Nmap Scripting Engine (NSE)

The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts (using the Lua programming language) to automate a wide variety of networking tasks. Those scripts are executed in parallel with the speed and efficiency you expect from Nmap. Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.

Tasks we had in mind when creating the system include network discovery, more sophisticated version detection, vulnerability detection. NSE can even be used for vulnerability exploitation.

To reflect those different uses and to simplify the choice of which scripts to run, each script contains a field associating it with one or more categories. Currently defined categories are `auth`, `broadcast`, `default`. `discovery`, `dos`, `exploit`, `external`, `fuzzer`, `intrusive`, `malware`, `safe`, `version`, and `vuln`. These are all described in the section called "Script Categories".

Scripts are not run in a sandbox and thus could accidentally or maliciously damage your system or invade your privacy. Never run scripts from third parties unless you trust the authors or have carefully audited the scripts yourself.

The Nmap Scripting Engine is described in detail in Chapter 9, Nmap Scripting Engine and is controlled by the following options:

`-sC`

Performs a script scan using the default set of scripts. It is equivalent to `--script=default`. Some of the scripts in this category are considered intrusive and should not be run against a target network without permission.

`--script <filename>|<category>|<directory>/|<expression>[,...]`

Runs a script scan using the comma-separated list of filenames, script categories, and directories. Each element in the list may also be a Boolean expression describing a more complex set of scripts. Each element is interpreted first as an expression, then as a category, and finally as a file or directory name.

There are two special features for advanced users only. One is to prefix script names and expressions with `+` to force them to run even if they normally wouldn't (e.g. the relevant service wasn't detected on the target port). The other is that the argument `all` may be used to specify every

script in Nmap's database. Be cautious with this because NSE contains dangerous scripts such as exploits, brute force authentication crackers, and denial of service attacks.

File and directory names may be relative or absolute. Absolute names are used directly. Relative paths are looked for in the `scripts` of each of the following places until found:

`--datadir`
`$NMAPDIR`
`~/.nmap` (not searched on Windows)
`<APPDATA>\nmap` (only on Windows)
the directory containing the `nmap` executable
the directory containing the `nmap` executable, followed by `../share/nmap` (not searched on Windows)
`NMAPDATADIR` (not searched on Windows)
the current directory.

When a directory name ending in `/` is given, Nmap loads every file in the directory whose name ends with `.nse`. All other files are ignored and directories are not searched recursively. When a filename is given, it does not have to have the `.nse` extension; it will be added automatically if necessary.

Nmap scripts are stored in a `scripts` subdirectory of the Nmap data directory by default (see Chapter 14, Understanding and Customizing Nmap Data Files). For efficiency, scripts are indexed in a database stored in `scripts/script.db`, which lists the category or categories in which each script belongs.

When referring to scripts from `script.db` by name, you can use a shell-style '`*`' wildcard.

nmap --script "http-*"

Loads all scripts whose name starts with `http-`, such as `http-auth` and `http-open-proxy`. The argument to `--script` had to be in quotes to protect the wildcard from the shell.

More complicated script selection can be done using the `and`, `or`, and `not` operators to build Boolean expressions. The operators have the same precedence as in Lua: `not` is the highest, followed by `and` and then `or`. You can alter precedence by using parentheses. Because expressions contain space characters it is necessary to quote them.

nmap --script "not intrusive"

Loads every script except for those in the `intrusive` category.

nmap --script "default or safe"

This is functionally equivalent to **nmap --script "default,safe"**. It loads all scripts that are in the `default` category or the `safe` category or both.

nmap --script "default and safe"

Loads those scripts that are in both the `default` and `safe` categories.

nmap --script "(default or safe or intrusive) and not http-*"

Loads scripts in the `default`, `safe`, or `intrusive` categories, except for those whose names start with `http-`.

`--script-args <n1>=<v1>,<n2>={<n3>=<v3>},<n4>={<v4>,<v5>}`

Lets you provide arguments to NSE scripts. Arguments are a comma-separated list of `name=value` pairs. Names and values may be strings not containing whitespace or the characters '`{`', '`}`', '`=`', or '`,`'. To include one of these characters in a string, enclose the string in single or double quotes. Within a quoted string, '`\`' escapes a quote. A backslash is only used to escape quotation marks in this special case; in all other cases a backslash is interpreted literally. Values may also be tables enclosed in `{}`, just as in Lua. A table may contain simple string values or more name-value pairs, including nested tables. Many scripts qualify their arguments with the script name, as in `xmpp-info.server_name`. You may use that full qualified version to affect just the specified script, or you may pass the unqualified version (`server_name` in this case) to affect all scripts using that argument name. A script will first check for its fully qualified argument name (the name specified in its documentation) before it accepts an unqualified argument name. A complex example of script arguments is `--script-args 'user=foo,pass=",{}=bar",whois={whodb=nofollow+ripe},xmpp-info.server_name=localhost'`. The online NSE Documentation Portal at https://nmap.org/nsedoc/ lists the arguments that each script accepts.

`--script-args-file <filename>`

Lets you load arguments to NSE scripts from a file. Any arguments on the command line supersede ones in the file. The file can be an absolute path, or a path relative to Nmap's usual search path (NMAPDIR, etc.) Arguments can be comma-separated or newline-separated, but otherwise follow the same rules as for `--script-args`, without requiring special quoting and escaping, since they are not parsed by the shell.

`--script-help <filename>|<category>|<directory>|<expression>|all[,...]`

Shows help about scripts. For each script matching the given specification, Nmap prints the script name, its categories, and its description. The specifications are the same as those accepted by `--script`; so for example if you want help about the `ftp-anon` script, you would run **nmap --script-help ftp-anon**. In addition to getting help for individual scripts, you can use this as a preview of what scripts will be run for a specification, for example with **nmap --script-help default**.

`--script-trace`

This option does what `--packet-trace` does, just one ISO layer higher. If this option is specified all incoming and outgoing communication performed by a script is printed. The displayed information includes the communication protocol, the source, the target and the transmitted data. If more than 5% of all transmitted data is not printable, then the trace output is in a hex dump format. Specifying `--packet-trace` enables script tracing too.

`--script-updatedb`

This option updates the script database found in `scripts/script.db` which is used by Nmap to determine the available default scripts and categories. It is only necessary to update the database if you have added or removed NSE scripts from the default `scripts` directory or if you have changed the categories of any script. This option is generally used by itself: **nmap --script-updatedb**.


OS Detection


Timing and Performance

[Nmap | Sec Tools | Mailing Lists | Site News | About/Contact | Advertising | Privacy]