



The WordPress Security Learning Center

Penetration Testing Your WordPress Website

Penetration Testing Your WordPress Website



Advanced

Updated January 25, 2016

Penetration testing or “pentesting” your website or network is the act of analyzing your systems to find vulnerabilities that an attacker might exploit.

A **‘white box’** pentest is a penetration test where an attacker has full knowledge of the systems they are attacking. White box penetration testing has the goal of providing maximum information to the penetration tester so that they can more effectively find vulnerabilities in the systems or organization. This information allows the pentester to test as widely and deeply as possible. Information provided to white box pentesters may include network diagrams, source code, access to staff for interviews, configuration information and more.

The WordPress Security Learning Center

From WordPress security fundamentals to expert developer resources, this learning center is meant for every skill level. Get serious about WordPress Security, start right here.

WordPress Security Fundamentals >

WordPress Security For Developers >

WordPress Malware Removal >

Building Blocks ✓

A **'black box'** pentest is one where the attacker has no knowledge and is emulating a real world attacker who would know very little about the targeted systems. Black box penetration testing is usually done from outside the targeted systems and uses reconnaissance to gather any information.

The goal of pentesting is to uncover vulnerabilities in the target systems, prioritize them by risk and manage the removal or mitigation (reduction of risk) of those vulnerabilities.

As a website owner or administrator, adding penetration testing to your list of tools and knowledge is one of the most effective ways to ensure that the websites that you manage stay safe. Penetration testing forces you to think like an attacker and to objectively assess your website vulnerabilities.

There is a large body of knowledge around the theory of penetration testing. This guide is designed to be a practical guide that will quickly get you started with some basic penetration testing tools. We will also provide a basic methodology that will allow you to manage and remove the vulnerabilities that you discover.

Getting Set Up for Penetration Testing

Kali Linux is the de facto standard tool for penetration testers. Kali is a Linux distribution that is created and maintained by **Offensive Security LTD**. Kali used to be called BackTrack and was based on Ubuntu Linux. It was relaunched as Kali in 2013 which is a Debian based distribution.

Installing Kali Linux

For our own penetration testing we use Kali Linux installed as a virtual machine on our laptops. Some of us use VMWare to host the virtual machine, which is a commercial

[Video – The OSI Model & How Information Travels Through The Internet](#)

[Networking for WordPress Administrators](#)

[The Tor Network – FAQ](#)

[Understanding Social Engineering Attacks](#)

[Password Authentication and Password Cracking](#)

[Penetration Testing Your WordPress Website](#)

[How to Restrict WordPress File Permissions](#)

[How to Manually Upgrade WordPress, Themes & Plugins](#)

[Introduction to Brute Force Attacks](#)

Videos, Infographics and More



product. Others use Virtual Box which is free. Another alternative if you're using OS X is Parallels which is also a virtual machine host. All of these work well.

If you're just getting started, we recommend you install Kali Linux with VirtualBox.

1. Download [VirtualBox](#) for your operating system and install it.
2. Download [Kali Linux 64 bit](#) (unless you have an older 32 bit machine and OS).
3. Create a VirtualBox virtual machine to host Kali and select Debian 64 bit as the OS. Make sure you select more than 10 GB of disk space or you will run out. We recommend at least 2048 megabytes of RAM.
4. Boot into the new virtual machine and select the Kali ISO you downloaded as your boot disk.
5. Boot into Kali and don't select the 'Live' option. Instead select the option to 'Install'.

Go through the installation steps Kali provides. We have provided the following video to help you configure your VirtualBox and install Kali Linux.

Protect your websites with the #1
WordPress Security Plugin

[GET PREMIUM](#)

Over 100 million downloads

**You have been temporarily
blocked**

vimeo
Unfortunately, our servers have detected a
high number of requests from your connection. To continue,
please verify that you are a human:



I'm not a robot



reCAPTCHA
[Privacy](#) - [Terms](#)

A Simple Penetration Testing Methodology

Penetration testing is not as simple as learning a single tool and launching it against your website. We suggest a systematic approach that will give you the maximum amount of information about the security posture of your network and will lead to the greatest security at the end of the process.

We suggest that you break down your penetration testing into the following steps:

1. **Reconnaissance** or 'recon'. In this step you will gather information about your network and servers.
2. **Scanning**. In this step you will scan for vulnerabilities and store the vulnerabilities you find in a document along with a severity level from 1 to 5 for each vulnerability. (5 being the most severe)
3. **Exploitation**. This step is optional and most penetration testers do not need to complete it. If you have a manager that requires a real-world demonstration of the danger posed by a vulnerability, you may have to exploit one or more of the vulnerabilities you have discovered to demonstrate they are a real threat.
4. **Mitigation**. In this step you remove the vulnerabilities you have found from your network in order of the highest priority to the lowest. You may also choose to remove a device that has a vulnerability from the network if it is not feasible or too expensive to remove the vulnerability itself.

How to Avoid Prison

Kali Linux, the tools that are included with it and the penetration testing tools and methodologies we discuss here give you the ability to gain unauthorized access to networks and hosts on the Internet and on private networks.

The [Computer Fraud and Abuse Act](#) (CFAA) in the United States provides severe penalties for gaining unauthorized access to a computer system. It also includes penalties for “exceeding authorized access”. What this means is that if you are granted access to penetration test a system and you exceed the privileges a company or individual has granted you, you may be subject to criminal prosecution under the CFAA.

Even if you are outside the United States, there are severe penalties for cyber trespass in many other countries. Furthermore, the United States actively seeks to extradite hackers who are outside the USA to face trial in the United States. An example is the case of [Gery Shalon and Ziv Orenstein in 2015 who are Israeli hackers currently facing extradition](#) to the United States for hacking related crimes.

To avoid spending time in Prison, we suggest the following guidelines for penetration testing:

- When at all possible, limit your penetration testing to your own virtual machines on a private network.
- When pentesting systems on the Internet, only penetration test systems that belong to you or your company where you have been granted permission to penetration test them. Understand the limits of the access you have been granted and stay within those limits.
- If you decide to work as a professional penetration tester, work with a lawyer to draft an agreement that defines what systems you will be testing, how much access you have, where your access limits are and the time period during which you will have

access to perform the penetration test. Work with an experienced lawyer to draft the agreement and have your customer sign the agreement after they fully understand it.

- Never exceed your authorized access when performing a penetration test. This is a crime under the CFAA.
- Be extremely careful about accidentally testing systems that are outside the scope of the access you have been granted, outside the time window during which you have access, or that belong to someone else. For example, the default gateway router on the network you are testing may belong to an outside company, so don't pentest that unless you are sure you have been granted access to do so. When scanning a subnet, if you mistype the CIDR notation that defines a subnet (10.1.2.0/24 instead of 10.1.2.0/27) you could accidentally perform a much wider scan or exploit than you intended.

It may seem too easy to enter a few commands on the command line in Kali Linux and start a test. You may think your intentions are honorable, but don't underestimate the legal consequences of using penetration testing tools on networks and systems you have not been granted access to.

The above bullet points are a very basic introductory guide on how to stay safe when penetration testing. If you plan to become a professional penetration tester, work with a competent attorney to draft the necessary agreements and to understand the limits of the access you are given and how to protect yourself legally and operationally from prosecution.

Reconnaissance

In this step of your penetration test you will gather as much information about your target as possible. We will use a combination of recon scanning and **open source**

intelligence or OSINT.

nmap

The most important and fundamental recon scanning tool that pentesters use is nmap. The nmap tool has been around for a very long time and has grown in its reliability and capability. To use nmap, launch a terminal on your Kali Linux virtual machine. Then simply run the nmap command on the command line. With Kali Linux 2.0 you should have nmap version 7 included which is the newest version of nmap.

nmap is a port scanner that enumerates services on a host or hosts. To perform a fast simple port scan using nmap, you can run:

```
1 | nmap -sS 192.168.91.249
```

The above command will scan a single host using a 'half-open' scan which is very fast. The output will look similar to this:

```
root@mkal:~# nmap -sS 192.168.91.249

Starting Nmap 7.00 ( https://nmap.org ) at 2015-12-05 12:07 PST
Nmap scan report for test1.com (192.168.91.249)
Host is up (0.00037s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3128/tcp  open  squid-http
3306/tcp  open  mysql
MAC Address: 00:0C:29:89:45:64 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

As you can see, the above scan shows which ports are listening on our target IP and what services they are likely running. This host is running an SSH server, a web server, a microsoft service (it's either Windows or Linux running Samba), a proxy server on port 3128 and it has a MySQL database server.

All of these services are potential targets that we can use to gain entry to the host. We could examine these services individually, but instead nmap provides us an option that lets us dig deeper with very little work. Note that this is an aggressive option which will trip any half-decent intrusion detection system and will cause suspicious entries in log files.

To dig deeper with nmap, we use the -A option as follows on our target:

```
root@mkal:~# nmap -A 192.168.91.249

Starting Nmap 7.00 ( https://nmap.org ) at 2015-12-05 12:11 PST
Nmap scan report for test1.com (192.168.91.249)
Host is up (0.00042s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 1024 ef:e0:5c:17:40:b0:45:76:90:59:2e:9e:08:14:d4:3a (DSA)
|_ 2048 04:0b:67:31:52:e0:fc:1c:f3:e1:ca:3a:7e:a6:a5:c1 (RSA)
|_ 256 d6:11:ee:ff:03:c4:8f:26:de:bf:41:82:a6:eb:39:b4 (ECDSA)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu) PHP/5.5.9-1ubuntu4.14 mod_apreq2-20090110/2.8.0)
|_ http-generator: WordPress 4.3.1
|_ http-robots.txt: 1 disallowed entry
|_ /site/wp-admin/
|_ http-server-header: Apache/2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu4.14 mod_apreq2-20090110/2.8.0
|_ http-title: test 1 | Just another WordPress site
139/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: TEST1)
445/tcp   open  netbios-ssn Samba smbd 3.X (workgroup: TEST1)
3128/tcp  open  http-proxy   Squid http proxy 3.3.8
|_ http-server-header: squid/3.3.8
|_ http-title: ERROR: The requested URL could not be retrieved
3306/tcp  open  mysql        MySQL 5.5.46-0ubuntu0.14.04.2-log
| mysql-info:
|_ Protocol: 53
|_ Version: .5.46-0ubuntu0.14.04.2-log
|_ Thread ID: 2307
|_ Capabilities flags: 63487
|_ Some Capabilities: FoundRows, Support41Auth, Speaks41Protocol0ld, LongColumnFlag, IgnoreSiggpipes, SupportsLoadDataLocal, SupportsTransactions, DontAllowDatabaseTableColumn, IgnoreSpaceBeforeParenthesis, ConnectWithDatabase, InteractiveClient, SupportsCompression, ODBCClient, LongPassword, Speaks41ProtocolNew
|_ Status: Autocommit
|_ Salt: $g\v\[)W\I8z!1pe+BL9
MAC Address: 00:0C:29:89:45:64 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ nbstat: NetBIOS name: TEST1, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb-os-discovery:
|_ OS: Unix (Samba 4.1.6-Ubuntu)
|_ Computer name: test1
|_ NetBIOS computer name: TEST1
|_ Domain name: com
```

```
| FQDN: test1.com
|_ System time: 2015-12-05T12:11:19-08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

TRACEROUTE
HOP RTT      ADDRESS
1   0.42 ms  test1.com (192.168.91.249)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.71 seconds
```

As you can see, nmap has saved us a huge amount of time by doing a deep scan of the services running on our target. It has given us:

- The operating system: Ubuntu Linux
- The version of SSH: OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; protocol 2.0)
- The SSH host key which allows us to uniquely identify this host.
- The web server type and version: Apache httpd 2.4.7 ((Ubuntu) PHP/5.5.9-1ubuntu4.14 mod_apreq2-20090110/2.8.0)
- The content management system which is: WordPress 4.3.1
- The home page title: "test 1 | Just another WordPress site"
- The version of Samba (Windows file sharing) that the server is using including the workgroup name: Samba smbd 3.X (workgroup: TEST1)
- The type and version of the proxy server running on our target: Squid http proxy 3.3.8
- The type and version of the database server: MySQL 5.5.46-0ubuntu0.14.04.2-log

- It includes a huge amount of additional information including an OS fingerprint and some helpful information about the Samba server that may be exploitable. It even gives us the system time which is: 2015-12-05T12:11:19-08:00

Nmap is very powerful and will allow you to launch a scan on an entire network. To perform a -A scan on an entire subnet, you can run the following command:

```
1 | nmap -A 192.168.91.1-254
```

The above command will launch a deep -A scan on every host on a subnet. We don't recommend you do this because it is a very deep scan and scanning a subnet indiscriminately is not advisable. Instead, we suggest that when you perform a penetration test, you take a more surgical approach and scan individual hosts with individual nmap options. However, the above illustration gives you a good introduction to the power of nmap.

To use nmap's data as part of your reconnaissance, make a note of all services and versions discovered. If you see any services that should not be available, log them as high priority vulnerabilities that need to be fixed. In addition, log the services that you want to target specifically during your vulnerability scan.

To learn more about nmap options and capabilities, simply type 'man nmap' on the Kali Linux command line.

Open Source Intelligence (OSINT)

Open Source Intelligence or OSINT is the gathering of intelligence using publicly available sources of information. When doing a penetration test on your site it is important to evaluate what information is publicly available about you and/or your company.

The reason this is important is because social engineering attacks start by analyzing publicly available sources of information about a company so that an attacker can, for example, convince customer support personnel that they are an internal employee because they have what may appear to be internal knowledge.

When looking at OSINT about your target, keep a log of information that you find that should not be publicly disclosed. These are vulnerabilities and should be fixed.

Sources of OSINT

- **Whois.** Drop to the Kali terminal and run 'whois example.com' on your domain or 'whois ip-address' on your website IP address to see the ownership info.
- <http://www.socialmention.com/> is a search engine that searches across social media. Use this to search for brand names, usernames and more. Search for the usernames you see on your website, for example.
- <https://inteltechniques.com/links.html> has a variety of OSINT search tools. Also useful to do username searches and searches for identities of people who appear on your website.
- **recon-ng** is a useful command line OSINT analysis tool that is bundled with Kali Linux.
- **Maltego Chlorine** is an open source edition of Maltego's advanced OSINT analysis tool. Very powerful and complex. You need to register for an API key. Chlorine includes 'machines' that perform useful operations and display the results graphically. The 'company stalker' will find email addresses belonging to a company based on the domain name, for example. You are limited to 12 results with the free edition.
- **Netcraft** is a useful tool to find subdomains and what each website is running. Enter your website domain and hit search.

- **IP to Virtual Host** lookup tools like <https://hackertarget.com/reverse-ip-lookup/> can show you what websites an IP is running. Enter your site IP address and see what else is running on the same IP address.
- **Geo IP lookup tools** show you where an IP address is physically located. [Maxmind](#) is the best in the business. Enter your IP to see where your servers are physically located and information about the hosting provider.
- **The Dark Web.** Fire up your Tor browser and visit the following Dark Web search engines and search for your company name and other strings related to your pentest target, like usernames and product names.
 - <http://onion.city/>
 - <https://ahmia.fi/search/>
 - <http://thehiddenwiki.org/>
 - <http://xmh57jrznw6insl.onion/> (Torch aka The Tor Search)

Use the above sources of open source intelligence to compile lists of additional targets that you want to scan and analyze for vulnerabilities. You should also consider an unauthorized information disclosure or information leakage as a vulnerability and make a note of that.

Once you have completed your information gathering during the reconnaissance phase, you can move on to the scanning phase of your penetration test.

Scanning

What is Fuzzing?

Fuzzing is a technique where you send large amounts of random data to an application, in our case a web application, to try and discover vulnerabilities.

Some of the tools we will introduce here, like SQLMap, are very good at fuzzing applications to try and find a vulnerability. They will try a large number of attacks that use random data or data that is known to work in previous attacks to test if the application is vulnerable.

Fuzzing is a useful tool that can expose zero day vulnerabilities in applications. However, it is no substitute for understanding how an application works and trying exploits that are based on a human understanding of the application's functioning and where the developer may have made a mistake.

Often you can use a combination of human intelligence and fuzzing. For example, if you suspect a query string parameter is vulnerable to [SQL injection](#), you may choose to focus your fuzzing energy on that part of the application.

Fuzzing is used more frequently in black box testing, where the penetration tester does not have access to the application source code. The fuzzing target may be a compiled application or a remote web application that the analyst is accessing via the network. Either way, fuzzing lets you quickly test a large number of inputs to try and find unusual behavior that may lead to a vulnerability.

You should note that fuzzing is generally very noisy because it requires a large amount of data be sent to the target. It will invariably trip intrusion detection systems (IDS's) and may also create a high load on the target. So be careful where you direct your fuzzing attacks.

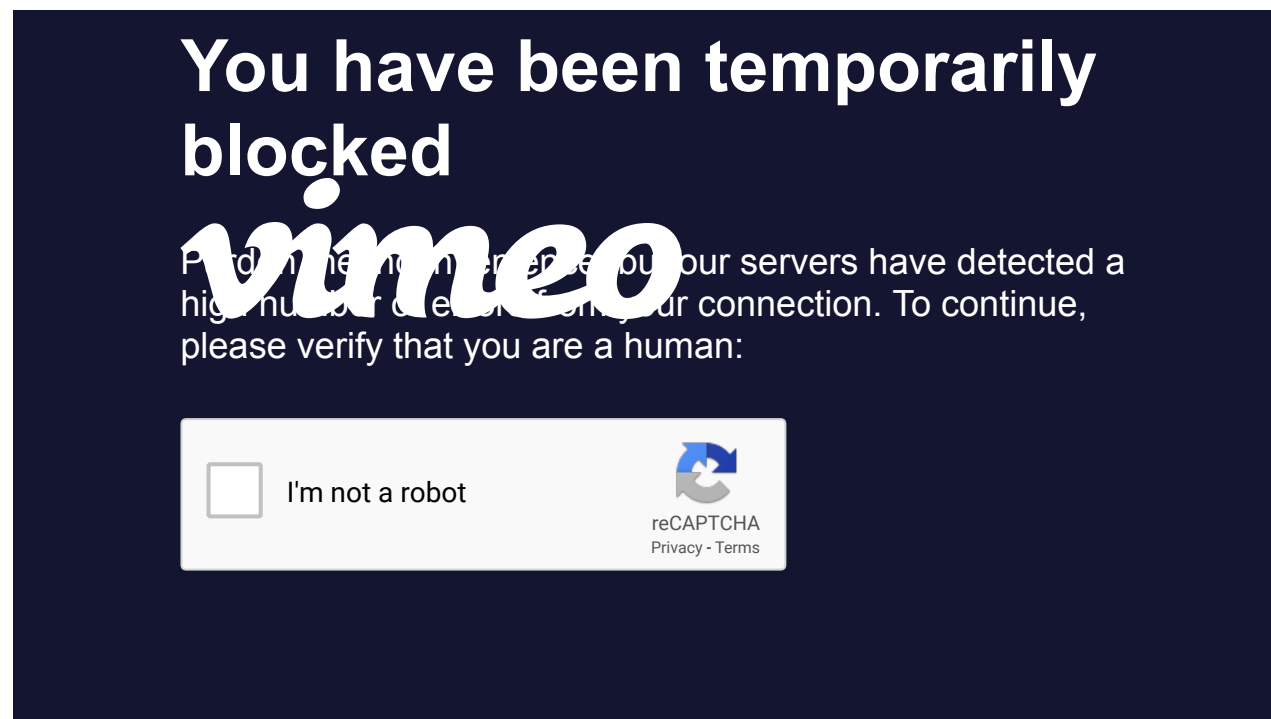
We will now describe several core penetration testing tools along with video demonstrations showing how to use them.

WPScan

WPScan is created by open source volunteers led by Ryan Dewhurst. It is written in Ruby and is a vulnerability scanner designed specifically for WordPress. It is very simple to use and can be quite effective, particularly on WordPress websites that have not updated either WordPress core, their themes or their plugins.

WPScan examines a website's core, theme and plugin versions and will alert you if it finds any vulnerable items. In most cases, if a website has upgraded all items to the newest versions, WPScan will likely not find any problems.

WPScan does not examine source code. It is a remote scanner and for that reason is well suited to black box testing. The following video provides a demonstration of how to use WPScan and how to ensure that it recognizes where your WordPress installation directory is.



Nikto

Nikto is a website vulnerability scanner. It takes much longer than WPScan, around 35 minutes in our video example below. Nikto includes many plugins and by default all plugins are enabled when you perform a scan. It will generate a significant amount of requests on your website during a scan.

In our example scan, Nikto discovers a test.php file and recognizes that the file prints the output from the phpinfo() function which contains information an attacker can use. During our demo, we didn't create this file for Nikto to find, it just found it automatically which is impressive.

Nikto does tend to generate a lot of false positives, so you will need to evaluate what Nikto finds to verify which are real vulnerabilities and which are false.

At the end of our video demonstration we show you how to control Nikto's output. By default it outputs to the console. So we recommend that you watch the entire video before launching a Nikto scan so that you understand how to write the output to a text file rather than to the console. You will probably want your Nikto output in a text file so that you can analyze it at your leisure and share it with colleagues.

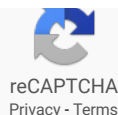
**You have been temporarily
blocked**

vineo

Unfortunately, our servers have detected a high number of connections from your IP address. To continue, please verify that you are a human:



I'm not a robot



Burp Suite

Burp Suite is a powerful tool to analyze and pentest web applications. It functions as a proxy server. You need to configure BurpSuite to listen on a port on your local address, usually IP 127.0.0.1 and port 8080. Then you need to configure your web browser to use Burp Suite as a proxy server.

Please refer to the [Burp Suite Getting Started Guide](#) for help getting your web browser configured and with the initial Burp Suite settings.

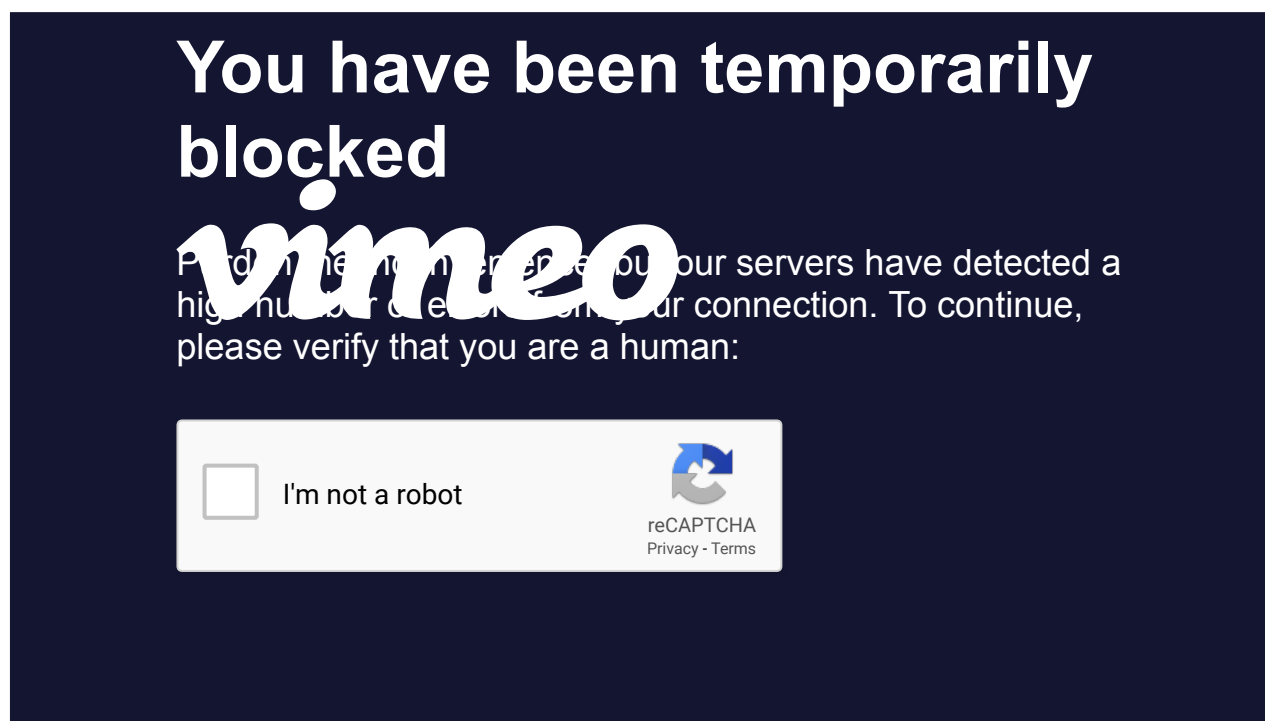
Once you have Burp Suite set up as a basic proxy, you probably are also going to want to be able to have Burp Suite analyze secure HTTPS requests. To do this, it needs to decrypt HTTPS traffic. To set this up, you should install Burp Suite's CA certificate as a trusted root in your web browser. In general, once you have Burp Suite installed you can visit <http://burp> and click the option to download the CA bundle. You will then install it in your browser and Burp Suite will be able to decrypt HTTPS traffic. [For browser specific instructions on how to do this, visit this page.](#)

Once you have Burp Suite installed, you can use it for the following:

- A full crawl of your target website including submitting forms.
- To edit browser requests at the raw header level before they are sent to a target site.
- To see browser requests and responses in detail.

- To perform an active vulnerability scan of a host or URL.
- To do static code analysis on your target site's javascript to find vulnerabilities.
- To attack a website with [XSS attacks](#), [SQL injection](#) attacks and more.

In the video demonstration below, we describe how Burp Suite is configured and demonstrate how to crawl a site, view HTTP traffic and how to launch a penetration test on our target site. We show how Burp Suite can detect an [XSS vulnerability](#) and the information it provides.



Burp Suite is an excellent tool for web application security analysis and penetration testing. It is also complex, but worth investing the time to understand it and add it to your toolbox of penetration testing tools.

SQLMap

SQLMap is a very easy to use and highly effective penetration testing tool. It focuses purely on finding and exploiting [SQL injection vulnerabilities](#). You can use SQL map to determine if a web application is injectable. For example, if you have an application where you suspect the 'id' query string parameter is injectable, you can use the following command:

```
1 | sqlmap -u http://example.com/?id=1 -p id
```

SQLMap will run a series of fuzzing tests on the 'id' parameter to try and determine if it is injectable. It won't assume the back-end database is MySQL and so will include tests for a variety of database engines. You can save some time by including the `--dbms` parameter:

```
1 | sqlmap --dbms=MySQL -u http://example.com/?id=1 -p id
```

During the scan, SQLMap will also give you the option to run extended tests on the DBMS that it detects. If you haven't yet found an injectable parameter, you might choose to run those tests.

The above commands will suffice to prove that an application has a SQL injection vulnerability. Once SQLMap has determined how to exploit a vulnerable website, it will store that information in a configuration file in your home directory.

If you would like to exploit the vulnerability, you can get a list of databases using the following command. Note that SQLMap won't re-fuzz the application to figure out how to inject it. It already knows how if you have done an initial run.

```
1 | sqlmap --dbms=MySQL -u http://example.com/?id=1 -p id --dbs
```

The `--dbs` parameter above will list the databases on the host. To select the 'test1' database and dump the `wp_users` table to the screen, you can use the following command:

```
1 | sqlmap --dbms=MySQL -u http://example.com/?id=1 -p id -D test1 --table=wp_users
```

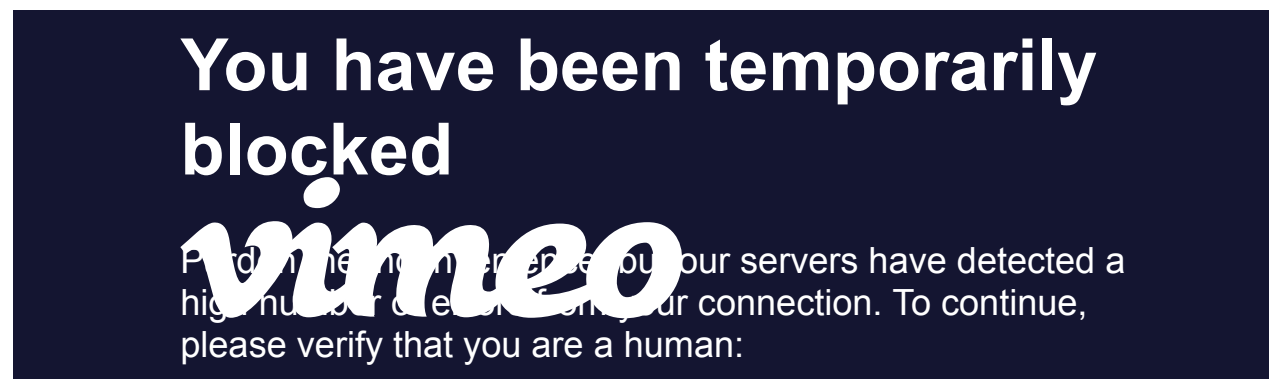
You will notice that SQLMap will ask you if it should store the hashes it finds in the table for further processing. You have that option if you want to run a cracking application on the hashes it finds.

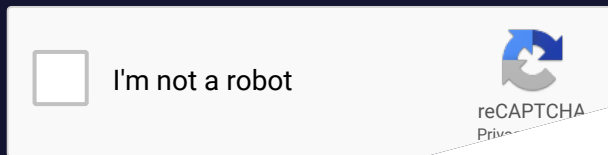
SQLMap can also go beyond the database. The following command will use the SQL injection vulnerability to make the database engine grab a file on the filesystem and send it back to you. In this case we will grab the `/etc/passwd` file:

```
1 | sqlmap --dbms=MySQL -u http://example.com/?id=1 -p id --file=/etc/passwd
```

The above command will download the file and save it in the SQLMap configuration directory and it will tell you where it saved the file.

In the video below we demonstrate some of the SQLMap features we have described above.





Exploitation

Exploitation is the act of using a vulnerability to gain access to a system or to increase the access level you already have. In general to successfully find vulnerabilities in your systems and close the security holes, you don't need to exploit the vulnerabilities.

You may be asked to verify the vulnerabilities you have found through exploitation on a test system. Or you may in rare cases be given permission to exploit a production system to demonstrate the seriousness of an issue.

When exploiting a vulnerability, be extremely careful because most vulnerabilities use applications in ways they were not designed to be used. 'Abuse' is a better term. [SQL injection](#), [XSS](#), [file upload vulnerabilities](#) and privilege escalation exploits can all cause a web application to become unstable.

Your exploitation may cause a denial-of-service in the application as the web server or database is placed under extreme load or core dumps. It may also cause a lot of garbage data to be logged in log files or inserted into the database in the case of [SQL injection](#) fuzzing and exploitation.

We recommend you avoid exploiting production systems if possible. The safest scenario if you are asked to demonstrate an exploit on a production application is to create a

virtual machine that emulates the live environment. Then exploit that virtual machine. This allows you to safely demonstrate your exploit. If you snapshot the VM before exploitation, you can revert it back to it's original configuration and re-exploit as many times as you would like without ill effects.

Here are a few tools for exploitation:

- SQLMap is very effective at uncovering [SQL injection](#) vulnerabilities. We have already covered how to exploit vulnerabilities with SQLMap above.
- Burp Suite is particularly good at discovering and exploiting XSS vulnerabilities. We introduced you to Burp Suite above.
- Metasploit is the gold standard in exploitation. We describe how to exploit a vulnerability with Metasploit below.

Exploiting Vulnerabilities with Metasploit

Metasploit is a command line tool that is included with Kali Linux. We don't recommend you use the version included with Kali because it is almost immediately out of date when a new version of Kali is released. Instead, grab the newest version from Github.

To install metasploit, we prefer using the [Metasploit Omnibus edition which you can find on GitHub](#).

You can install Metasploit Omnibus using the following command:

```
1 | curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus
```

If that doesn't work for you, take a look at the github page for Omnibus which has detailed instructions.


Once you have Metasploit installed, run 'msfupdate' and then run 'msfconsole' to start Metasploit. When it starts you should see a cute ascii graphic and prompt:

```
Call trans opt: received. 2-10-98 13:24:18 REC:Loc

Trace program: running

    wake up, Neo...
    the matrix has you
    follow the white rabbit.

    knock, knock, Neo.



    http://metasploit.pro

    =[ metasploit v4.11.5-dev-f4d35116bd9577eacd40662b879642cd447ab58b9 ]
+ -- --=[ 1588 exploits - 870 auxiliary - 253 post           ]
+ -- --=[ 434 payloads - 37 encoders - 0 nops              ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

gef >
```

Metasploit is very easy to use. You can get started with the following commands. Comments are in square brackets:

- search wpsshop [Will find you an exploit for the wpsshop WordPress plugin]
- use exploit/unix/webapp/wp_wpsshop_ecommerce_file_upload [Uses the exploit module]
- show options [shows you what parameters you need to set to run the exploit]
- set RHOST 192.168.100.200 [Replace the IP with your target]
- set TARGETURI /install_dir/ [replace install_dir with your target's base URL, or with /]
- set VHOST example.com [replace example.com with your target host]

- exploit [Or you can just type 'run']

The above commands illustrate the basics of using metasploit. The process is to find an exploit module, set the options and run the exploit.

Metasploit also includes several commands that start with 'db_' which are very useful database functions. They require integration with PostgreSQL, but once you have that configured you'll notice your exploit searches are faster and you can use features like db_nmap to probe a network and store the results in a database.

There are many other exploit tools available including many commercial tools, some of which are very expensive (over \$30,000). We have covered some of the most popular and powerful tools and mastering them will give you a solid foundation in vulnerability scanning and exploitation.

Mitigation

Mitigation is the final stage of vulnerability management. It is the process whereby you close the security holes you have found. During your reconnaissance and scanning you took note of the vulnerabilities you found and prioritized them.

Now that you have completed your data gathering you need to combine all the vulnerabilities you found with each individual tool into a single document. We recommend you use a spreadsheet. You will need to add a severity to each vulnerability and we suggest assigning a number from 1 to 5, where 5 is the most severe.

Then sort your vulnerabilities from highest to lowest. Once you have done that, you will have a clear picture of your current 'security posture', or how secure your network is.

Now it's time to get to work. Start working to fix each vulnerability starting at the top of your spreadsheet and working your way down.

Next to each vulnerability in your spreadsheet you should add a note describing how the vulnerability was fixed or why it is no longer an issue. For example, it may be too expensive or complex to fix a certain severe vulnerability, so you may choose to remove the affected host from the network.

Once you have completed this process, you will have completed your penetration test and mitigation.

Conclusion

We have given you a short and very intense introduction to penetration testing. We have introduced you to pentesting tools used by the best in the industry. Metasploit and Burp Suite are industry standards among pentesters.

Using the information provided and the approach we have outlined in this document you are now equipped to come up with a penetration testing strategy, assemble your tools, perform reconnaissance, perform scans, exploit where needed and mitigate the vulnerabilities that you find. Good luck!

Did you enjoy this post? Share it!



COMPANY

[ABOUT](#)

[CAREERS](#)

[CONTACT](#)

[EVENTS](#)

[SECURITY](#)

RESOURCES

[BLOG](#)

[CENTRAL](#)

[HELP](#)

[LEARN](#)

[MAILING LIST](#)

[PODCAST](#)

[WORDFENCE WEEKLY](#)

SERVICES

[SITE CLEANING](#)

[SECURITY AUDIT](#)

SOCIAL

