# Nikto — Professional Cheat Sheet

**One-page professional reference** for **Nikto** — open-source web server scanner. Quick commands, common checks, tuning, output formats, integration tips, and OPSEC notes for pentesters and web auditors.

---

## 1) At-a-glance

- **Tool:** `nikto` — scanner focused on identifying known web server vulnerabilities, misconfigurations, outdated server software, and dangerous files/CGIs.
- **Primary uses:** Quickly surface a wide range of known issues (missing security headers, default files, dangerous scripts, server version disclosures, outdated modules).
- **Caveat:** Nikto is noisy and may cause false positives; it performs many requests and can trigger WAF/IDS. Use only with permission.

---

## 2) Install / update

```
# Kali (often preinstalled)
sudo apt update && sudo apt install nikto

# From source (latest/plugins)
git clone https://github.com/sullo/nikto.git
cd nikto/program
# run via: perl nikto.pl -h target

# Update plugin/db
perl nikto.pl -update
```

---

## 3) Basic usage

```
# Simple scan
nikto -h http://example.com

# Scan with host:port
nikto -h example.com:8443

# Use HTTPS explicitly
nikto -h https://example.com
```

## 4) Important flags & options

- `-h <host|url>` : target host or URL (required).
- `-p <port>` : set port.
- `-ssl` : force SSL (useful for non-standard HTTPS ports).
- `-Tuning x` : tuning options to focus scans; combine values (see list below).
- `-id <user>:<pass>` : test basic auth credentials.
- `-Cgidirs <csv>` : specify CGI directories to test.
- `-Plugins <p1,p2>` : enable specific plugins.
- `-evasion <num>` : evasion techniques (URL encoding, case, fake headers; may bypass WAF but increases noise).
- `-useragent <UA>` : set custom user agent.
- `-timeout <secs>` : request timeout.
- `-no404` : don't test for 404 comparisons.
- `-Nikto <option>` : pass raw Nikto options.

### Output formats

- `-o <file>` : write output to file.
- `-Format <format>` : output format: `txt` (default), `csv`, `xml`, `html`, `nbe` (Nessus DB), `json`.
- `-Display V` : verbose display level; `-Display 1` normal, higher values show more.

---

## 5) Tuning values (common)

Nikto's `-Tuning` accepts a bitmask covering groups of tests. Common values: - `1` -> File/dir checks - `2` -> Misconfiguration checks - `4` -> Server version checks - `8` -> Authentication tests - `16` -> Information disclosure - `32` -> IIS specific - `64` -> Web application tests (CGI, scripts) - `-Tuning 1,2,4` or `-Tuning 123` to combine.

Example: `-Tuning 1,2,4` for core file/misconfig/version checks (less noisy than full scan).

---

## 6) Practical examples

### 6.1 Quick surface scan (less noisy)

```
nikto -h http://10.10.10.5 -Tuning 1,2,4 -o nikto_quick.txt -Format txt
```

### 6.2 Full aggressive scan

```
nikto -h https://example.com -Tuning 1,2,4,8,16,32,64 -evasion 1 -o
full_scan.html -Format html
```

### 6.3 Scan with custom user agent and proxy (Burp)

```
nikto -h http://app.local -useragent 'Mozilla/5.0 (Nikto Test)' -useproxy
http://127.0.0.1:8080 -o proxy_scan.json -Format json
```

### 6.4 Scan specific virtual host / host header

```
nikto -h http://10.0.0.5 -Host example.com
# or
nikto -h http://10.0.0.5 -head "Host: example.com"
```

### 6.5 Use Burp request file approach (via proxy)

Send traffic through Burp with `-useproxy` or capture and craft requests manually for targeted testing.

---

## 7) Plugins & db tuning

- Nikto uses plugin files for checks; update with `-update`.
- Check `plugins/` directory for enabled tests and customize if you need to add or remove checks.
- For large engagements, create custom plugin files to test organization-specific paths and patterns.

---

## 8) Performance, stealth & safety

- **Start small:** use `-Tuning` to avoid noisy/slow tests.
- **Respect rate limits:** add `-timeout` and use `-pause` (if available via wrapper) between requests.
- **WAFs & IDS:** use `-evasion` carefully; discuss with client before evasive tests.
- **Avoid destructive checks:** some plugins attempt dangerous actions — review plugin behavior first.

---

## 9) Integration & reporting

- **Combine with Nikto output parsers:** export to `csv`, `xml`, or `json` for ingestion into reporting tools or SIEM.
- **Chain with other tools:** run Nikto after initial discovery (nmap/dirbuster) to check discovered hosts/paths.

- **Correlation:** cross-reference Nikto findings with Burp scans and manual verification to reduce false positives.

---

## 10) Troubleshooting & common pitfalls

- **False positives:** verify findings manually — many tests are heuristic.
- **Proxy issues:** ensure `-useproxy` points to your interception proxy and that SSL interception is configured for HTTPS.
- **Encoding problems:** some servers require specific encodings; use `-evasion` or craft custom requests.
- **Plugin outdated:** run `perl nikto.pl -update` and keep plugins current.

---

## 11) Reporting checklist

1. Tuning level used, date/time, target IP/Host header, and user agent.
2. Exact Nikto command and version.
3. List of findings with reproduction steps and risk rating.
4. Evidence screenshots or HTTP request/response samples.
5. Remediation recommendations (disable directory listing, remove default files, update server modules, add security headers).

---

## 12) One-liners (copy-paste)

```
# Quick test
nikto -h http://target -Tuning 1,2,4 -o quick.txt -Format txt

# Full scan, output HTML
nikto -h https://target -Tuning 1,2,4,8,16,32,64 -o report.html -Format html

# Through Burp
nikto -h http://target -useproxy http://127.0.0.1:8080 -o burp.json -Format json

# Update DB/plugins
perl nikto.pl -update
```

---

*Nikto is a quick reconnaissance tool — treat results as leads for manual verification and deeper testing. Only use on systems you are authorized to test.*