All Options

Below are all of the Nikto command line options and explanations. A brief version of this text is available by running Nikto with the -h (-help) option.

`-ask`

Whether to ask about submitting updates: yes (ask about each-- the default), no (don't ask, just send), auto (don't ask, just send).

`-Cgidirs`

Scan these CGI directories. Special words "none" or "all" may be used to scan all CGI directories or none, (respectively). A literal value for a CGI directory such as "/cgi-test/" may be specified (must include trailing slash). If this is option is not specified, all CGI directories listed in nikto.conf will be tested.

`-config`

Specify an alternative config file to use instead of the nikto.conf file located in the install directory.

`-dbcheck`

Check the scan databases for syntax errors.

`-Display`

Control the output that Nikto shows. See Chapter 5 for detailed information on these options. Use the reference number or letter to specify the type. Multiple may be used:

1 - Show redirects

2 - Show cookies received

3 - Show all 200/OK responses

4 - Show URLs which require authentication

D - Debug Output

E - Display all HTTP errors

P - Print progress to STDOUT

V - Verbose Output

`-evasion`

Specify the LibWhisker encoding/evasion technique to use (see the LibWhisker docs for detailed information on these). Note that these are not likely to actually bypass a modern IDS system, but may be useful for other purposes. Use the reference number to specify the type, multiple may be used:

1 - Random URI encoding (non-UTF8)

2 - Directory self-reference (/./)

3 - Premature URL ending

4 - Prepend long random string

5 - Fake parameter

6 - TAB as request spacer

7 - Change the case of the URL

8 - Use Windows directory separator (\)

A - Use a carriage return (0x0d) as a request spacer

B - Use binary value 0x0b as a request spacer

`-findonly`

Only discover the HTTP(S) ports, do not perform a security scan. This will attempt to connect with HTTP or HTTPS, and report the Server header. Note that as of version 2.1.4, -findonly has been deprecated and simply sets '-Plugins "@@NONE"' which will override any command line or config file settings for -Plugins.

`-Format`

Save the output file specified with -o (-output) option in this format. If not specified, the default will be taken from the file extension specified in the -output option. Valid formats are:

csv - a comma-seperated list

htm - an HTML report

msf - log to Metasploit

txt - a text report

xml - an XML report

`-host`

Host(s) to target. Can be an IP address, hostname or text file of hosts. A single dash (-) maybe used for stdin. Can also parse nmap -oG style output

`-Help`

Display extended help information.

`-id`

ID and password to use for host Basic host authentication. Format is "id:password".

`-IgnoreCode`

Ignore these HTTP codes as negative responses (always). Format is "302,301".

`-list-plugins`

Will list all plugins that Nikto can run against targets and then will exit without performing a scan. These can be tuned for a session using the *`-Plugins`* option.

The output format is:

Plugin name

full name - description

Written by author, Copyright (C) copyright

-maxtime

Maximum execution time per host, in seconds. Accepts minutes and hours such that all of these are one hour: 3600s, 60m, 1h

-mutate

Specify mutation technique. A mutation will cause Nikto to combine tests or attempt to guess values. These techniques may cause a tremendous amount of tests to be launched against the target. Use the reference number to specify the type, multiple may be used:

1 - Test all files with all root directories

2 - Guess for password file names

3 - Enumerate user names via Apache (/~user type requests)

4 - Enumerate user names via cgiwrap (/cgi-bin/cgiwrap/~user type requests)

5 - Attempt to brute force sub-domain names, assume that the host name is the parent domain

6 - Attempt to guess directory names from the supplied dictionary file

-mutate-options

Provide extra information for mutates, e.g. a dictionary file

-nolookup

Do not perform name lookups on IP addresses.

-nocache

Disable response cache

-nointeractive

Disable interactive features

-nossl

Do not use SSL to connect to the server.

-no404

Disable 404 (file not found) checking. This will reduce the total number of requests made to the webserver and may be preferable when checking a server over a slow link, or an embedded device. This will generally lead to more false positives being discovered.

-output

Write output to the file specified. The format used will be taken from the file extension. This can be over-riden by using the -Format option (e.g. to write text files with a different extenstion. Existing files will have new information appended.

A single dot (.) may be specified for the output file name, in which case the file name will be automatically generated based on the target being tested. Note that the -Format option is required when this is used. The scheme is: nikto_HOSTNAME_PORT_TIMESTAMP.FORMAT

For '-Format msf' the output option takes special meaning. It should contain the password and location of the Metasploit RPC service. For example, it may look like: '-o msf: <password>@http://localhost:55553/RPC2'

`-Plugins`

Select which plugins will be run on the specified targets. A semi-colon separated list should be provided which lists the names of the plugins. The names can be found by using -list-plugins.

There are two special entries: @@ALL, which specifies all plugins shall be run and @@NONE, which specifies no plugins shall be run. The default is @@DEFAULT

`-port`

TCP port(s) to target. To test more than one port on the same host, specify the list of ports in the -p (-port) option. Ports can be specified as a range (i.e., 80-90), or as a comma-delimited list, (i.e., 80,88,90). If not specified, port 80 is used.

`-Pause`

Seconds (integer or floating point) to delay between each test.

`-root`

Prepend the value specified to the beginning of every request. This is useful to test applications or web servers which have all of their files under a certain directory.

`-ssl`

Only test SSL on the ports specified. Using this option will dramatically speed up requests to HTTPS ports, since otherwise the HTTP request will have to timeout first.

`-Save`

Save request/response of findings to this directory. Files are plain text and will contain the raw request/response as well as JSON strings for each. Use a "." to auto-generate a directory name for each target. These saved items can be replayed by using the included replay.pl script, which can route items through a proxy.

`-timeout`

Seconds to wait before timing out a request. Default timeout is 10 seconds.

`-Tuning`

Tuning options will control the test that Nikto will use against a target. By default, all tests are performed. If any options are specified, only those tests will be performed. If the "x" option is used, it will reverse the logic and exclude only those tests. Use the reference number or letter to specify the type, multiple may be used:

0 - File Upload

1 - Interesting File / Seen in logs

2 - Misconfiguration / Default File

3 - Information Disclosure

4 - Injection (XSS/Script/HTML)

5 - Remote File Retrieval - Inside Web Root

6 - Denial of Service

7 - Remote File Retrieval - Server Wide

8 - Command Execution / Remote Shell

9 - SQL Injection

a - Authentication Bypass

b - Software Identification

c - Remote Source Inclusion

x - Reverse Tuning Options (i.e., include all except specified)

The given string will be parsed from left to right, any x characters will apply to all characters to the right of the character.

`-Userdbs`

Load user defined databases instead of standard databases. User defined databases follow the same syntax as the standard files, but are prefixed with a 'u', e.g., 'udb_tests'

all - Disable all standard databases and load only user databases

tests - Disable db_tests and load udb_tests. All other databases are loaded normally.

`-until`

Run until the specified time or duration, then pause.

Durations in hours, minutes or seconds, like: 1h, 60m, 3600s

Times like "mm dd hh:mm:ss" (mm, dd, ss optional): 12 1 22:30:00

`-update`

Update the plugins and databases directly from cirt.net.

`-useproxy`

Use the HTTP proxy defined in the configuration file. The proxy may also be directly set as an argument.

`-Version`

Display the Nikto software, plugin and database versions.

`-vhost`

Specify the Host header to be sent to the target.

Mutation Techniques

A mutation will cause Nikto to combine tests or attempt to guess values. These techniques may cause a tremendous amount of tests to be launched against the target. Use the reference number to specify the type, multiple may be combined.

> **Note**
> The *-mutate* and *-mutate-options* parameters have been deprecated in nikto 2.1.2. Plugin selections, using the *-Plugin* parameter, should be used instead. Nikto will automagically turn a mutate option into the appropriate selection string.

1. Test all files with all root directories. This takes each test and splits it into a list of files and directories. A scan list is then created by combining each file with each directory.
2. Guess for password file names. Takes a list of common password file names (such as "passwd", "pass", "password") and file extensions ("txt", "pwd", "bak", etc.) and builds a list of files to check for.
3. Enumerate user names via Apache (/~user type requests). Exploit a misconfiguration with Apache UserDir setups which allows valid user names to be discovered. This will attempt to brute-force guess user names. A file of known users can also be supplied by supplying the file name in the *-mutate-options* parameter.
4. Enumerate user names via cgiwrap (/cgi-bin/cgiwrap/~user type requests). Exploit a flaw in cgiwrap which allows valid user names to be discovered. This will attempt to brute-force guess user names. A file of known users can also be supplied by supplying the file name in the *-mutate-options* parameter.
5. Attempt to brute force sub-domain names. This will attempt to brute force know domain names, it will assume the given host (without a www) is the parent domain.
6. Attempt to brute directory names. This is the only mutate option that requires a file to be passed in the *-mutate-options* parameter. It will use the given file to attempt to guess directory names. Lists of common directories may be found in the OWASP DirBuster project.

Display

By default only some basic information about the target and vulnerabilities is shown. Using the *-Display* parameter can produce more information for debugging issues.

- 1 - Show redirects. This will display all requests which elicit a "redirect" response from the server.
- 2 - Show cookies received. This will display all cookies that were sent by the remote host.
- 3 - Show all 200/OK responses. This will show all responses which elicit an "okay" (200) response from the server. This could be useful for debugging.
- 4 - Show URLs which require authentication. This will show all responses which elicit an "authorization required" header.
- D - Debug Output. Show debug output, which shows the verbose output and extra information such as variable content.
- E - Display all HTTP errors. Show details for any HTTP error encountered.
- P - Print progress to STDOUT. Show status report to STDOUT during testing (interval set in

nikto.conf).

- V - Verbose Output. Show verbose output, which typically shows where Nikto is during program execution.
- E - Error Output. Display all HTTP and communications errors, which show a lot of output on some servers.

Scan Tuning

Scan tuning can be used to decrease the number of tests performed against a target. By specifying the type of test to include or exclude, faster, focused testing can be completed. This is useful in situations where the presence of certain file types are undesired -- such as XSS or simply "interesting" files.

Test types can be controlled at an individual level by specifying their identifier to the *-T* (*-Tuning*) option. In the default mode, if *-T* is invoked only the test type(s) specified will be executed. For example, only the tests for "Remote file retrieval" and "Command execution" can performed against the target:

```
perl nikto.pl -h 192.168.0.1 -T 58
```

If an "x" is passed to *-T* then this will negate all tests of types following the x. This is useful where a test may check several different types of exploit. For example:

```
perl nikto.pl -h 192.168.0.1 -T 58xb
```

The valid tuning options are:

- 0 - File Upload. Exploits which allow a file to be uploaded to the target server.
- 1 - Interesting File / Seen in logs. An unknown but suspicious file or attack that has been seen in web server logs (note: if you have information regarding any of these attacks, please contact CIRT, Inc.).
- 2 - Misconfiguration / Default File. Default files or files which have been misconfigured in some manner. This could be documentation, or a resource which should be password protected.
- 3 - Information Disclosure. A resource which reveals information about the target. This could be a file system path or account name.
- 4 - Injection (XSS/Script/HTML). Any manner of injection, including cross site scripting (XSS) or content (HTML). This does not include command injection.
- 5 - Remote File Retrieval - Inside Web Root. Resource allows remote users to retrieve unauthorized files from within the web server's root directory.
- 6 - Denial of Service. Resource allows a denial of service against the target application, web server or host (note: no intentional DoS attacks are attempted).
- 7 - Remote File Retrieval - Server Wide. Resource allows remote users to retrieve unauthorized files from anywhere on the target.
- 8 - Command Execution / Remote Shell. Resource allows the user to execute a system command or spawn a remote shell.

9 - SQL Injection. Any type of attack which allows SQL to be executed against a database.

- a - Authentication Bypass. Allows client to access a resource it should not be allowed to access.
- b - Software Identification. Installed software or program could be positively identified.
- c - Remote source inclusion. Software allows remote inclusion of source code.
- x - Reverse Tuning Options. Perform exclusion of the specified tuning type instead of inclusion of the specified tuning type.

Replay Saved Requests

When using the Save functionality (-Save), findings requests are saved in text files. While these files contain human readable text, they also contain JSON representations of the request and response. This JSON request can be replayed by using the "replay.pl" script.

The replay.pl reads and parses a saved file via the -file option, and can optionally run the request through a proxy, such as Burp. This will allow further exploration of vulnerabilities in a program better suited to replay and resend attacks.

```
$ ./replay -file savedir_host_80_2012-09-11-00-07-42/host_80_2012-09-11_002114.txt -
proxy localhost:8080
-----------------------------------------  Info
Request to:     http://host:80/manual/
Test ID:        002114
OSVDB ID:       3092
Message:        OSVDB-3092: /manual/: Web server manual found.
-----------------------------------------  Response
date: Tue, 11 Sep 2012 04:14:20 GMT
server: Apache/2.2.21 (Unix) mod_ssl/2.2.21 OpenSSL/0.9.8r DAV/2 PHP/5.3.10 with
Suhosin-Patch
content-location: index.html.en
vary: negotiate,accept-language,accept-charset
tcn: choice
last-modified: Tue, 06 Sep 2011 02:41:54 GMT
etag: "15eb9df-1f07-4ac3cc53d8080;15eb9dd-32b-4ac3cc53d8080"
accept-ranges: bytes
content-length: 7943
keep-alive: timeout=5, max=100
connection: Keep-Alive
content-type: text/html
content-language: en

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"><head><!--
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
              This file is generated from xml source: DO NOT EDIT
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      -->
<title>Apache HTTP Server Version 2.2
Documentation - Apache HTTP Server</title>
...
```

Plugin selection

From Nikto 2.1.2 plugins can be selected on an individual basis and may have parameters passed to them.

A plugin selection string may be passed on the command line through the *-Plugin* parameter. It consists of a semi-colon separated list of plugin names with option parameters placed in brackets. In simple form a plugin statement is like:

plugin-name[(*parameter name*[:*parameter value*][,*other parameters*])]

For example we can do:

```
tests(report:500,verbose)
```

Which will set the parameters report to a value of 500 and verbose to a value of 1. The parameters and plugin names can be found be running:

```
./nikto.pl -list-plugins
```

This also means that we deprecate the mutate options and replace them with parameters passed to plugins, so the mutate options now internally translate to:

1. tests(all)
2. tests(passfiles)
3. apacheusers(enumerate,home[,dictionary:dict.txt])
4. apacheusers(enumerate,cgiwrap[,dictionary:dict.txt])
5. subdomain
6. dictionary(dictionary:dict.txt)

Macros for commonly run plugin sets can also be defined in nikto.conf, the default ones are:

```
@@MUTATE=dictionary;subdomain
@@DEFAULT=@@ALL;-@@MUTATE;tests(report:500)
```

These are expanded by using -list-plugins and can be overridden through -Plugins.

Altogether this can allow a customised set of plugins that may need to be run for a specific circumstance. For example if a normal test bought up that the server was vulnerable to the apache Expect header XSS attack and we want to run a test just to see that it is vulnerable by adding debugging, we can run:

```
nikto.pl -host target.txt -Plugins "apache_expect_xss(verbose,debug)"
```

And then manually check the output to see whether it was truly vulnerable.

It should be noted that reports are also plugins, so if you need to customize the plugin string and want an output, include the report plugin:

```
nikto.pl -host targets.txt -Plugins
"apacheusers(enumerate,dictionary:users.txt);report_xml" -output apacheusers.xml
```
