**SearchSploit on Kali Linux**

Author: Subhan Ahmed

# 1. Overview

SearchSploit is the command‑line interface to the Exploit‑DB repository. It lets you search, review, and copy proof‑of‑concept exploits and shellcodes locally—ideal for air‑gapped or restricted environments.

**Ethical Use**

Use SearchSploit only on systems you own or have explicit permission to test. Many payloads are destructive or grant system access.

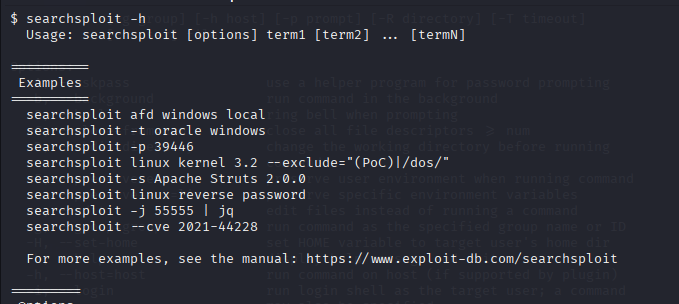
# 2. Installation on Kali

SearchSploit ships with the Kali package exploitdb. If it’s missing, install it:

sudo apt update && sudo apt -y install exploitdb exploitdb-papers exploitdb-bin-sploits

Update the local Exploit‑DB copy at any time:

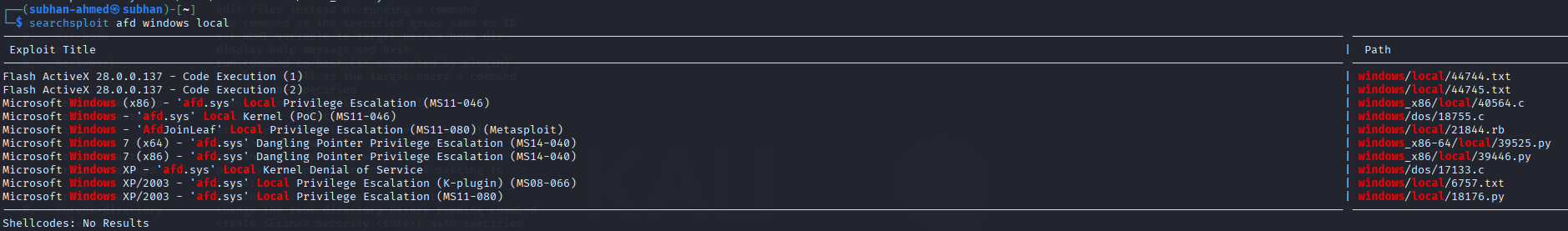
searchsploit –u



# 3. Quick Start

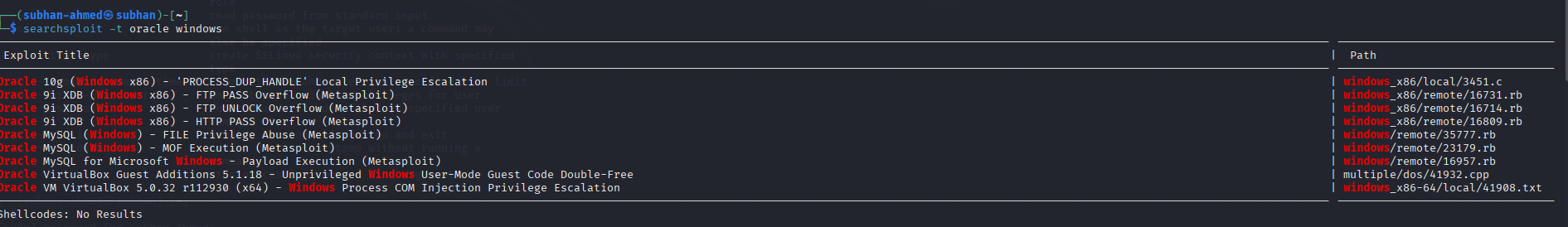
Basic search with multiple terms (AND logic):

searchsploit afd windows local



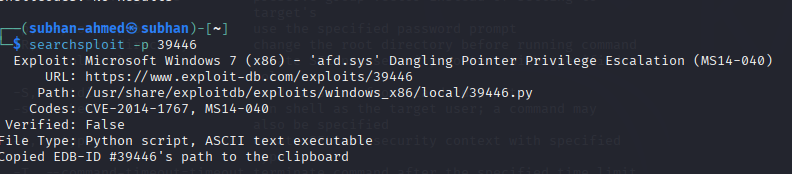
Search only within exploit titles:

searchsploit -t oracle windows



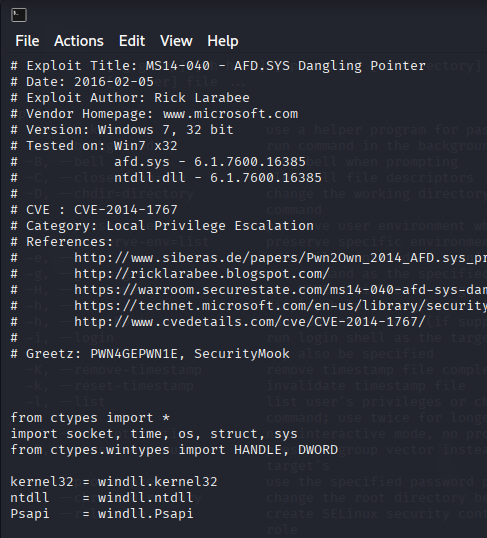
Show full local path (also copied to clipboard when possible):

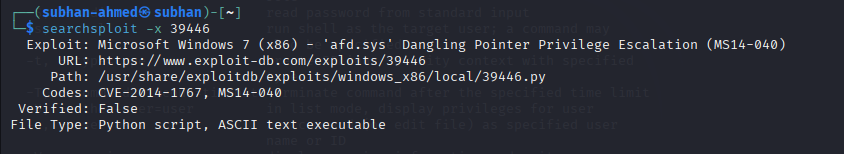
searchsploit -p 39446



Open an exploit for quick viewing (uses $PAGER):

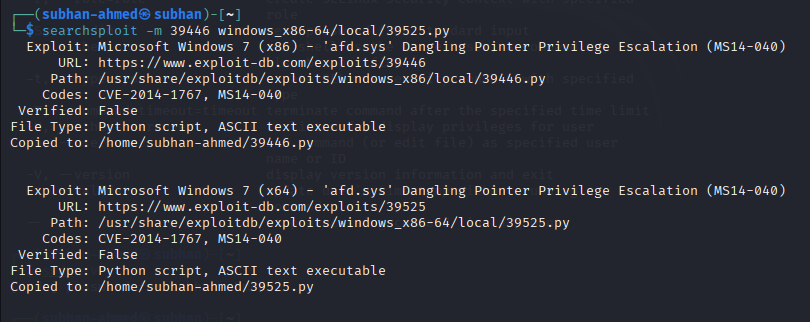
searchsploit -x 39446





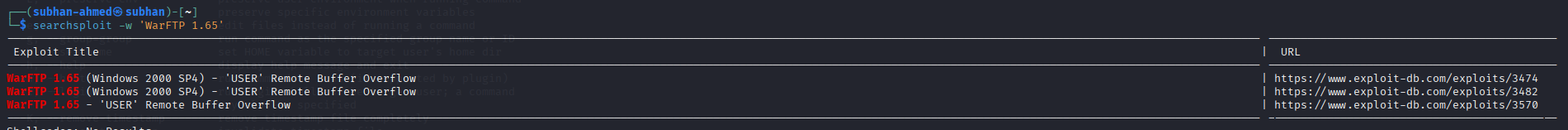
Copy one or more exploits into your working directory:

searchsploit -m 39446 windows\_x86-64/local/39525.py



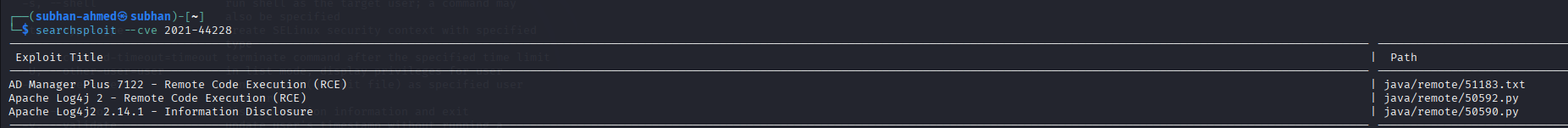
Get clickable web URLs instead of local paths:

searchsploit -w 'WarFTP 1.65'



Search by CVE:

searchsploit --cve 2021-44228



JSON output (great for scripting):

searchsploit -j 55555 | jq

# 4. Command Reference (All Options)

SearchSploit uses an AND operator, not OR. The more terms you add, the fewer results you’ll see. Order is not important.

## 4.1 Search Terms

|  |  |  |
| --- | --- | --- |
| Option | What it does | Example |
| -c, --case [term] | Case‑sensitive search (default is case‑insensitive). | searchsploit -c Windows afd |
| -e, --exact [term] | Exact & ordered match on the title (implies -t). | searchsploit -e "WordPress 5.8.1" |
| -s, --strict | Strict version matching; disables fuzzy range matching. | searchsploit -s "Apache Struts 2.0.0" |
| -t, --title [term] | Search only the exploit title (default: title + path). | searchsploit -t oracle windows |
| --exclude="term1|term2" | Exclude terms (chain with |). Useful to remove PoC/DoS. | searchsploit kernel 3.2 --exclude="(PoC)|/dos/" |
| --cve [ID] | Find by CVE identifier. | searchsploit --cve 2021-44228 |

## 4.2 Output

|  |  |  |
| --- | --- | --- |
| Option | What it does | Example |
| -j, --json [term] | JSON output for programmatic use. | searchsploit -j 39446 | jq |
| -o, --overflow [term] | Allow titles to overflow columns (no truncation). | searchsploit -o afd windows |
| -p, --path [EDB-ID] | Show full local path and copy to clipboard if possible. | searchsploit -p 39446 |
| -v, --verbose | Show more information (codes, verified flag, file type). | searchsploit -v afd windows |
| -w, --www [term] | Show Exploit‑DB website URLs instead of local paths. | searchsploit -w warftp 1.65 |
| --id | Display EDB‑ID instead of local path. | searchsploit --id afd windows |
| --disable-colour | Disable color highlighting (useful for piping/grep). | searchsploit --disable-colour wordpress mail list | grep 'Mailing List' |

## 4.3 Non‑Searching

|  |  |  |
| --- | --- | --- |
| Option | What it does | Example |
| -m, --mirror [EDB-ID|path] | Copy exploit(s) into the current directory. | searchsploit -m 39446 windows\_x86-64/local/39525.py |
| -x, --examine [EDB-ID|path] | Open exploit with $PAGER for quick reading. | searchsploit -x 39446 |
| -h, --help | Show help screen. | searchsploit -h |
| -u, --update | Update the local Exploit‑DB data (brew, deb, git). | searchsploit -u |

## 4.4 Automation

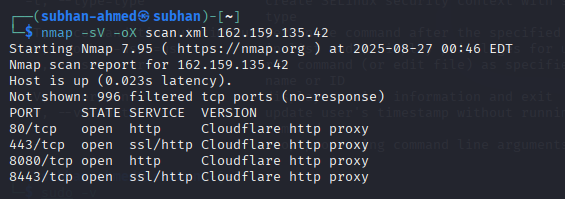
|  |  |  |
| --- | --- | --- |
| Option | What it does | Example |
| --nmap [file.xml] | Parse Nmap -sV XML and cross‑reference for likely exploits. | nmap 10.10.10.10 -sV -oX scan.xml searchsploit --nmap scan.xml -v |

# 5. Practical Workflows (Step‑by‑Step)

## 5.1 From Service Scan to Candidate Exploits (Nmap → SearchSploit)

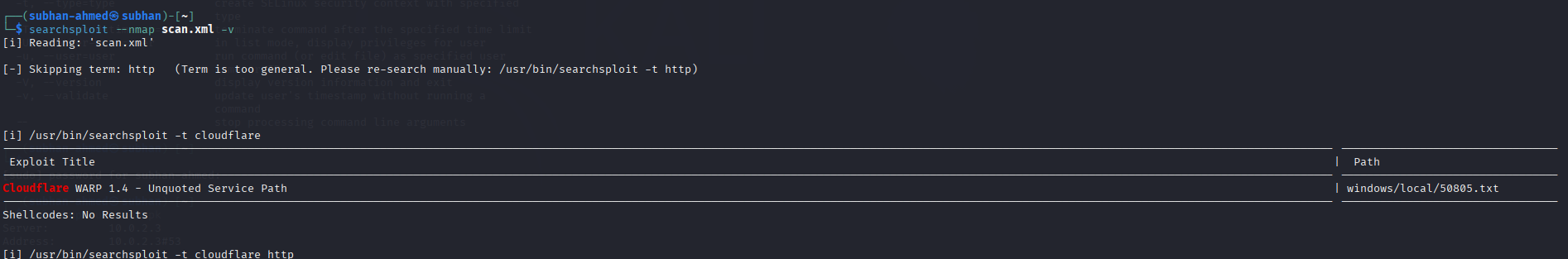
1) Run a version scan and save XML:

nmap -sV -oX scan.xml <target>



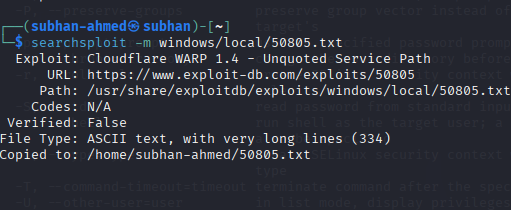
2) Correlate services with Exploit‑DB:

searchsploit --nmap scan.xml –v



3) Review matches, then copy promising PoCs to your workspace:

searchsploit -m <EDB-ID-1> <path-or-EDB-ID-2>



4) Open and read the exploit’s comments/usage:

searchsploit -x <EDB-ID>

## 5.2 Clean Results by Excluding Noise

Remove denial‑of‑service or PoC‑only entries using --exclude:

searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"



## 5.3 Exact, Case‑Sensitive, and Strict Version Searches

Exact title phrase (ordered tokens) and strict versions help reduce false positives:

searchsploit -e "WordPress 5.8.1"

searchsploit -s "Apache Struts 2.0.0"

searchsploit -c Windows afd

## 5.4 Working With IDs, Paths, and URLs

Show the full local path (and copy it to clipboard when possible):

searchsploit -p 39446

Copy the file(s) into the current directory to avoid modifying originals:

searchsploit -m 39446

Open a web view of results when you need screenshots/tags/mirrors:

searchsploit -w "WarFTP 1.65"

## 5.5 Scriptable Output (JSON + jq)

Use JSON when building automation tooling or dashboards:

searchsploit --cve 2021-44228 -j | jq '.[] | {Title, "EDB-ID": .EDBID, Path, URL}'

# 6. Tips & Troubleshooting

• Start broad, then narrow: try product and major version (e.g., “Kernel 5.x”) before full version.

• Avoid abbreviations in queries (use “SQL Injection”, not “SQLi”).

• If output is truncated, use -o (overflow) to see long titles.

• Pipe results through grep/awk for custom filtering—disable colour when grepping.

• Keep Exploit‑DB up to date (searchsploit -u) before every engagement.

• Add Exploit‑DB Papers to include whitepapers in results (install exploitdb‑papers and edit ~/.searchsploit\_rc if needed).

# 7. Appendix: Paths & Files

Key paths on Kali:

/usr/share/exploitdb/ # Repository base  
/usr/share/exploitdb/exploits/ # PoCs by platform and type  
/usr/share/exploitdb/shellcodes/ # Shellcodes  
~/.searchsploit\_rc # User config (paths, sections)

EDB‑IDs (e.g., 39446) uniquely identify an exploit. You can feed either an ID or a repository path to -m/-x.