

# Nikto

## *Linux Laboratory*

### Overview

Nikto is a comprehensive web server scanner designed to fingerprint and test web servers for a variety of possible weaknesses including potentially dangerous files and out-of-date versions of applications and libraries. Nikto can make use of the LibWhisker anti-IDS routines developed by Rain Forest Puppy (*which is probably not the name his momma gave him*). In this lab you will install and update Nikto, perform web server tests and analyze the logs left on the scanned web server(s).

### Laboratory Objective

At the conclusion of this laboratory the student should be able to

1. Understand the purpose of Nikto
2. Install and configure Nikto
3. Utilize Nikto as a scanning tool
4. Identify evidence, on a web server, of a Nikto scan
5. Identify and understand various IDS avoidance techniques

### Laboratory Assumptions

1. The testing (*attacking*) machine is running RedHat Linux 9.0. Its IP address is 192.168.2.200 and it has a subnet mask of 255.255.255.0.
2. There are two tested (*victim*) machines.
  - ⑩ One is running RedHat Linux 7.1 and Apache Web Server. No patches or updates have been installed. Its IP address is 192.168.2.3 and it has a subnet mask of 255.255.255.0.
  - ⑩ The other machine is running Microsoft Windows 2000 Advanced Server with IIS 5.0 installed. No patches or updates have been installed. Its IP address is 192.168.2.1 and it has a subnet mask of 255.255.255.0.

### Class Preparation

1. The student should review the Nikto documentation found at [http://www.cirt.net/nikto/README\\_nikto.html](http://www.cirt.net/nikto/README_nikto.html).
2. The student should review the IDS avoidance techniques used by LibWhisker. These techniques are described in a paper by Rain Forest Puppy which may be accessed at either <http://packetstormsecurity.nl/papers/IDS/whiskerids.html> or <http://www.ussrback.com/docs/papers/IDS/whiskerids.html>.

Estimated Completion Time: 60 Minutes

## **1. Installing and Updating Nikto**

The Nikto source files may either be supplied by your instructor or you may download them from the Internet.

To download the files,

- ⑩ Use Mozilla to access <http://www.cirt.net/code/nikto.shtml>.
- ⑩ Under the heading Download, click the current version of Nikto (at the time of this writing it is Version 1.34).
- ⑩ Click OK to download the file.
- ⑩ Save it to the /root directory.

The files are in a gzipped tar file such as nikto-current.tar.gz. Use the following command to decompress the file:

```
tar -zxvf nikto-current.tar.gz
```

The files will be decompressed into a series of directories under /root/nikto-1.34. Type the following command to change to the Nikto directory:

```
cd /root/nikto-1.34
```

The config.txt file found in the /root/nikto-1.34 directory is used to configure certain program defaults. You will need to edit this file. Find the line that reads as follows:

```
# PLUGINDIR=/usr/local/nikto/plugins
```

Remove the # from the beginning of the line and edit the text to read as follows:

```
PLUGINDIR=/root/nikto-1.34/plugins
```

From the /root/nikto-1.34 directory, type

```
./nikto.pl -V
```

You should get a listing of all the plug-ins currently available for use. The listing will look something like figure 1.

```
[root@rh9 nikto-1.34]# ./nikto.pl -V
-----
Nikto Versions
-----
File                               Version      Last Mod
-----
Nikto main                          1.34
LibWhisker                          1.7
nikto_apacheusers.plugin           1.02        05.23.2003
nikto_core.plugin                  1.29        07.25.2004
nikto_headers.plugin               1.09        11.07.2003
```

## *Nikto Laboratory*

nikto_httptoptions.plugin	1.05	07.22.2003
nikto_msgs.plugin	1.04	03.26.2003
nikto_mutate.plugin	1.07	07.02.2004
nikto_outdated.plugin	1.13	09.06.2004
nikto_passfiles.plugin	1.03	07.24.2004
nikto_plugin_order.txt	1.04	05.27.2003
nikto_realms.plugin	1.02	09.07.2004
nikto_robots.plugin	1.05	05.27.2003
nikto_user_enum_apache.plugin	1.02	06.17.2003
nikto_user_enum_cgiwrap.plugin	1.01	06.17.2003
outdated.db	1.109	06.30.2004
realms.db	1.003	08.24.2003
scan_database.db	1.189	09.06.2004
server_msgs.db	1.094	01.18.2003
servers.db	1.006	12.07.2003

-----  
[root@rh9 nikto-1.34]#

Most of the plug-ins appear to be somewhat out of date. To update Nikto from cert.net's web site, type

```
./nikto.pl -update
```

If there are any updates available, they will be automatically installed.

## **2. Web Server Testing.**

The most basic scan only looks at port 80 of the targeted machine. From the /root/nikto-1.34 directory, enter the following:

```
./nikto.pl -h 192.168.2.3
```

The -h parameter must precede the IP address or name of the server you are testing.

Below is a portion of the output generated by the scan. Notice that Nikto has identified the machine platform (RedHat Linux), the web server software and version (Apache 1.3.19) and the versions of ssl, PHP and Perl that are in use. Additionally, it has gone to lengths to identify the HTTP methods that the server is allowing and to point out which exploits to which various outdated modules are susceptible.

```
+Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) mod_ssl/2.8.1 OpenSSL/0.9.6 DAV/1.0.2  
PHP/4.0.4pl1 mod_perl/1.24_01
```

```
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH,  
PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK,  
UNLOCK, TRACE
```

```
+HTTP method 'PUT' method may allow clients to save files on the web server.
```

```
+HTTP method 'CONNECT' may allow server to proxy client requests
```

```
+HTTP method 'DELETE' may allow clients to remove files on the web server.
```

```
+ Apache/1.3.19 appears to be outdated (current is at least Apache/2.0.50). Apache  
1.3.31 is still maintained and considered  
secure.
```

```
+ mod_ssl/2.8.1 appears to be outdated (current is at least 2.8.19) (may depend on  
server version)
```

```
+ OpenSSL/0.9.6 appears to be outdated (current is at least 0.9.7d) (may depend on  
server version)
```

```
+ DAV/1.0.2 appears to be outdated (current is at least 2)
```

```
+ PHP/4.0.4pl1 appears to be outdated (current is at least 5.0.1)
```

```
+ mod_perl/1.24_01 appears to be outdated (current is at least 1.99_14)
```

```
+ PHP/4.0.4 - PHP 4.1.1 is vulnerable to remote exploits and must be upgraded.
```

```
+ Apache/1.3.19 (Unix) (Red-Hat/Linux) mod_ssl/2.8.1 OpenSSL/0.9.6 DAV/1.0.2
```

## Nikto Laboratory

PHP/4.0.4pl1 mod\_perl/1.24\_01 - Apache 1.x up 1.2.34 are vulnerable to a remote DoS and possible code execution. CAN-2002-0392.  
+ 2.8.1 OpenSSL/0.9.6 DAV/1.0.2 PHP/4.0.4pl1 mod\_perl/1.24\_01 - TelCondex Simpleserver 2.13.31027 Build 3289 and below allow directory traversal with '/../' entries.  
+ mod\_ssl/2.8.1 - mod\_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell (difficult to exploit). CAN-2002-0082.  
+ PHP/4.0.4pl1 mod\_perl/1.24\_01 - PHP below 4.3.3 may allow local attackers to safe mode and gain access to unauthorized files. BID-8203.

List at least three major flaws that were detected on the Apache server.

---

---

---

---

---

A scan of a Windows 2000 Advanced Server with IIS proves just as interesting.

```
+ Server: Microsoft-IIS/5.0
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK,
UNLOCK
+ HTTP method 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get
directory listings if indexing is allowed but a default page exists.
+ HTTP method 'SEARCH' may be used to get directory listings if Index Server is
running.
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled.
+ Microsoft-IIS/5.0 appears to be outdated (4.0 for NT 4, 5.0 for Win2k)
+ / - Appears to be a default IIS install. (GET)
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACE)
+ / - TRACK option ('TRACE' alias) appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACK)
+ /scripts - Redirects to http://192.168.2.1/scripts/, Remote scripts directory is
browsable.
+ /blahb.idq - Reveals physical path. To fix: Preferences -> Home directory ->
Application & check 'Check if file exists' for the ISAPI mappings. MS01-033. (GET)
+ /xxxxx.htw - Server may be vulnerable to a Webhits.dll arbitrary file retrieval.
Ensure Q252463i, Q252463a or Q251170 is installed. MS00-006. (GET)
+ /NULL.printer - Internet Printing (IPP) is enabled. Some versions have a buffer
overflow/DoS in Windows 2000 which allows
remote attackers to gain admin privileges via a long print request that is passed to
the extension through IIS 5.0. Disabling the .printer mapping is recommended. EEYE-
AD20010501, CVE-2001-0241, MS01-023, CA-2001-10, BID 2674 (GET)
+ /scripts/samples/search/qfullhit.htw - Server may be vulnerable to a Webhits.dll
arbitrary file retrieval. MS00-006. (GET)
+ /scripts/samples/search/qsumrhit.htw - Server may be vulnerable to a Webhits.dll
arbitrary file retrieval. MS00-006. (GET)
+ /_vti_bin/fpcount.exe - Frontpage counter CGI has been found. FP Server version 97
allows Remote users to execute arbitrary system commands, though a vulnerability in
this version could not be confirmed. CAN-1999-1376. BID-2252. (GET)
+ /_vti_bin/shtml.dll/_vti_rpc?method=server+version%3a4%2e0%2e2%2e2611 - Gives info
about server settings. CAN-2000-0413, CAN-2000-0709, CAN-2000-0710, BID-1608, BID-
1174. (POST)
+ /_vti_bin/shtml.exe - Attackers may be able to crash FrontPage by requesting a DOS
device, like shtml.exe/aux.htm -- a DoS
was not attempted. CAN-2000-0413, CAN-2000-0709, CAN-2000-0710, BID-1608, BID-1174.
(GET)
+ /_vti_bin/shtml.exe/_vti_rpc - FrontPage may be installed. (GET)
+ /_vti_bin/shtml.exe/_vti_rpc?method=server+version%3a4%2e0%2e2%2e2611 - Gives info
about server settings. CAN-2000-0413, CAN-2000-0709, CAN-2000-0710, BID-1608, BID-
1174. (POST)
```

Not only does Nikto correctly identify the platform and the web server, in this case it also discovers FrontPage extensions running and identifies the sample scripts that are on the server along with the methods by which they may be compromised.

## Nikto Laboratory

These two scans have “trusted” the scanned server to return a “Server:” string that correctly identifies the platform and web server software. Since it is possible for some web servers to send back fake information, the -g parameter will force Nikto to do a full scan of the machine instead of trusting it. Try the following command.

```
./nikto.pl -h 192.168.2.3 -g
```

The three scans you have done thus far have been rather “plain jane”. They have only tested the target on port 80. The -p parameter must be used to scan using ports other than 80. Try the following:

```
./nikto.pl -h 192.168.2.1 -p 443 -g
```

This command will scan 192.168.2.1 on port 443 only. Were any additional (or different) potential problems detected?

---

---

---

---

---

---

---

---

---

---

You can also scan multiple ports. Examples include the following:

```
./nikto.pl -h 192.168.2.1 -p 80,443,8080 -g
```

```
./nikto.pl -h 192.168.2.1 -p 80-100 -g
```

The first example scans 192.168.2.1 on ports 80, 443 and 8080 only. The second example scans all ports between 80 and 100.

### 3. Detecting Nikto

Nikto is not an incredibly stealthy scanner. Examine the access logs of the Apache server. On the RedHat Linux 7.1 machine, the access log is located at /etc/httpd/logs/access\_log. A portion of an Apache access log appears below.

```
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/nph-exploitcanget.cgi HTTP/1.0" 404 284 "-" "Mozilla/4.75"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/parse-file HTTP/1.0" 404 272 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/quikstore.cfg HTTP/1.0" 404 275 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/register.cgi HTTP/1.0" 404 274 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/simplestquest.cgi HTTP/1.0" 404 279 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/simplestmail.cgi HTTP/1.0" 404 278 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/statusconfig.pl HTTP/1.0" 404 277 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/sws/manager.pl HTTP/1.0" 404 276 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/texis/phine HTTP/1.0" 404 273 "-" "Mozilla/4.75 (Nikto/1.34 )"
192.168.2.200-- [01/Oct/2004:08:54:36-0600] "GET /cgi-bin/utm/admin HTTP/1.0" 404 271 "-" "Mozilla/4.75 (Nikto/1.34 )"

```



## *Nikto Laboratory*

intrusion, automatically notify an administrator that an intrusion is in progress and/or automatically take action to “cut off” the intruder. For an IDS to detect intrusions, it must be able to recognize the “tell-tale” signs that an intrusion is in progress. Some IDSs are better than others at noticing “interesting” network traffic. The idea behind IDS evasion is to make the intrusion not look like what the IDS expects an intrusion to look like. You must realize that some IDS systems will be vulnerable to one or more intrusion methods and that some will not be “tricked” by any of the common methods. Only very old and/or very bad IDS systems will be fooled by all the methods.

Nikto uses LibWhisker for IDS evasion. LibWhisker includes nine different evasion methods. These methods are listed below and are described in detail by LibWhisker author Rain Forest Puppy in a paper that may be accessed at either <http://packetstormsecurity.nl/papers/IDS/whiskerids.html> or <http://www.ussrback.com/docs/papers/IDS/whiskerids.html>.

- 1 Random URI encoding (non-UTF8)
- 2 Add directory self-reference ./
- 3 Premature URL ending
- 4 Prepend long random string to request
- 5 Fake parameters to files
- 6 TAB as request spacer instead of spaces
- 7 Random case sensitivity
- 8 Use Windows directory separator \ instead of /
- 9 Session splicing

Enter the following command.

```
./nikto.pl -h 192.168.2.3 -g -e 6
```

This command will scan 192.168.2.3 using evasion technique 6. After viewing the scan output, access the access\_log and error\_log of the RedHat 7.1 machine that was just scanned.

Are there any differences in the logs compared to when you originally viewed the logs in step 3? If so, what are the differences and why do they exist?

---

---

---

---

---

---

---

---

---

---



*Nikto Laboratory*

<http://httpd.apache.org/docs-2.0/misc/tutorials.html>

**Securing Microsoft IIS**

<http://www.microsoft.com/technet/Security/chklist/iis50srg.msp>

## **Instructors Notes**

This laboratory may be completed using multiple machines or virtual machines installed with VMWARE or like software. This laboratory assumes that three machines exist as is outlined below.

1. The testing (*attacking*) machine is running RedHat Linux 9.0. It's IP address is 192.168.2.200 and it has a subnet mask of 255.255.255.0. This machine should have Perl and OpenSSL installed. For simplicity you may wish to download or copy the Nikto tar.gz to the /root directory.
2. There are two tested (*victim*) machines.
  - ⑩ One is running RedHat Linux 7.1 and Apache Web Server. No patches or updates have been installed. It's IP address is 192.168.2.3 and the subnet mask is 255.255.255.0. It is assumed that the log and configuration files are located in the directory structure under /etc/httpd.
  - ⑩ The other machine is running Microsoft Windows 2000 Advanced Server with IIS 5.0 installed. No patches or updates have been installed. It's IP address is 192.168.2.1 and it has a subnet mask of 255.255.255.0. The exercise assumes that the operating system is installed on drive C: and that the systemroot directory is WINNT.