CSC427 NIKTO
ERTAN, ADITYA, KELVIN

It can check certain vulnerabilities, use a proxy, can activate it's evasion techniques, single host scan, multiple port scan, can be mutated, can fine tune the scanning capacity and the best part is... wait for it... wait for it... Voila! It can be integrated with Nessus!!!.

1. OVERVIEW

Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

Examine a web server to find potential problems and security vulnerabilities, including:

- Server and software misconfigurations
- Default files and programs
- Insecure files and programs
- Outdated servers and programs

Nikto is built on LibWhisker2 (by RFP) and can run on any platform which has a Perl environment. It supports SSL, proxies, host authentication, attack encoding and more. It can be updated automatically from the command-line, and supports the optional submission of updated version data back to the maintainers.

2. USE CASES

2A. FOR THE ATTACKER

Nikto can be used by an attacker to find various default and insecure files, configurations, and programs on the web server.If the vulnerability is severe enough, It can lead to your whole web server being compromised!In the demo, we will be looking at how the attacker can use Nikto to find vulnerabilities in the web application and exploit them

2B. FOR THE DEFENDER

The tool enables security professionals to scan either one port or a range of ports for Web servers, which provides the additional benefit of finding rogue servers that weren’t set up by the enterprise. Once a server is found, Nikto displays any known vulnerabilities from the open sourced vulnerability database; it can also scan for over 65,000 potentially dangerous files and 1,250 outdated server software versions. With this level of visibility, enterprises will be able to measure the insecurity of their Web servers and take concrete steps toward patching and updating systems.

3. OPTIONS/FEATURES

-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.

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.

-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

-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

-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.

-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'

-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

-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.

Detailed explanation for some of the more important options:

Mutate:

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.

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.

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.

4. DEMO

4A. SETUP

We use two VMs for this demo, the first one has a vulnerable web service, FourFours, the other being Kali Linux

Setting up FourFours:
The zip file for FourFours is located at:
dh2020pc01.utm.utoronto.ca:/virtual/arnold/Ubuntu804Server_2015a2.zip

If you are doing the demo from the school lab, you can just copy the zip file into your virtual directory and unzip:
scp
$USER@dh2020pc01.utm.utoronto.ca:/virtual/arnold/Ubuntu804Server_2015a2.zip
/virtual/$USER
cd /virtual/$USER
unzip Ubuntu804Server_2015a2.zip

Open vmplayer, load the .vmx file, Before starting the VM, Edit network adapter, select VMNET 8.

fire up the VM, select "I moved it"

Once booted, login with arnold/bpbthisisvulnerable2015

sudo su

cp /etc/network/interfaces.dhcp /etc/network/interfaces

/etc/init.d/networking restart

ifconfig, get the VM, browse to it from your Host machine, and if all goes well, you'll see the vulnerable fourfours Web App
ex (on firefox): http://192.168.82.134/fourFours

if ip doesn't show up in step 8, perhaps that lab machine network interface is in use by another logged in user - temporary work around is to physically reboot the lab machine.

Steps to setup Kali VM

1. cd /virtual/$USER
2. unzip /virtual/arnold/kali.zip
3. Open vmplayer, Open the VM from the filessystem, Edit network adapter, select VMNET 8
4. Log in is root/toor

Setup for fourFours and Kali VMs cited from http://infosec.csprojects.me/week7.html

4B. EXECUTION

Once you have both the VMs running, go to Kali and open up the fourFours web application
To start using Nikto:
Applications -> Web Applications -> Web Vulnerability Scanners -> nikto
This will open up a terminal with the man page for nikto
To view a more detailed man page, do nikto -Help
Some of the more important options that are valuable to us for computer security purposes are: -Cgidirs, -Display, -evasion, -mutate, -Tuning, -output, -host, -port,