, are both very secure -- unless the WPA key needed to access the wireless network is short enough to be brute-forced (in a similar way to that illustrated earlier using John).

To maintain security it is important to ensure some form of WPA with long, random passwords protects your wireless networks. There is also a danger that users within your organization decide to connect their own access points to the network for their own purposes -- perhaps so that they can use a Wi-Fi equipped mobile device -- or that a hacker sets up an open access point.

These unauthorized or rogue access points are security risks, so it is important to scan your workplace regularly to detect them as quickly as possible.



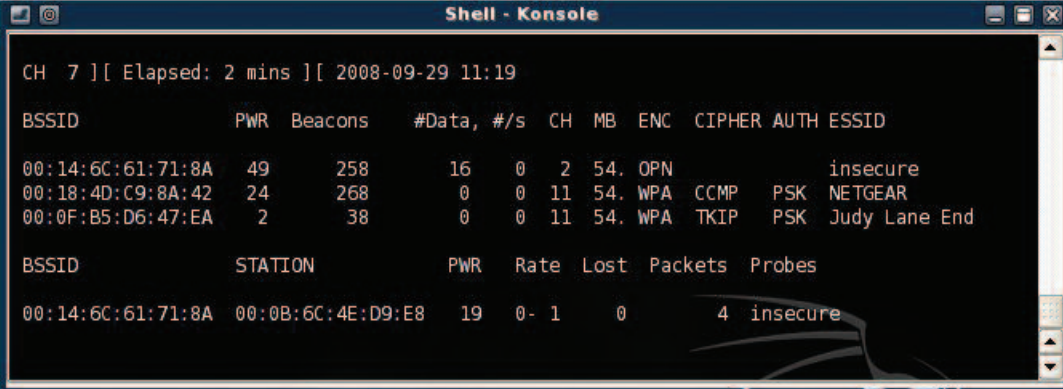
Jupiterimages

“

To maintain security it is important to ensure some form of WPA with long, random passwords protects your wireless networks.

”

## The Do-It-Yourself Security Audit



```
CH 7 ][ Elapsed: 2 mins ][ 2008-09-29 11:19

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:14:6C:61:71:8A  49    258      16    0   2  54.  OPN             insecure
00:18:4D:C9:8A:42   24    268       0    0  11  54.  WPA  CCMP  PSK  NETGEAR
00:0F:B5:D6:47:EA    2     38       0    0  11  54.  WPA  TKIP  PSK  Judy Lane End

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
00:14:6C:61:71:8A  00:0B:6C:4E:D9:E8  19   0- 1    0         4 insecure
```

Airodump-ng detects one open and two WPA protected access points.

### Looking for Rogue Access Points

The easiest way to scan for rogue access points, including hidden access points that do not broadcast their network name and that many people believe are therefore invisible to scanners, is to use a BackTrack 3 tool called airodump-ng, part of a suite of wireless tools called aircrack-ng.

Before using these tools, the wireless card on your penetration testing machine must be put in to monitor mode, allowing it to access packets without being associated (connected) to any particular access point.

1. Put the wireless card in monitor mode:

```
airmon-ng stop ath0
airmon-ng start wifi0
```

2. Start airodump-ng:

```
airodump-ng ath0
```

Airodump-ng will display a list of every access point it can detect in the vicinity, and information including:

**BSSID** – the MAC address of a detected access point

**CH** – the wireless channel it is operating on

**ENC** – the encryption system it is using (WPA, WEP or OPN [for open networks])

**ESSID** – the network name. This may remain blank until a device connects to it if the network is set to "hidden." Note: airodump-ng will still detect the network, even if it is "hidden"

**STATION** – the MAC address of a connect device

By looking at the results of airodump-ng it is possible to spot rogue access points (which will then need to be

tracked down, and to ensure that all authorized access points are using WPA encryption.

If your organization is large it is wise to walk around and run airodump-ng from many different places to ensure that you cover the whole area. You should also walk around outside the buildings to see if any authorized access points can be detected in public places. If this is the case you should consider moving your access points or reducing the power of the internal radio to minimize this, even if WPA protects access.

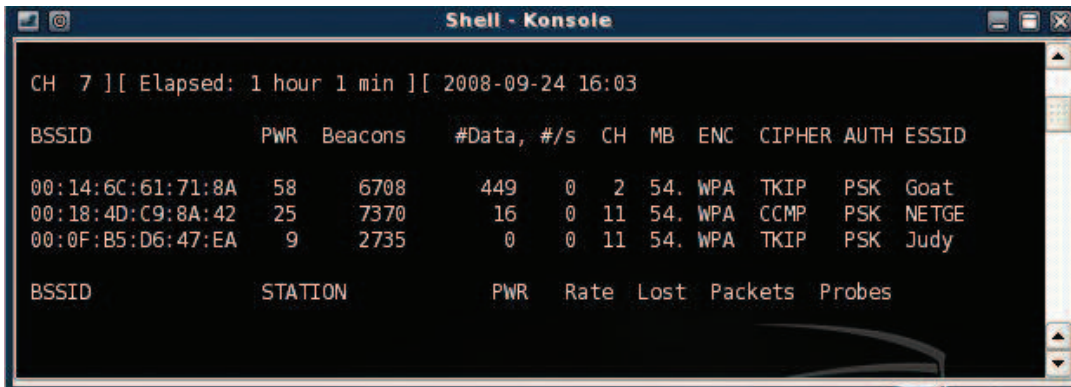
### Mac Address Filtering

Some organizations set up Mac address filtering on their access point as an alternative to using WPA encryption under the mistaken belief that it is only possible for a device to connect to the network if its MAC address is on a list of devices recognized by the access point.

But as can be seen in the illustration above, airodump-ng records the MAC address of any clients connected any access points (under "STATION".) A hacker can easily overcome MAC filtering by making a note of a permitted MAC address (in this case 00:0B:6C:4E:D9:E8), waiting until that client disconnects, and then connecting to the network using that MAC address. This is easy to do this in Linux using a utility called macchanger, which changes the true MAC address of a network interface card to any arbitrary false MAC address. To spoof the MAC address 00:0B:6C:4E:D9:E8 a hacker need only open a terminal window and type:

```
airmon-ng stop ath0
ifconfig wifi0 down
macchanger -m 00:0B:6C:4E:D9:E8
ifconfig wifi0 up
```

## The Do-It-Yourself Security Audit



```
CH 7 ][ Elapsed: 1 hour 1 min ][ 2008-09-24 16:03

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:14:6C:61:71:8A  58   6708     449    0   2   54.  WPA   TKIP  PSK  Goat
00:18:4D:C9:8A:42   25   7370      16    0  11   54.  WPA   CCMP  PSK  NETGE
00:0F:B5:D6:47:EA    9   2735       0    0  11   54.  WPA   TKIP  PSK  Judy

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
```

airodump-ng detects three WPA protected networks.

MAC address filtering therefore offers no security against a hacker, and should not be used as an alternative to WPA encryption.

### Cracking WPA

Hackers can get access to WPA or WPA2 (but not WPA-Enterprise) protected networks if the password is not long or uses a guessable password (or passphrase). To do this they need to capture the packets that are transmitted when a user authenticates and joins the wireless network, known as the WPA handshake. Once the hacker has captured the WPA handshake they can subject it to a dictionary attack.

### Testing WPA Passwords

1. In a terminal window, place the wireless card in monitor mode as run airodump-ng as before:  
`airmon-ng stop ath0`

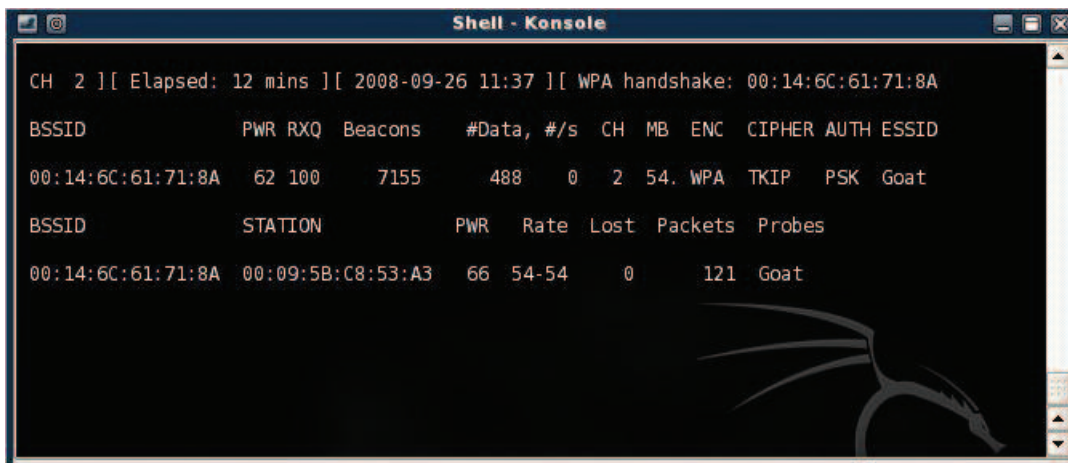
```
airmon-ng start wifi0
airodump-ng ath0
```

2. Start airodump-ng to capture a WPA handshake on the network "GOAT":

```
airodump-ng -c 2 --bssid
00:14:6C:61:71:8A -w wpacapture ath0
```

In this example we are testing the network on channel 2 (-c 2) which has the BSSID 00:14:6C:61:71:8A, and attempting to capture a WPA handshake and write it to a file called wpacapture (-w wpacapture)

3. Connect another device to the WPA network, so that airodump-ng can capture the WPA handshake. When this is successful the message "WPA handshake: 00:14:6C:61:71:8A" will appear in the top right of the screen as illustrated below:



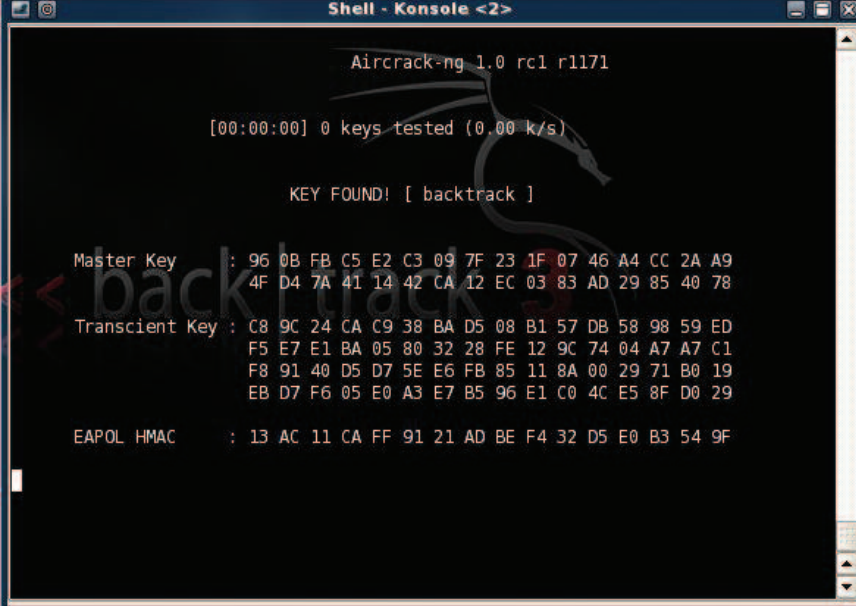
```
CH 2 ][ Elapsed: 12 mins ][ 2008-09-26 11:37 ][ WPA handshake: 00:14:6C:61:71:8A

BSSID          PWR RXQ Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:14:6C:61:71:8A  62 100   7155     488    0   2   54.  WPA   TKIP  PSK  Goat

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
00:14:6C:61:71:8A 00:09:5B:C8:53:A3  66  54-54    0    121  Goat
```

airodump-ng displaying a successful WPA handshake capture

## The Do-It-Yourself Security Audit



```
Shell - Konsole <2>
Aircrack-ng 1.0 rc1 r1171

[00:00:00] 0 keys tested (0.00 k/s)

KEY FOUND! [ backtrack ]

Master Key   : 96 0B FB C5 E2 C3 09 7F 23 1F 07 46 A4 CC 2A A9
              4F D4 7A 41 14 42 CA 12 EC 03 83 AD 29 85 40 78
Transcient Key : C8 9C 24 CA C9 38 BA D5 08 B1 57 DB 58 98 59 ED
              F5 E7 E1 BA 05 80 32 28 FE 12 9C 74 04 A7 A7 C1
              F8 91 40 D5 D7 5E E6 FB 85 11 8A 00 29 71 B0 19
              EB D7 F6 05 E0 A3 E7 B5 96 E1 C0 4C E5 8F D0 29
EAPOL HMAC   : 13 AC 11 CA FF 91 21 AD BE F4 32 D5 E0 B3 54 9F
```

aircrack-ng find the password "backtrack"

4. Run aircrack-ng on the wpacapture file containing the handshake, to see if the password is easily crackable using guesses from a word list wordlist.txt:

```
aircrack-ng -w wordlist.txt -b
00:14:6C:61:71:8A wpacapture*.cap
```

5. If successful, aircrack-ng will display the WPA password: see image above

6. Important note: Due to an apparent bug, on some computers aircrack-ng running from BackTrack3 will fail to find the password even if it is in the password list. To check whether this is the case with your setup, set the passphrase for an authorized access point to a simple password like "test", and run the process above with a wordlist.txt file containing only this password. If it fails

to find the password, try running the aircrack-ng suite on another machine.

If you can crack your own WPA password then it is essential that you change it to something more secure. The ideal WPA password is a random string of 63 characters, and you can generate a suitable one online at: <https://www.grc.com/passwords.htm>. ■

*Paul Rubens is an IT consultant based in Marlow, England, and has been writing about business technology for leading US and UK publications for almost 20 years.*