 3. Introduction to Nikto

# ﻿https://assets.tryhackme.com/additional/web-enumeration-redux/nikto.png

# Introduction to Nikto

Initially released in 2001, Nikto has made leaps and bounds over the years and has proven to be a very popular vulnerability scanner due to being both open-source nature and feature-rich. Nikto is capable of performing an assessment on all types of webservers (and isn't application-specific such as WPScan.). Nikto can be used to discover possible vulnerabilities including:

* Sensitive files
* Outdated servers and programs (i.e. [vulnerable web server installs](https://httpd.apache.org/security/vulnerabilities_24.html))
* Common server and software misconfigurations (Directory indexing, cgi scripts, x-ss protections)

### Installing Nikto

Thankfully for us, Nikto comes pre-installed on the latest versions of penetration testing systems such as Kali Linux and Parrot. If you are using an older version of Kali Linux (such as 2019) for example, Nikto is in the apt repository, so can be installed by a simple sudo apt update && sudo apt install nikto

﻿Installing Nikto on other operating systems such as Ubuntu or Debian involves extra steps. Whilst the TryHackMe AttackBox comes pre-installed with Nikto, you can follow the [developer's installation guide](https://cirt.net/nikto2-docs/installation.html#id2780292) for your local environment.

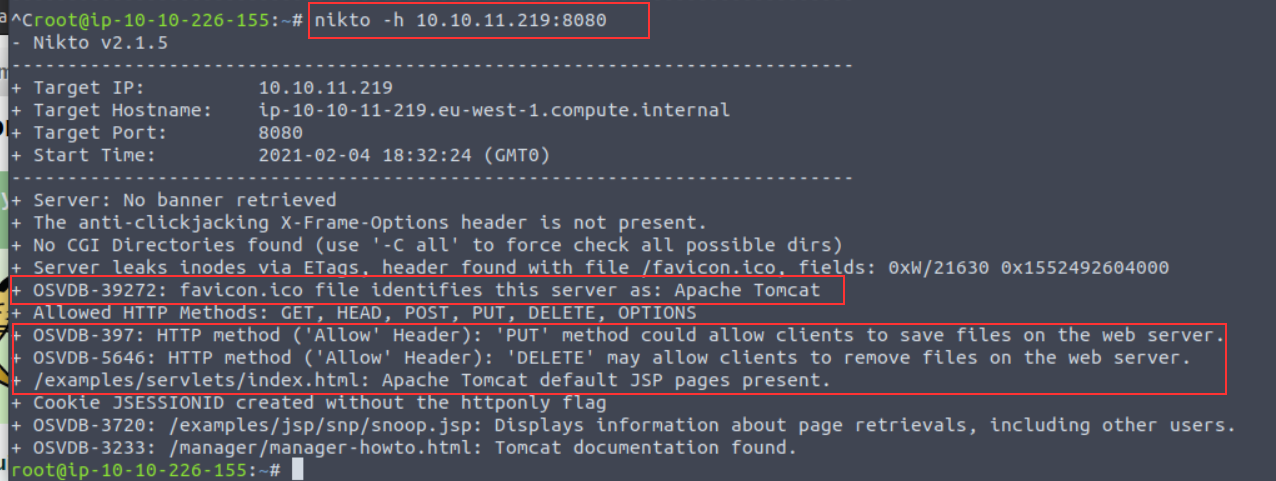
In the next task, we will explore some common syntax and features of Nikto!

 3.1. Nikto Modes

**Basic Scanning**

The most basic scan can be performed by using the -h flag and providing an IP address or domain name as an argument. This scan type will retrieve the headers advertised by the webserver or application (I.e. Apache2, Apache Tomcat, Jenkins or JBoss) and will look for any sensitive files or directories (i.e. login.php, /admin/, etc)

An example of this is the following: nikto -h vulnerable\_ip



Note a few interesting things are given to us in this example:

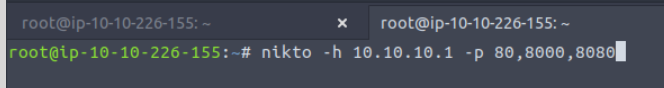
* Nikto has identified that the application is Apache Tomcat using the favicon and the presence of "*/examples/servlets/index.html*" which is the location for the default Apache Tomcat application.
* HTTP Methods "*PUT*" and "*DELETE*" can be performed by clients - we may be able to leverage these to exploit the application by uploading or deleting files.

**Scanning Multiple Hosts & Ports**

Nikto is extensive in the sense that we can provide multiple arguments in a way that's similar to tools such as Nmap. In fact, so much so, we can take input directly from an Nmap scan to scan a host range. By scanning a subnet, we can look for hosts across an entire network range. We must instruct Nmap to output a scan into a format that is friendly for Nikto to read using Nmap's  -oG  flags

For example, we can scan 172.16.0.0/24 (subnet mask 255.255.255.0, resulting in 254 possible hosts) with Nmap (using the default web port of 80) and parse the output to Nikto like so: nmap -p80 172.16.0.0/24 -oG - | nikto -h -

There are not many circumstances where you would use this other than when you have gained access to a network. A much more common scenario will be scanning multiple ports on one specific host. We can do this by using the-p flag and providing a list of port numbers delimited by a comma - such as the following: nikto -h 10.10.10.1 -p 80,8000,8080



**Introduction to Plugins**

Plugins further extend the capabilities of Nikto. Using information gathered from our basic scans, we can pick and choose plugins that are appropriate to our target. You can use the --list-plugins flag with Nikto to list the plugins or [view the whole list in an easier to read format online](https://github.com/sullo/nikto/wiki/Plugin-list).

Some interesting plugins include:

|  |  |
| --- | --- |
| Plugin Name | Description |
| apacheusers | Attempt to enumerate Apache HTTP Authentication Users |
| cgi | Look for CGI scripts that we may be able to exploit |
| robots | Analyse the robots.txt file which dictates what files/folders we are able to navigate to |
| dir\_traversal | Attempt to use a directory traversal attack (i.e. LFI) to look for system files such as /etc/passwd on Linux (http://ip\_address/application.php?view=../../../../../../../etc/passwd) |

We can specify the plugin we wish to use by using the-Plugin argument and the name of the plugin we wish to use...For example, to use the "*apacheuser*" plugin, our Nikto scan would look like so: nikto -h 10.10.10.1 -Plugin apacheuser

https://assets.tryhackme.com/additional/web-enumeration-redux/nikto/plugin-scan.png

**Verbosing our Scan**

We can increase the verbosity of our Nikto scan by providing the following arguments with the-Display flag. Unless specified, the output given by Nikto is not the entire output, as it can sometimes be irrelevant (but that isn't always the case!)

|  |  |  |
| --- | --- | --- |
| Argument | Description | Reasons for Use |
| 1 | Show any redirects that are given by the web server. | Web servers may want to relocate us to a specific file or directory, so we will need to adjust our scan accordingly for this. |
| 2 | Show any cookies received | Applications often use cookies as a means of storing data. For example, web servers use sessions, where e-commerce sites may store products in your basket as these cookies. Credentials can also be stored in cookies. |
| E | Output any errors | This will be useful for debugging if your scan is not returning the results that you expect! |

**Tuning Your Scan for Vulnerability Searching**

Nikto has several categories of vulnerabilities that we can specify our scan to enumerate and test for. The following list is not extensive and only include the ones that you may commonly use. We can use the -Tuningflag and provide a value in our Nikto scan:

|  |  |  |
| --- | --- | --- |
| Category Name | Description | Tuning Option |
| File Upload | Search for anything on the web server that may permit us to upload a file. This could be used to upload a reverse shell for an application to execute. | 0 |
| Misconfigurations / Default Files | Search for common files that are sensitive (and shouldn't be accessible such as configuration files) on the web server. | 2 |
| Information Disclosure | Gather information about the web server or application (i.e. verison numbers, HTTP headers, or any information that may be useful to leverage in our attack later) | 3 |
| Injection | Search for possible locations in which we can perform some kind of injection attack such as XSS or HTML | 4 |
| Command Execution | Search for anything that permits us to execute OS commands (such as to spawn a shell) | 8 |
| SQL Injection | Look for applications that have URL parameters that are vulnerable to SQL Injection | 9 |

**Saving Your Findings**

Rather than working with the output on the terminal, we can instead, just dump it directly into a file for further analysis - making our lives much easier!

Nikto is capable of putting to a few file formats including:

* Text File
* HTML report

We can use the -o argument (short for -Output) and provide both a filename and compatible extension. We *can*specify the format (-f) specifically, but Nikto is smart enough to use the extension we provide in the-o argument to adjust the output accordingly.

For example, let's scan a web server and output this to "*report.html*": nikto -h http://ip\_address -o report.html

