Kali Linux Web Penetration Testing Cookbook

Web applications are a huge point of attack for malicious hackers and a critical area for security professionals and penetration testers to lock down and secure. Kali Linux is a Linux-based penetration testing platform and operating system that provides a huge array of testing tools, many of which can be used specifically to execute web penetration testing.

Starting with setting up a testing laboratory, this book will give you the skills you need to cover every stage of a penetration test, from gathering information about the system and the application to identifying vulnerabilities through manual testing and the use of vulnerability scanners, to both basic and advanced exploitation techniques that may lead to a full compromise of the system.

What this book will do for you...

- Use crawlers and spiders to investigate an entire website in minutes
- Discover security vulnerabilities in web applications in the web browser and using command-line tools
- Exploit vulnerabilities that require a complex setup, run custom-made exploits, and prepare for extraordinary scenarios
- Set up man-in-the-middle attacks and use them to identify and exploit security flaws within the communication between users and the web server
- Create a malicious site that will find and exploit vulnerabilities in the user's web browser
- Repair the most common web vulnerabilities and understand how to prevent them becoming a threat to a site's security

Inside the Cookbook...

- A straightforward and easy-to-follow format
- A selection of the most important tasks and problems
- Carefully organized instructions to solve problems efficiently
- Clear explanations of what you did
- Solutions that can be applied to solve real-world problems

Kali Linux Web Penetration Testing Cookbook

Over 80 recipes on how to identify, exploit, and test web application security with Kali Linux 2
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Setting Up Kali Linux'
- A synopsis of the book’s content
- More information on Kali Linux Web Penetration Testing Cookbook
About the Author

**Gilberto Nájera-Gutiérrez** leads the Security Testing Team (STT) at Sm4rt Security Services, one of the top security firms in Mexico.

He is also an Offensive Security Certified Professional (OSCP), an EC-Council Certified Security Administrator (ECSA), and holds a master's degree in computer science with specialization in artificial intelligence.

He has been working as a Penetration Tester since 2013 and has been a security enthusiast since high school; he has successfully conducted penetration tests on networks and applications of some of the biggest corporations in Mexico, such as government agencies and financial institutions.
Nowadays, information security is a hot topic all over the news and the Internet; we hear almost every day about web page defacements, data leaks of millions of user accounts and passwords or credit card numbers from websites, and identity theft on social networks; terms such as cyber attack, cybercrime, hacker, and even cyberwar are becoming a part of the daily lexicon in the media.

All this exposition to information security subjects and the real need to protect sensitive data and their reputation have made organizations more aware of the need to know where their systems are vulnerable; especially, for the ones that are accessible to the world through the Internet, how could they be attacked, and what will be the consequences, in terms of information lost or system compromise if an attack was successful. And more importantly, how to fix those vulnerabilities and minimize the risk.

This task of detecting vulnerabilities and discovering their impact on organizations is the one that is addressed through penetration testing. A penetration test is an attack or attacks made by a trained security professional who is using the same techniques and tools that real hackers use in order to discover all the possible weak spots in the organization's systems. These weak spots are exploited and their impact is measured. When the test is finished, the penetration tester informs all their findings and tells how they can be fixed to prevent future damage.

In this book, we follow the whole path of a web application penetration test and, in the form of easy-to-follow, step-by-step recipes, show how the vulnerabilities in web applications and web servers can be discovered, exploited, and fixed.

**What this book covers**

*Chapter 1, Setting Up Kali Linux,* takes the reader through the process of configuring and updating the system; also, the installation of virtualization software is covered, including the configuration of the virtual machines that will comprise our penetration testing lab.
Chapter 2, *Reconnaissance*, enables the reader to put to practice some of the information gathering techniques in order to gain intelligence about the system to be tested, the software installed on it, and how the target web application is built.

Chapter 3, *Crawlers and Spiders*, shows the reader how to use these tools, which are a must in every analysis of a web application, be it a functional one or more security focused, such as a penetration test.

Chapter 4, *Finding Vulnerabilities*, explains that the core of a vulnerability analysis or a penetration test is to discover weak spots in the tested applications; recipes are focused on how to manually identify some of the most common vulnerabilities by introducing specific input values on applications' forms and analyzing their outputs.

Chapter 5, *Automated Scanners*, covers a very important aspect of the discovery of vulnerabilities, the use of tools specially designed to automatically find security flaws in web applications: automated vulnerability scanners.

Chapter 6, *Exploitation – Low Hanging Fruits*, is the first chapter where we go further than just identifying the existence of some vulnerability. Every recipe in this chapter is focused on exploiting a specific type of vulnerability and using that exploitation to extract sensitive information or gain a more privileged level of access to the application.

Chapter 7, *Advanced Exploitation*, follows the path of the previous chapter; here, the reader will have the opportunity to practice a more advanced and a more in-depth set of exploitation techniques for the most difficult situations and the most sophisticated setups.

Chapter 8, *Man in the Middle Attacks*. Although not specific to web applications, MITM attacks play a very important role in the modern information security scenario. In this chapter, we will see how these are performed and what an attacker can do to their victims through such techniques.

Chapter 9, *Client-Side Attacks and Social Engineering*, explains how it's constantly said that the user is the weakest link in the security chain, but traditionally, penetration testing assessments exclude client-side attacks and social engineering campaigns. It is the goal of this book to give the reader a global view on penetration testing and to encourage the execution of assessments that cover all the aspects of security; this is why in this chapter we show how users can be targeted by hackers through technological and social means.

Chapter 10, *Mitigation of OWASP Top 10*, shows that organizations hire penetration testers to attack their servers and applications with the goal of knowing what's wrong, in order to know what they should fix and how. This chapter covers that face of penetration testing by giving simple and direct guidelines on what to do to fix and prevent the most critical web application vulnerabilities according to OWASP (Open Web Application Security Project).
In this chapter, we will cover:

- Updating and upgrading Kali Linux
- Installing and running OWASP Mantra
- Setting up the Iceweasel browser
- Installing VirtualBox
- Creating a vulnerable virtual machine
- Creating a client virtual machine
- Configuring virtual machines for correct communication
- Getting to know web applications on a vulnerable VM

**Introduction**

In the first chapter, we will cover how to prepare our Kali Linux installation to be able to follow all the recipes in the book and set up a laboratory with vulnerable web applications using virtual machines.

**Updating and upgrading Kali Linux**

Before we start testing web applications' security, we need to be sure that we have all the necessary up-to-date tools. This recipe covers the basic task of keeping Kali Linux and its tools at their most recent versions.
Setting Up Kali Linux

Getting ready

We start from having Kali Linux installed as the main operating system on a computer with Internet access; the version that we will be using through this book is 2.0. You can download the live CD and installer from https://www.kali.org/downloads/.

How to do it...

Once you have a working instance of Kali Linux up and running, perform the following steps:

1. Log in as a root on Kali Linux; the default password is "toor", without the quotes. You can also use su to switch the user or sudo to execute single commands if using a regular user is preferred instead of root.

2. Open a terminal.

3. Run the `apt-get update` command. This will download the updated list of packages (applications and tools) that are available to install.

```
root@kali:~$ apt-get update
```

4. Once the update is finished, run the following command to update non-system packages to their last stable version:

```
apt-get upgrade
```
5. When asked to continue, press Y and then press Enter.

6. Next, let's upgrade our system. Type the following command and press Enter:

```
apt-get dist-upgrade
```

7. Now, we have our Kali Linux up-to-date and ready to continue.
How it works...

In this recipe, we have covered a basic procedure for package update in Debian-based systems (such as Kali Linux). The first call to `apt-get` with the `update` parameter downloaded the most recent list of packages available for our specific system in the configured repositories. After it downloads and installs all the packages that have the most recent versions in the repository, the `dist-upgrade` parameter downloads and installs system packages (such as kernel and kernel modules) not installed with `upgrade`.

In this book, we assume that Kali Linux is installed as the main operating system on the computer; there is also the option of installing it in a virtual machine. In such a case, skip the recipe called Installing VirtualBox and configure the network options of your Kali VM as stated in Configuring virtual machines for correct communication.

There's more...

There are tools, such as the Metasploit Framework, that have their own update commands; these can be executed after following this recipe. The command is as follows:

`msfupdate`

Installing and running OWASP Mantra

People in OWASP (Open Web Application Security Project, https://www.owasp.org/) have put together a Mozilla Firefox mod with plenty of add-ons aimed at helping penetration testers and developers to test web applications for bugs or security flaws. In this recipe, we will install OWASP-Mantra (http://www.getmantra.com/) in our Kali Linux, run it for the first time, and see some of its features.

Most of the web application penetration testing is done through a web browser; that's the reason why we need to have one with the correct set of tools to perform such a task. The OWASP Mantra includes a collection of add-ons to perform tasks, such as:

- Sniffing and intercepting HTTP requests
- Debugging client-side code
- Viewing and modifying cookies
- Gathering information about sites and applications
Getting ready

Fortunately for us, OWASP Mantra is included in the default Kali Linux repositories. So, to make sure that we get the latest version of the browser, we need to update the packages list:

```
apt-get update
```

How to do it...

1. Open a terminal and run:
   
   ```
   apt-get install owasp-mantra-ff
   ```

2. After the installation is finished, navigate to menu: **Applications | 03 - Web Application Analysis | Web Vulnerability Scanners | owasp-mantra-ff** to start Mantra for the first time. Or use a terminal with the following command:
   
   ```
   owasp-mantra-ff
   ```
3. With the new browser open, click on the OWASP logo and then **Tools**. Here we can access all the tools that OWASP Mantra includes.

4. We will use some of these tools in later chapters.

**See also**

You may also be interested in **Mantra on Chromium (MoC)**, which is an alternative release of Mantra based on the Chromium web browser. Currently, it is only available for Windows: [http://www.getmantra.com/mantra-on-chromium.html](http://www.getmantra.com/mantra-on-chromium.html)
Setting up the Iceweasel browser

If we don't like OWASP Mantra, we can use the latest version of Firefox and install our own selection of testing-related add-ons. Kali Linux includes Iceweasel, another variant of Firefox, which we will use in this recipe to see how to install our testing tools in a browser.

How to do it...

1. Open Iceweasel and navigate to Tools | Add-ons, as shown in the following screenshot:
2. In the search box, type `tamper data` and hit `Enter`.

3. Click on `Install` in the `Tamper Data` add-on.

4. A dialog box will pop up, asking us to accept the EULA; click on `Accept and Install...`

   ![Tamper Data](image)

   You might have to restart your browser to complete the installation of certain add-ons.

5. Next, we search for `cookies manager+` in the search box.

6. Click on `Install` in the `Cookies Manager+` add-on.

7. Now, search and install `Firebug`.

8. Search and install `Hackbar`.

9. Search and install `HTTP Requester`.

10. Search and install `Passive Recon`.

**How it works...**

So far we've just installed some tools on our web browser but what are these tools good for when it comes to penetration-testing a web application?

- **Cookies Manager+:** This add-on will allow us to view and sometimes modify the value of cookies the browser receives from applications.

- **Firebug:** This is a must-have for any web developer; its main function is to be an in-line debugger for web pages. It will also be useful when you have to perform some client-side modifications to pages.
Hackbar: This is a very simple add-on that helps us to try different input values without having to change or rewrite the full URL. We will be using this a lot when doing manual checks for Cross-site scripting and injections.

Http Requester: With this tool it is possible to craft HTTP requests including GET, POST, and PUT methods and watch the raw response from the server.

Passive Recon: It allows us to get public information about the website being visited by querying DNS records, Whois, and searching information, such as email addresses, links, and collaborators in Google, among other things.

Tamper Data: This add-on has the ability to capture any request on the server just after it is sent by the browser, thus giving us the chance to modify the data after introducing it in the application's forms and before it reaches the server.

There's more...

Other add-ons that could prove useful for web application penetration testing are:

- XSS Me
- SQL Inject Me
- FoxyProxy
- iMacros
- FirePHP
- RESTClient
- Wappalyzer

Installing VirtualBox

This is the first of the four recipes that will help us to get a virtual laboratory up and running to practice our penetration tests. We will use a VirtualBox to run the virtual machines in such a laboratory. In this recipe, we will see how to install VirtualBox and get it working.

Getting ready

Before we install anything in Kali Linux, we must make sure that we have the latest version of package lists:

```
apt-get update
```
How to do it...

1. Our first step is the actual installation of VirtualBox:

   `apt-get install virtualbox`

   ```
   root@kali:~ # apt-get install virtualbox
   Reading package lists... Done
   Building dependency tree
   Reading state information... Done
   The following extra packages will be installed:
   dkms libsoap4 libvncserver0 linux-headers-3.18.0-kali1-amd64
   linux-headers-3.18.0-kali1-common linux-headers-amd64 linux-kbuild-3.18
   virtualbox-dkms virtualbox-qt
   Suggested packages:
   libvncserver0-dbg vde2 virtualbox-guest-additions-iso
   Recommended packages:
   linux-image
   The following NEW packages will be installed:
   dkms libsoap4 libvncserver0 linux-headers-3.18.0-kali1-amd64
   linux-headers-3.18.0-kali1-common linux-headers-amd64 linux-kbuild-3.18
   virtualbox virtualbox-dkms virtualbox-qt
   3 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
   Need to get 27.2 MB of archives.
   After this operation, 124 MB of additional disk space will be used.
   Do you want to continue [Y/n]?
   Y
   ```

2. After the installation finishes, we will find VirtualBox in the menu by navigating to Applications | Usual applications | Accessories | VirtualBox. Alternatively, we can call it from a terminal:

   `virtualbox`
Now, we have VirtualBox running and we are ready to set up the virtual machines to make our own testing laboratory.

**How it works...**

VirtualBox will allow us to run multiple machines inside our Kali Linux computer through virtualization. With this, we can mount a full laboratory with different computers using different operating systems and run them in parallel as far as the memory resources and processing power of our Kali host allow us to.

**There's more...**

The VirtualBox Extension Pack gives the VirtualBox's virtual machine extra features, such as USB 2.0/3.0 support and Remote Desktop capabilities. It can be downloaded from [https://www.virtualbox.org/downloads](https://www.virtualbox.org/downloads). After it is downloaded, just double click on it and VirtualBox will do the rest.

**See also**

There are some other virtualization options out there. If you don't feel comfortable using VirtualBox, you may want to try:

- VMware Player/Workstation
- Qemu
- Xen
- KVM

**Creating a vulnerable virtual machine**

Now we are ready to create our first virtual machine, it will be the server that will host the web applications we'll use to practice and improve our penetration testing skills.

We will use a virtual machine called OWASP-bwa (OWASP Broken Web Apps) that is a collection of vulnerable web applications specially set up to perform security testing.
How to do it...

1. Go to http://sourceforge.net/projects/owaspbwa/files/ and download the latest release's .ova file. At the time of writing, it is OWASP_Broken_Web_Apps_VM_1.1.1.ova.

2. Wait for the download to finish and then open the file.
3. VirtualBox's import dialog will launch. If you want to change the machine's name or description, you can do it by double-clicking on the values. We will name it `vulnerable_vm` and leave the rest of the options as they are. Click on **Import**.

4. The import should take a minute and after that we will see our virtual machine displayed in VirtualBox's list. Let's select it and click on **Start**.
5. After the machine starts, we will be asked for login and password, type `root` as the login and `owaspbwa` as the password and we are set.

**How it works...**

OWASP-bwa is a project aimed at providing security professionals and enthusiasts with a safe environment to develop attacking skills and identify and exploit vulnerabilities in web applications, in order to be able to help developers and administrators fix and prevent them.

This virtual machine includes different types of web applications, some of them are based on PHP, some in Java; we even have a couple of .NET-based vulnerable applications. There are also some vulnerable versions of known applications, such as WordPress or Joomla.
See also

There are many options when we talk about vulnerable applications and virtual machines. A remarkable website that holds a great collection of such applications is VulnHub (https://www.vulnhub.com/). It also has walkthroughs that will help you to solve some challenges and develop your skills.

In this book, we will use another virtual machine for some recipes: bWapp Bee-box, which can also be downloaded from VulnHub: https://www.vulnhub.com/entry/bwapp-bee-box-v16,53/.

Creating a client virtual machine

When we get to the man in the middle (MITM) and client-side attacks, we will need another machine to make requests to the already set up server. In this recipe, we will download a Microsoft Windows virtual machine and import it to VirtualBox.

How to do it...

1. First we need to go to the download site http://dev.modern.ie/tools/vms/#downloads.
2. Through this book we will use the IE8 on Win7 virtual machine.
3. After the file is downloaded, we need to unzip it. Go to where it was downloaded.
4. Right-click on it and then click on **Extract Here**.
5. Once extracted, open the .ova file and import it in VirtualBox.

6. Now, start the virtual machine (named **IE8 - Win7**) and we will have our client ready:
Microsoft provides these virtual machines for developers to test their applications with the help of different versions of Windows and Internet Explorer with a free license limited to 30 days, which is enough for us to practice.

As penetration testers, it is important to be aware that real-world applications can be multiplatform and that users of those applications may have a lot of different systems and web browsers to communicate with them; knowing this, we should be prepared to perform successful tests with any of the client-server infrastructure combinations.

See also

As for server and client virtual machines, if you are not comfortable using an already built configuration, you can always build and configure your own virtual machines. Here is some information about how to do it: https://www.virtualbox.org/manual/.
To be able to communicate with our virtual server and client, we need to be in the same network segment; however, having virtual machines with known vulnerabilities in our local network may pose an important security risk. To avoid this risk, we will perform a special configuration in VirtualBox to allow us to communicate with both server and client virtual machines from our Kali Linux host without exposing them to the network.

### Getting ready

Before we proceed, open VirtualBox and make sure that the vulnerable server and client virtual machines are turned off.

### How to do it...

1. In VirtualBox navigate to File | Preferences... | Network.
2. Select the Host-only Networks tab.
3. Click on the ( ) button to add a new network.
4. The new network (vboxnet0) will be created and its "details window" will pop up. If it doesn’t, select the network and click on the ( ) button to edit its properties.
5. In this dialog box, you can specify the network configuration, if it doesn't interfere with your local network configuration, leave it as it is. You may change it and use some other address in the segments reserved for local networks (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16).

6. After proper configuration is done, click **OK**.

7. The next step is to configure the vulnerable virtual machine (vulnerable_vm). Select it and go to its settings.

8. Click **Network** and, in the **Attached to:** drop-down menu, select **Host-only Adapter**.

9. In **Name** , select **vboxnet0**.

10. Click **OK**.

11. Follow steps 7 to 10 in the client virtual machine (**IE8 - Win7**).

12. After having both virtual machines configured, let's test if they can actually communicate. Start both the machines.
13. Let's see the network configuration of our host system: open a terminal and type:

\texttt{ifconfig}

\begin{verbatim}
root@kali:~ # ifconfig
eth0 Link encap:Ethernet  HWaddr 88:ac:6f:ff:7e:67
  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:1008
  RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
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:349 errors:0 dropped:0 overruns:0 frame:0
  TX packets:349 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0  txqueuelen:0
  RX bytes:99649 (97.3 KiB)  TX bytes:99649 (97.3 KiB)

vboxnet0 Link encap:Ethernet  HWaddr 0a:0b:27:00:00:00
  inet addr:192.168.56.1  Bcast:192.168.56.255  Mask:255.255.255.0
  inet6 addr: fe80::0b02:7bff:fe00:0/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:0  errors:0  dropped:0  overruns:0  frame:0
  TX packets:153 errors:0  dropped:0  overruns:0  carrier:0
  collisions:0  txqueuelen:1008
\end{verbatim}

14. We can see that we have a network adapter called \texttt{vboxnet0} and it has the IP address \texttt{192.168.56.1}. Depending on the configuration you used, this may vary.

15. Log into \texttt{vulnerable_vm} and check its IP address for adapter \texttt{eth0}:

\texttt{ifconfig}

16. Now, let's go to our client machine \texttt{IE8 - Win7}; open a command prompt and type:

\texttt{ipconfig}

17. Now, we have the IP addresses of our three machines:

- 192.168.56.1 for the host
- 192.168.56.102 for \texttt{vulnerable_vm}
- 192.168.56.103 for \texttt{IE8 - Win7}
18. To test the communication, we are going to ping both virtual machines from our host:

```
ping -c 4 192.168.56.102
ping -c 4 192.168.56.103
```

```
root@kali:~# ping -c 4 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.369 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.243 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.252 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.247 ms
--- 192.168.56.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
Round trip times min/avg/max/mdev = 0.243/0.277/0.369/0.056 ms
```

```
root@kali:~# ping -c 4 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.
From 192.168.56.1 icmp_seq=1 Destination Host Unreachable
From 192.168.56.1 icmp_seq=2 Destination Host Unreachable
From 192.168.56.1 icmp_seq=3 Destination Host Unreachable
From 192.168.56.1 icmp_seq=4 Destination Host Unreachable
--- 192.168.56.103 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 000ms
```

Ping sends an ICMP request to the destination and waits for the reply; this is useful to test whether communication is possible between two nodes in the network.

19. We do the same from both the virtual machines thus checking communication to the server and the other virtual machine.

20. The IE8 - Win7 machine may not respond to pings; that's normal because Windows 7 is configured by default to not respond to ping requests. To check connectivity in this case, we can use arping from the Kali host:

```
arping -c 4 192.168.56.103
```

**How it works...**

A host-only network is a virtual network that acts as a LAN but its reach is limited to the host that is running the virtual machines without exposing them to external systems. This kind of network also provides a virtual adapter for the host to communicate with the virtual machines as if they were in the same network segment.

With the configuration we just made, we will be able to communicate between a client and server and both of them can communicate with the Kali Linux host, which will act as the attacking machine.
Getting to know web applications on a vulnerable VM

OWASP-bwa contains many web applications, intentionally made vulnerable to the most common attacks. Some of them are focused on the practice of some specific technique while others try to replicate real-world applications that happen to have vulnerabilities.

In this recipe, we will take a tour of our vulnerable_vm and get to know some of the applications it includes.

Getting ready

We need to have our vulnerable_vm running and its network correctly configured. For this book, we will be using 192.168.56.102 as its IP address.

How to do it...

1. With vulnerable_vm running, open your Kali Linux host's web browser and go to http://192.168.56.102. You will see a list of all applications the server contains:
2. Let’s go to **Damn Vulnerable Web Application**.

3. Use `admin` as a user name and `admin` as a password. We can see a menu on the left; this menu contains links to all the vulnerabilities that we can practice in this application: `Brute Force`, `Command Execution`, `SQL Injection`, and so on. Also, the **DVWA Security** section is where we can configure the security (or complexity) levels of the vulnerable inputs.

4. Log out and return to the server’s homepage.
5. Now we click on **OWASP WebGoat.NET**. This is a .NET application where we will be able to practice file and code injection attacks, cross-site scripting, and encryption vulnerabilities. It also has a WebGoat Coins Customer Portal that simulates a shopping application and can be used to practice not only the exploitation of vulnerabilities but also their identification.

6. Now return to the server’s home page.

7. Another interesting application included in this virtual machine is BodgeIt, which is a minimalistic version of an online store based on JSP—it has a list of products that we can add to a shopping basket, a search page with advanced options, a registration form for new users, and a login form. There is no direct reference to vulnerabilities; instead, we will need to look for them.
8. We won't be able to look at all the applications in a single recipe, but we will be using some of them in this book.

How it works...

The applications in the home page are organized in the following six groups:

- **Training applications**: These are the ones that have sections dedicated to practice-specific vulnerabilities or attack techniques; some of them include tutorials, explanations, or other kind of guidance.

- **Realistic, intentionally vulnerable applications**: Applications that act as real-world applications (stores, blogs, and social networks) and are intentionally left vulnerable by their developers for the sake of training.

- **Old (vulnerable) versions of real applications**: Old versions of real applications, such as WordPress and Joomla are known to have exploitable vulnerabilities; these are useful to test our vulnerability identification skills.

- **Applications for testing tools**: The applications in this group can be used as a benchmark for automated vulnerability scanners.

- **Demonstration pages / small applications**: These are small applications that have only one or a few vulnerabilities, for demonstration purposes only.

- **OWASP demonstration application**: OWASP AppSensor is an interesting application, it simulates a social network and could have some vulnerabilities in it. But it will log any attack attempts, which is useful when trying to learn; for example, how to bypass some security devices such as a web application firewall.
Where to buy this book

You can buy Kali Linux Web Penetration Testing Cookbook from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.