CMS Vulnerability Scanners for WordPress, Joomla, Drupal, Moodle, Typo3..

infosecmatter.com/cms-vulnerability-scanners-for-wordpress-joomla-drupal-moodle-typo3

November 30, 2020

In this article we will look on 12 free and open-source vulnerability scanners for CMS (Content Management System) such as WordPress, Joomla, Drupal, Moodle, Typo3 and similar publishing platforms.

We will look on Droopescan, CMSmap, CMSeeK, WPXF, WPScan, WPSeku, WPForce, JoomScan, JoomlaVS, JScanner, Drupwn, Typo3Scan vulnerability scanners that were developed specifically to find vulnerabilities in various CMS platforms.

The following table provides list of the scanners discussed in this article and the CMS platforms which they are designed to scan:

Vulnerability scanner	Supported CMS platform
Droopescan	WordPress, Joomla, Drupal, Moodle, SilverStripe
CMSmap	WordPress, Joomla, Drupal, Moodle
CMSeeK	WordPress, Joomla, Drupal and many others
WPXF	WordPress
WPScan	WordPress
WPSeku	WordPress
WPForce	WordPress
JoomScan	Joomla
JoomlaVS	Joomla
JScanner	Joomla
Drupwn	Drupal
Typo3Scan	Typo3

In the rest of the article, we will look in detail on each scanner and go over:

- Features and capabilities
- Command line examples
- Results and sample scan reports

Let’s start!

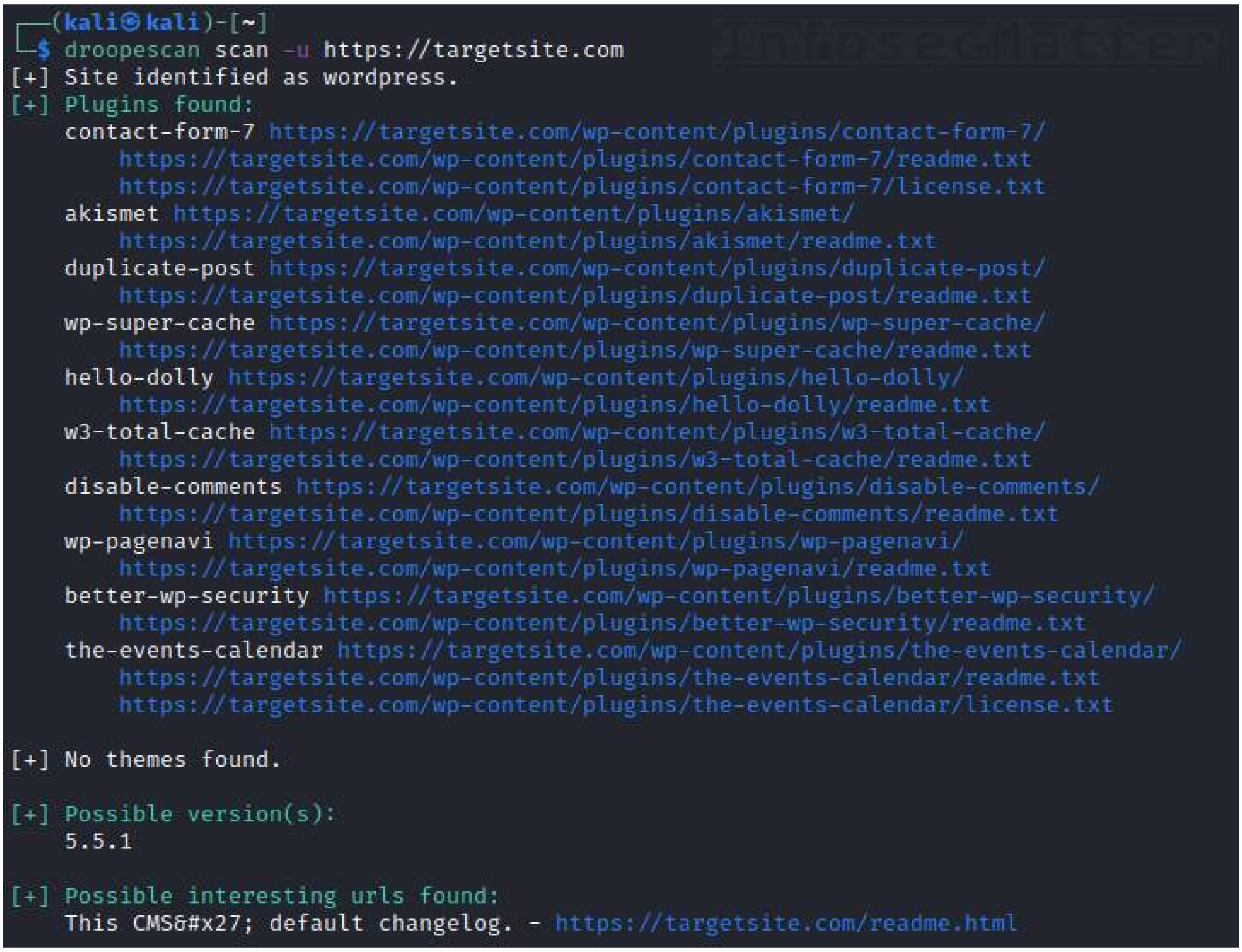
Multi CMS vulnerability scanners

This section contains vulnerability scanners with support of multiple different CMSs such as WordPress, Joomla, Drupal, Moodle and others.

These scanners can be used in general to scan any CMS platform.

Droopescan

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~]
└─$ droopescan scan -u https://targetsite.com
[+] Site identified as wordpress.
[+] Plugins found:
    contact-form-7 https://targetsite.com/wp-content/plugins/contact-form-7/
        https://targetsite.com/wp-content/plugins/contact-form-7/readme.txt
        https://targetsite.com/wp-content/plugins/contact-form-7/license.txt
    akismet https://targetsite.com/wp-content/plugins/akismet/
        https://targetsite.com/wp-content/plugins/akismet/readme.txt
    duplicate-post https://targetsite.com/wp-content/plugins/duplicate-post/
        https://targetsite.com/wp-content/plugins/duplicate-post/readme.txt
    wp-super-cache https://targetsite.com/wp-content/plugins/wp-super-cache/
        https://targetsite.com/wp-content/plugins/wp-super-cache/readme.txt
    hello-dolly https://targetsite.com/wp-content/plugins/hello-dolly/
        https://targetsite.com/wp-content/plugins/hello-dolly/readme.txt
    w3-total-cache https://targetsite.com/wp-content/plugins/w3-total-cache/
        https://targetsite.com/wp-content/plugins/w3-total-cache/readme.txt
    disable-comments https://targetsite.com/wp-content/plugins/disable-comments/
        https://targetsite.com/wp-content/plugins/disable-comments/readme.txt
    wp-pagenavi https://targetsite.com/wp-content/plugins/wp-pagenavi/
        https://targetsite.com/wp-content/plugins/wp-pagenavi/readme.txt
    better-wp-security https://targetsite.com/wp-content/plugins/better-wp-security/
        https://targetsite.com/wp-content/plugins/better-wp-security/readme.txt
    the-events-calendar https://targetsite.com/wp-content/plugins/the-events-calendar/
        https://targetsite.com/wp-content/plugins/the-events-calendar/readme.txt
        https://targetsite.com/wp-content/plugins/the-events-calendar/license.txt

[+] No themes found.

[+] Possible version(s):
    5.5.1

[+] Possible interesting urls found:
    This CMS&#x27; default changelog. - https://targetsite.com/readme.html
```

Droopescan is a plugin-based vulnerability scanner written in python capable of scanning several popular CMS. Currently it supports the following CMS:

- Drupal
- WordPress
- SilverStripe
- Joomla (partial support)
- Moodle (partial support)

When it comes to features, this is what Droopescan can do:

- Autodetect remote CMS
- Enumerate installed themes and plugins

- Partially fingerprint component versions
- Find interesting URLs (admin panels, readme files etc.)
- Simultaneous scanning of multiple sites

Here’s how to run Droopescan to scan remote CMS for vulnerabilities:

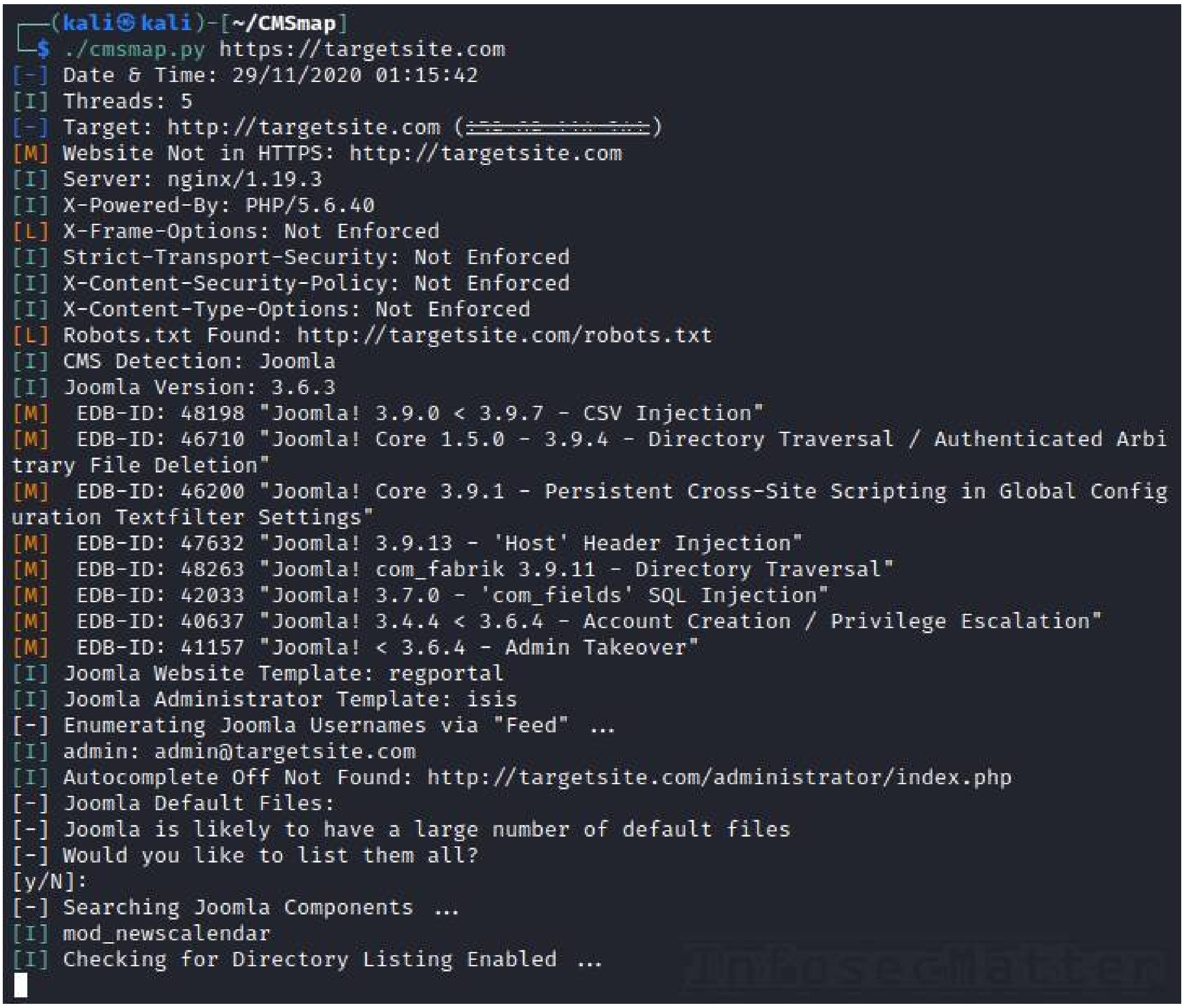
```
droopescan scan -u https://targetsite.com
```

Without specifying the CMS platform, Droopescan will autodetect it and then fingerprint the target site accordingly.

Although the version detection capabilities are not so detailed when compared to other CMS scanners, Droopescan can definitely provide very good and reliable information about what is running on the remote website(s).

CMSmap

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/CMSmap]
└─$ ./cmsmap.py https://targetsite.com
[-] Date & Time: 29/11/2020 01:15:42
[I] Threads: 5
[-] Target: http://targetsite.com ([redacted])
[M] Website Not in HTTPS: http://targetsite.com
[I] Server: nginx/1.19.3
[I] X-Powered-By: PHP/5.6.40
[L] X-Frame-Options: Not Enforced
[I] Strict-Transport-Security: Not Enforced
[I] X-Content-Security-Policy: Not Enforced
[I] X-Content-Type-Options: Not Enforced
[L] Robots.txt Found: http://targetsite.com/robots.txt
[I] CMS Detection: Joomla
[I] Joomla Version: 3.6.3
[M]  EDB-ID: 48198 "Joomla! 3.9.0 < 3.9.7 - CSV Injection"
[M]  EDB-ID: 46710 "Joomla! Core 1.5.0 - 3.9.4 - Directory Traversal / Authenticated Arbi
trary File Deletion"
[M]  EDB-ID: 46200 "Joomla! Core 3.9.1 - Persistent Cross-Site Scripting in Global Config
uration Textfilter Settings"
[M]  EDB-ID: 47632 "Joomla! 3.9.13 - 'Host' Header Injection"
[M]  EDB-ID: 48263 "Joomla! com_fabrik 3.9.11 - Directory Traversal"
[M]  EDB-ID: 42033 "Joomla! 3.7.0 - 'com_fields' SQL Injection"
[M]  EDB-ID: 40637 "Joomla! 3.4.4 < 3.6.4 - Account Creation / Privilege Escalation"
[M]  EDB-ID: 41157 "Joomla! < 3.6.4 - Admin Takeover"
[I] Joomla Website Template: regportal
[I] Joomla Administrator Template: isis
[-] Enumerating Joomla Usernames via "Feed" ...
[I] admin: admin@targetsite.com
[I] Autocomplete Off Not Found: http://targetsite.com/administrator/index.php
[-] Joomla Default Files:
[-] Joomla is likely to have a large number of default files
[-] Would you like to list them all?
[y/N]:
[-] Searching Joomla Components ...
[I] mod_newscalendar
[I] Checking for Directory Listing Enabled ...
█
```

CMSmap is another popular and capable vulnerability scanner for CMS. It is written in python and it currently supports all these CMS:

- WordPress
- Joomla
- Drupal
- Moodle

It can enumerate installed components of any of the supported CMS. Here are the main CMSmap features:

- Detection of outdated versions with referenced exploits (EDB-ID)
- Detection of misconfigurations, default files, interesting URLs etc.
- Automation ready (accepts a list of targets)
- Username enumeration
- Login brute force attacks
- Password hash cracking

CMSmap can also autodetect the installed remote CMS, which makes the usage very simple as well. To scan a remote site with CMSmap, simply run:

```
cmsmap.py https://targetsite.com
```

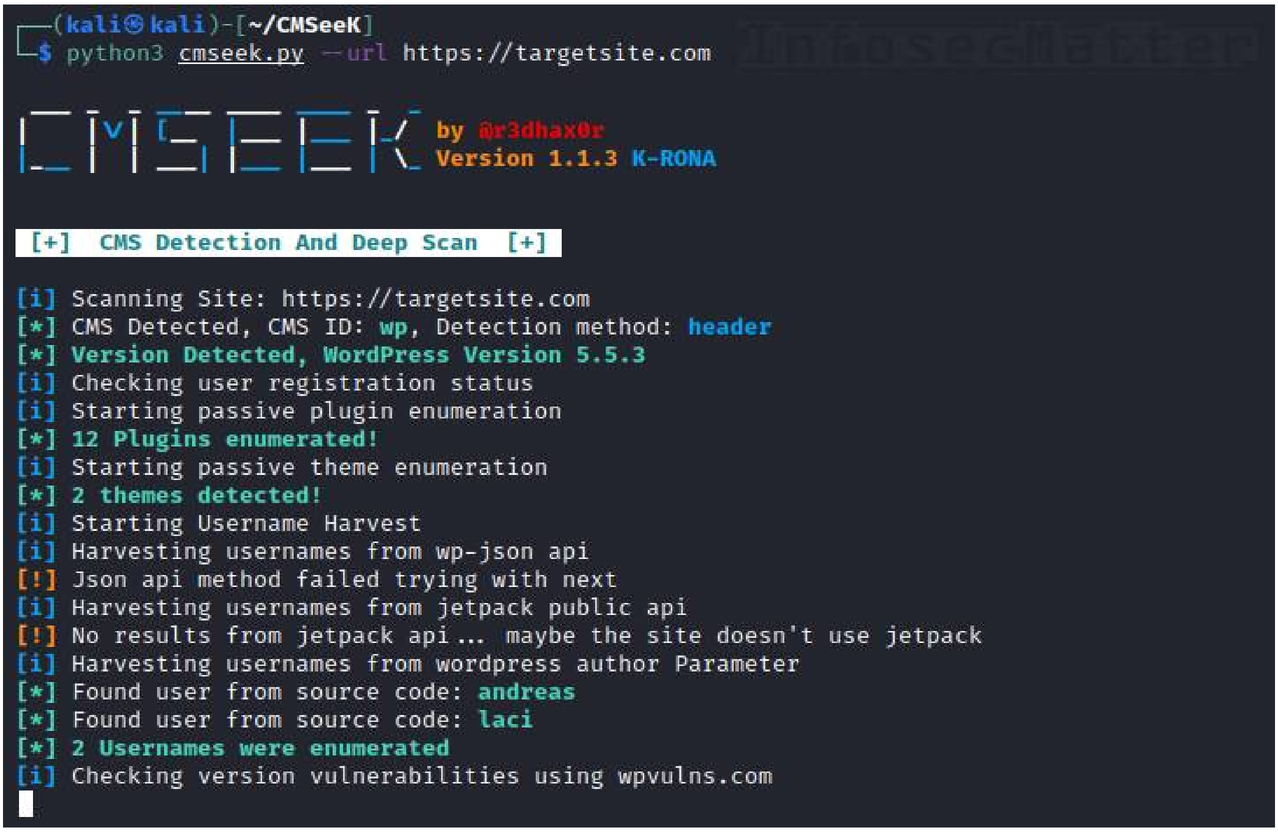
From the sample report you can see that CMSmap provides very useful and actionable information about the target CMS and can even detect various misconfigurations such as missing HTTP security headers, enabled “Autocomplete” browser feature for a password field and other things.

Make sure to update CMSmap before running it on your target:

```
cmsmap.py --update PC
```

CMSeeK

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/CMSeeK]
└─$ python3 cmseek.py --url https://targetsite.com

                                   by @r3dhax0r
                                   Version 1.1.3 K-RONA

 [+]  CMS Detection And Deep Scan  [+]

[i] Scanning Site: https://targetsite.com
[*] CMS Detected, CMS ID: wp, Detection method: header
[*] Version Detected, WordPress Version 5.5.3
[i] Checking user registration status
[i] Starting passive plugin enumeration
[*] 12 Plugins enumerated!
[i] Starting passive theme enumeration
[*] 2 themes detected!
[i] Starting Username Harvest
[i] Harvesting usernames from wp-json api
[!] Json api method failed trying with next
[i] Harvesting usernames from jetpack public api
[!] No results from jetpack api... maybe the site doesn't use jetpack
[i] Harvesting usernames from wordpress author Parameter
[*] Found user from source code: andreas
[*] Found user from source code: laci
[*] 2 Usernames were enumerated
[i] Checking version vulnerabilities using wpvulns.com
```

CMSeeK is a CMS detection and exploitation suite. It is quite a remarkable vulnerability scanner since it supports more than 180 of CMS platforms. It supports:

- Joomla (advanced features)
- WordPress (advanced features)
- 180+ other publishing, shopping and CMS platforms (basic detection only)
  [Click here for the complete list]

When it comes to capabilities, this is what CMSeeK can do:

- Enumeration of installed CMS components and their versions
- Detection of misconfigurations, exposed admin consoles, backup files etc.
- Detection of outdated versions with direct links to exploits
- Automation ready (accepts a list of targets)
- Username enumeration
- JSON reporting

The usage is very straightforward, simply provide the target URL to CMSeeK like this:

```
python3 cmseek.py --url https://targetsite.com
```

CMSeeK will autodetect the installed CMS and enumerate it.

In the end it will print out detailed information with all the findings and also produce a JSON report for further machine processing.

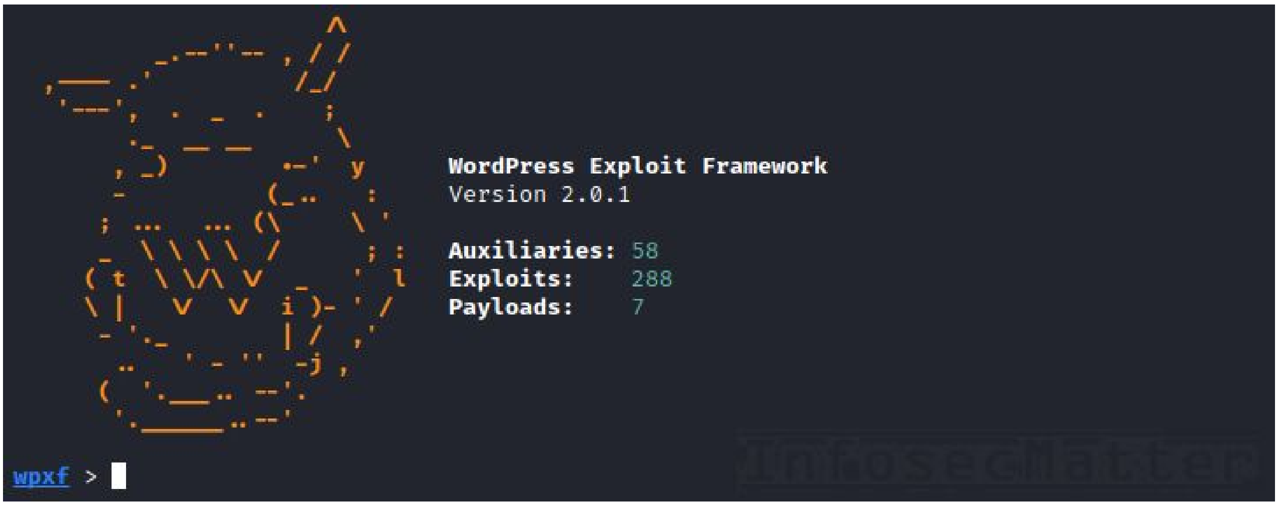
Go back to top.

WordPress vulnerability scanners

This section contains vulnerability scanners and tools designed specifically for identifying and exploiting vulnerabilities in WordPress CMS.

WPXF

Github repository | Homepage



```
WordPress Exploit Framework
Version 2.0.1

Auxiliaries: 58
Exploits:    288
Payloads:    7

wpxf >
```

WordPress Exploit Framework (WPXF) is a framework written in Ruby for penetration testing of WordPress powered websites. The user interface is very similar to the Metasploit framework, so it is intuitive for anyone familiar with Metasploit.

The framework currently contains more than 288 exploits, 58 auxiliary modules and 7 payloads for exploiting of WordPress instances. It allows to exploit known WordPress vulnerabilities and chain them together with any of the following payloads:

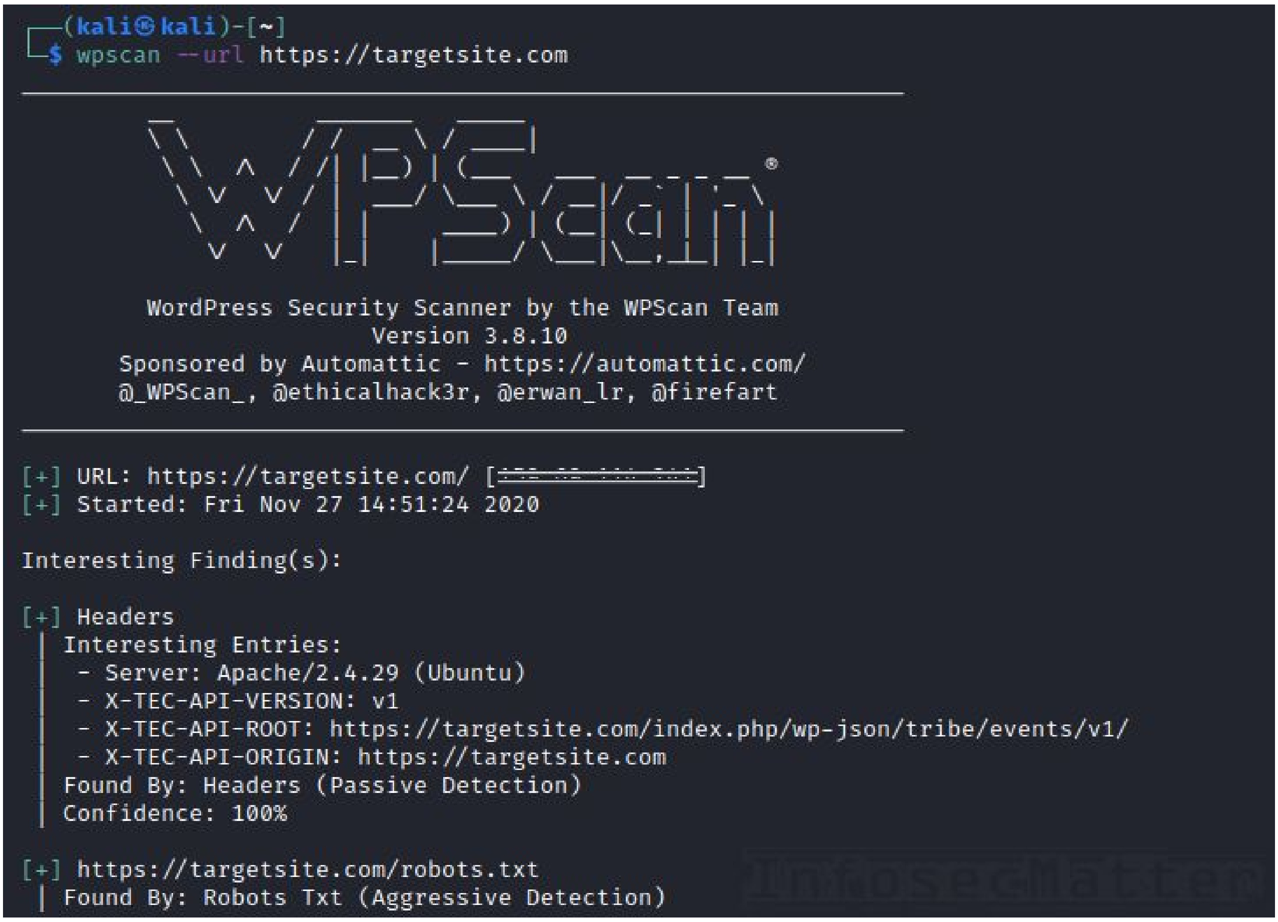
- **bind_php** – bind PHP shell
- **download_exec** – download file from a URL and execute it
- **meterpreter_bind_tcp** – bind meterpreter session
- **meterpreter_reverse_tcp** – reverse meterpreter session
- **reverse_tcp** – reverse shell via TCP
- **custom** – your own payload

Note that there are no modules for enumeration, fingerprinting nor scanning for vulnerabilities. This means that you have to do the reconnaissance before you use WPXF for the actual exploitation.

On this page you can find the list of supported commands and details on how to use this framework.

WPScan

GitHub repository | Homepage | Sample report



```
┌──(kali㉿kali)-[~]
└─$ wpscan --url https://targetsite.com
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.10
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: https://targetsite.com/ [[redacted]]
[+] Started: Fri Nov 27 14:51:24 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.29 (Ubuntu)
 |  - X-TEC-API-VERSION: v1
 |  - X-TEC-API-ROOT: https://targetsite.com/index.php/wp-json/tribe/events/v1/
 |  - X-TEC-API-ORIGIN: https://targetsite.com
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] https://targetsite.com/robots.txt
 | Found By: Robots Txt (Aggressive Detection)
```

WordPress Security Scanner (WPScan) is currently the most advanced vulnerability scanner for WordPress powered sites. It has many useful features such as:

- Enumeration of installed components (plugins, themes) and their version
- Detection of config backups, db exports and other misconfigurations
- Automatic reporting of outdated versions with links to exploits
- Enumeration of remote users
- Login brute force attacks

Usage is very straightforward – to scan a remote WordPress site for vulnerabilities, simply run:

```
wpscan --url https://targetsite.com
```

Note that WPScan can also provide a list of exploits associated with the detected vulnerable versions. This works by using the WordPress Vulnerability Database which currently contains more than 21,850 vulnerabilities.

In order to use it, you have to get an API token by registering on the wpscan.com website (for free). Once you have the token, you can run the scan like this:

```
wpscan --url https://targetsite.com --api-token <YOUR-TOKEN>
```

WPScan will then provide direct links to vulnerabilities and exploits.

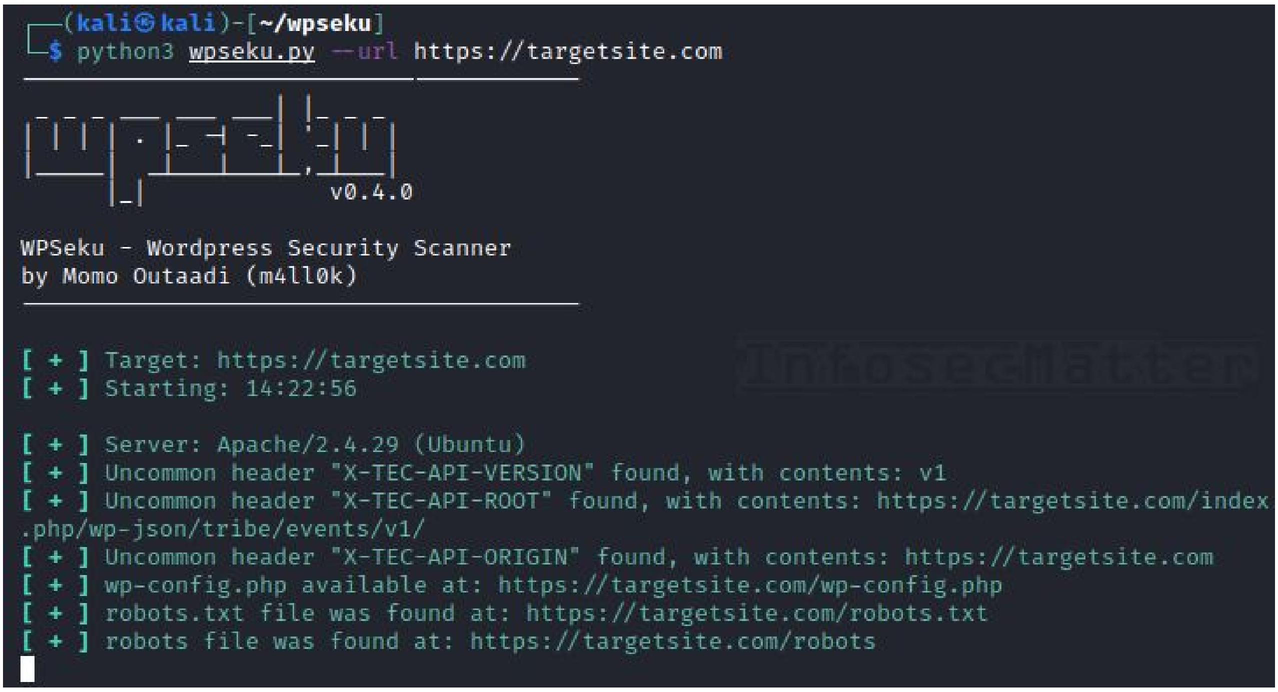
WPScan also uses a local database with various useful WordPress metadata, latest version strings and so on. Make sure to update the database every now and then by running:

```
wpscan --update
```

More details about the WPScan usage can be found here.

WPSeku

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/wpseku]
└─$ python3 wpseku.py --url https://targetsite.com

                          v0.4.0

WPSeku - Wordpress Security Scanner
by Momo Outaadi (m4ll0k)

[ + ] Target: https://targetsite.com
[ + ] Starting: 14:22:56

[ + ] Server: Apache/2.4.29 (Ubuntu)
[ + ] Uncommon header "X-TEC-API-VERSION" found, with contents: v1
[ + ] Uncommon header "X-TEC-API-ROOT" found, with contents: https://targetsite.com/index
.php/wp-json/tribe/events/v1/
[ + ] Uncommon header "X-TEC-API-ORIGIN" found, with contents: https://targetsite.com
[ + ] wp-config.php available at: https://targetsite.com/wp-config.php
[ + ] robots.txt file was found at: https://targetsite.com/robots.txt
[ + ] robots file was found at: https://targetsite.com/robots
```

WPSeku is another popular and capable WordPress security scanner with one very unique feature. Here’s what you can do with WPSeku:

- Enumeration of installed components (plugins, themes)
- Detection of various misconfigurations and exposed files
- Static code analysis of any WordPress plugin code
- Login brute force attacks

Here’s how you can scan a remote WordPress site with WPSeku:

```
python3 wpseku.py --url https://targetsite.com
```

As you can see from the sample report, WPSeku can find out various useful information about the target, but that’s not all what it can do..

Scan WordPress plugin code for vulnerabilities

As mentioned above, WPSeku can also find security vulnerabilities in WordPress plugins by statically analyzing their code. It goes through the entire plugin directory you specify and it finds potentially insecure code constructs and functions in the identified PHP code.

All you have to do is to obtain copy of the plugin you want to scan and get it locally on your disk.

One way to do it is e.g. from the official https://plugins.svn.wordpress.org/ public repository. Here’s the complete step-by-step process:

(1) Get the plugin code (in this example we are obtaining the ‘wp-photo-gallery’ plugin):

```
wget --no-parent --mirror https://plugins.svn.wordpress.org/wp-photo-gallery/
```

(2) Now do the static analysis on the plugin code by running:

```
python3 wpseku.py --scan plugins.svn.wordpress.org/wp-photo-gallery/ --verbose
```

WPSeku will print out any potential security vulnerabilities.

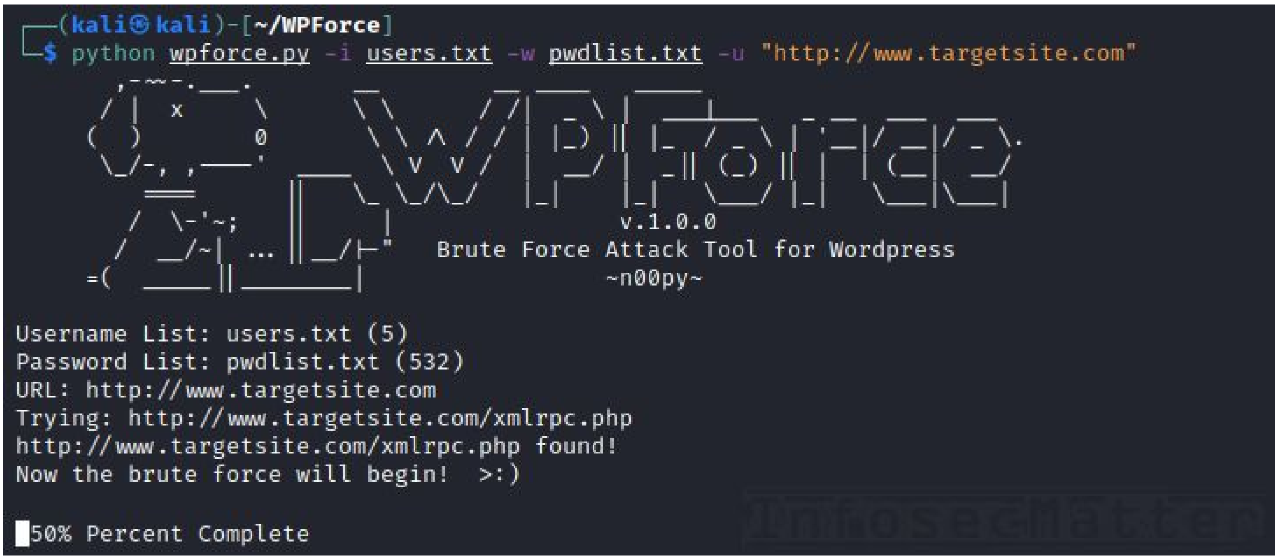
Note that not all plugins are available in the official plugin repository linked above, so you may have to obtain the plugin code in some other way – e.g. download from the plugin vendor / developer.

You could also replicate the target environment by installing the same plugins as the target site and then you can get the plugin code directly from the ‘<wordpress-dir>/wp-content/plugins/’ directory.

Zip it, download it and scan it with WPSeku.

WPForce

GitHub repository | Homepage



```
┌──(kali㉿kali)-[~/WPForce]
└─$ python wpforce.py -i users.txt -w pwdlist.txt -u "http://www.targetsite.com"
                                                  v.1.0.0
                                   Brute Force Attack Tool for Wordpress
                                                 ~n00py~

Username List: users.txt (5)
Password List: pwdlist.txt (532)
URL: http://www.targetsite.com
Trying: http://www.targetsite.com/xmlrpc.php
http://www.targetsite.com/xmlrpc.php found!
Now the brute force will begin!  >:)

█50% Percent Complete
```

WPForce is a collection of WordPress attack tools with another unique set of capabilities. Currently it contains these 2 python scripts:

- **wpforce.py** – admin login brute force tool (stealth via WordPress API)
- **yertle.py** – backdoor shell upload with a number of post exploitation modules

Here’s how to use it:

First we have to find valid login credentials to the target WordPress site by performing brute force login attack with **wpforce.py**:

```
python wpforce.py -i users.txt -w pwdlist.txt -u "http://www.targetsite.com"
```

Once we find valid credentials with **wpforce.py**, we can then use the **yertle.py** script for the post exploitation tasks.

This is what we can do with the **yertle.py** script:

- Upload an interactive system shell
- Spawn a full featured reverse shell
- Dump WordPress password hashes
- Backdoor authentication functions to capture plain text passwords
- Inject BeEF hook into all pages and attack the website’s visitors
- Pivot to meterpreter if needed

Here’s a typical usage:

```
python yertle.py -u admin -p password -t https://targetsite.com/ --interactive
```

This will upload a backdoor on the target site and spawn an interactive console (os-shell) which will give us control over the target.

See this page for details on how to use the os-shell console.

Online WordPress security scanners

There are also 3rd party online services which offer vulnerability scanning of WordPress CMS:

- https://www.malcare.com/wordpress-malware-scan/
- https://gf.dev/wordpress-security-scanner
- https://wpsec.com/

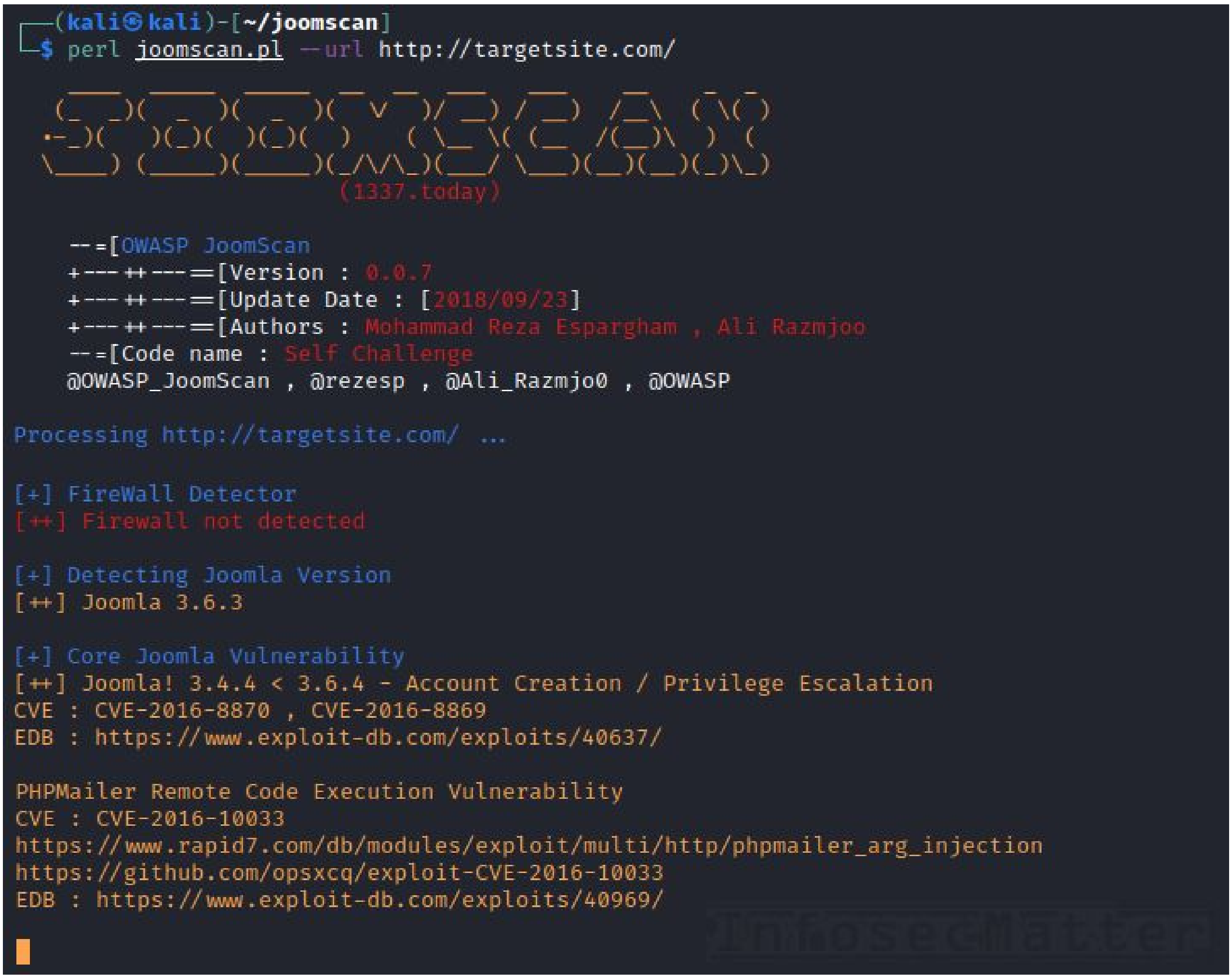
Go back to top.

Joomla vulnerability scanners

This section contains list vulnerability scanners designed specifically for identifying vulnerabilities in Joomla CMS.

JoomScan

GitHub repository | Homepage | Sample report



```
┌──(kali㉿kali)-[~/joomscan]
└─$ perl joomscan.pl --url http://targetsite.com/

    ____  _____  _____  __  __  ___   ___    __    _  _
   (_  _)(  _  )(  _  )(  \/  )/ __) / __)  /__\  ( \( )
  .-_)(   )(_)(  )(_)(  )    ( \__ \( (__  /(__)\  )  (
  \____) (_____)(_____)(_/\/\_)(___/ \___)(__)(__)(_)\_)
                        (1337.today)

    --=[OWASP JoomScan
    +---++---==[Version : 0.0.7
    +---++---==[Update Date : [2018/09/23]
    +---++---==[Authors : Mohammad Reza Espargham , Ali Razmjoo
    --=[Code name : Self Challenge
    @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

Processing http://targetsite.com/ ...

[+] FireWall Detector
[++] Firewall not detected

[+] Detecting Joomla Version
[++] Joomla 3.6.3

[+] Core Joomla Vulnerability
[++] Joomla! 3.4.4 < 3.6.4 - Account Creation / Privilege Escalation
CVE : CVE-2016-8870 , CVE-2016-8869
EDB : https://www.exploit-db.com/exploits/40637/

PHPMailer Remote Code Execution Vulnerability
CVE : CVE-2016-10033
https://www.rapid7.com/db/modules/exploit/multi/http/phpmailer_arg_injection
https://github.com/opsxcq/exploit-CVE-2016-10033
EDB : https://www.exploit-db.com/exploits/40969/
```

OWASP Joomla! Vulnerability Scanner (JoomScan) is currently the most popular vulnerability scanner for Joomla powered sites. It has many useful features such as:

- Enumeration of installed Joomla components and their version
- Detection of backup files, log files and other misconfigurations
- Reporting of outdated versions with direct links to exploits
- Detection of Firewall, WAF and CDN
- HTML / text reporting

Usage is very simple. To scan a remote website with JoomScan, simply run:

```
perl joomscan.pl --url http://targetsite.com/
```

We can see the results nicely formatted on the console and there will also be an HTML and text report automatically generated in the current folder.

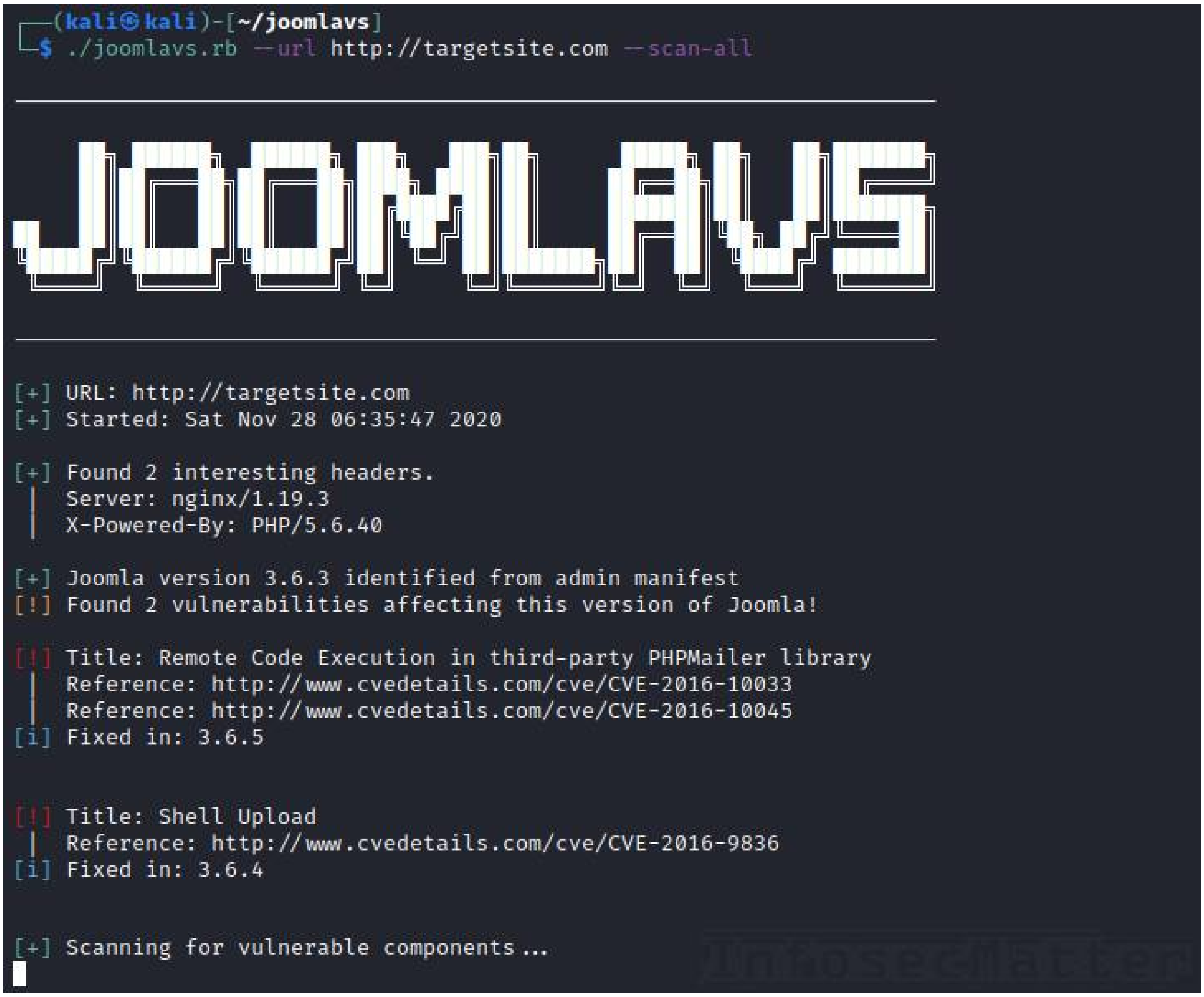
Note that JoomScan can also perform detailed component enumeration by probing various endpoints (URLs) using a built-in wordlist. Here's how to do it:

```
perl joomscan.pl --url http://targetsite.com/ --enumerate-components
```

This will take more time, but it can potentially identify additional components, some of which could contain security vulnerabilities.

JoomlaVS

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/joomlavs]
└─$ ./joomlavs.rb --url http://targetsite.com --scan-all

[+] URL: http://targetsite.com
[+] Started: Sat Nov 28 06:35:47 2020

[+] Found 2 interesting headers.
 |  Server: nginx/1.19.3
 |  X-Powered-By: PHP/5.6.40

[+] Joomla version 3.6.3 identified from admin manifest
[!] Found 2 vulnerabilities affecting this version of Joomla!

[!] Title: Remote Code Execution in third-party PHPMailer library
 |  Reference: http://www.cvedetails.com/cve/CVE-2016-10033
 |  Reference: http://www.cvedetails.com/cve/CVE-2016-10045
[i] Fixed in: 3.6.5


[!] Title: Shell Upload
 |  Reference: http://www.cvedetails.com/cve/CVE-2016-9836
[i] Fixed in: 3.6.4


[+] Scanning for vulnerable components...
```

JoomlaVS is another capable vulnerability scanner for Joomla powered websites. It is written in Ruby and it has the following features:

- Enumeration of installed Joomla components and their version
- Reporting of outdated versions with direct links to exploits
- Basic detection of misconfigurations and insecurities
- Output on the console

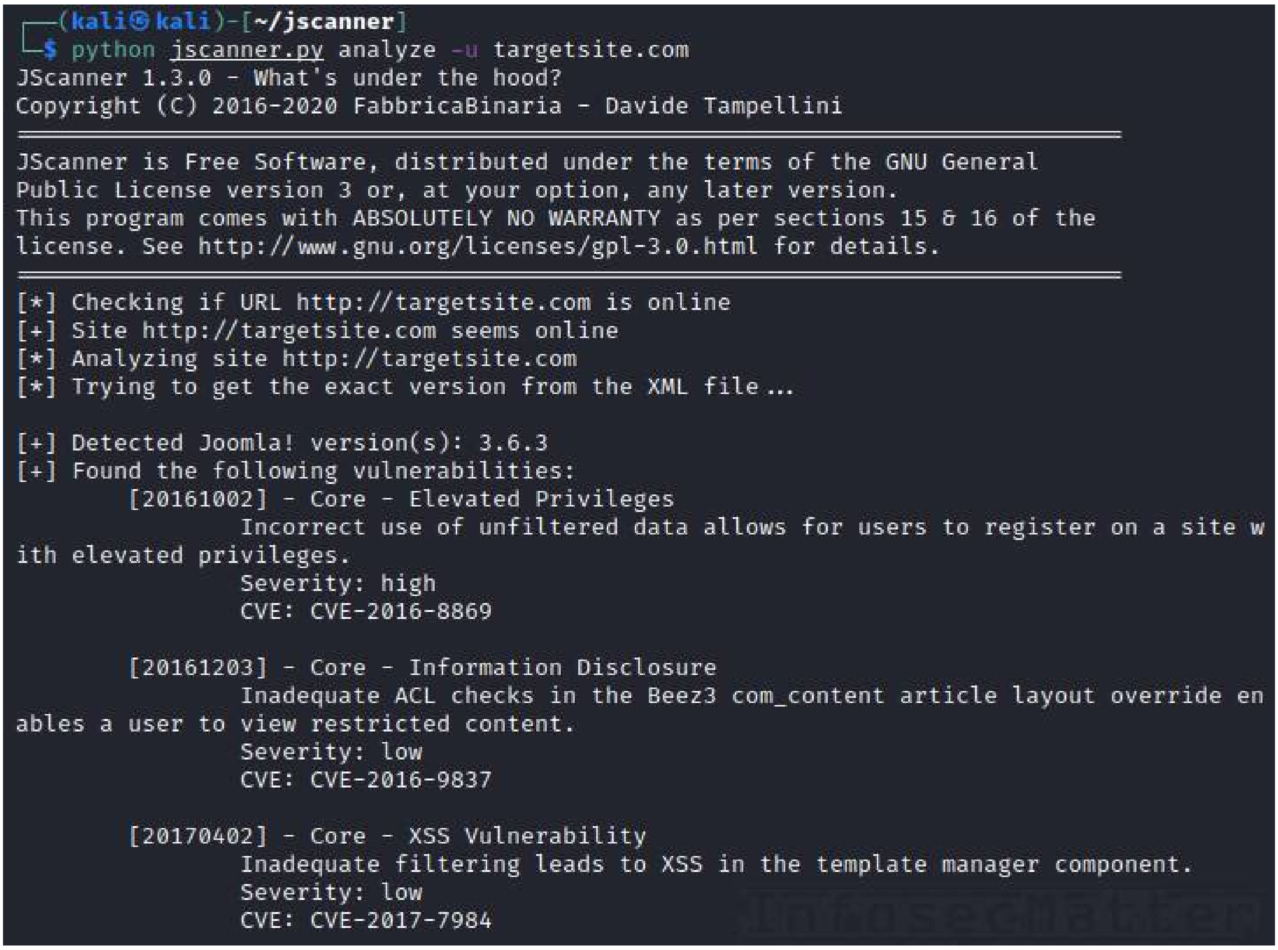
Here’s how to scan a remote Joomla installation with JoomlaVS:

```
joomlavs.rb --url https://targetsite.com --scan-all
```

As you can see from the sample report, if there is a known exploit for any of the identified vulnerabilities, JoomlaVS will print it out and also provide link to the exploit.

JScanner

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/jscanner]
└─$ python jscanner.py analyze -u targetsite.com
JScanner 1.3.0 - What's under the hood?
Copyright (C) 2016-2020 FabbricaBinaria - Davide Tampellini
═══════════════════════════════════════════════════════════════════════════════
JScanner is Free Software, distributed under the terms of the GNU General
Public License version 3 or, at your option, any later version.
This program comes with ABSOLUTELY NO WARRANTY as per sections 15 & 16 of the
license. See http://www.gnu.org/licenses/gpl-3.0.html for details.
═══════════════════════════════════════════════════════════════════════════════
[*] Checking if URL http://targetsite.com is online
[+] Site http://targetsite.com seems online
[*] Analyzing site http://targetsite.com
[*] Trying to get the exact version from the XML file...

[+] Detected Joomla! version(s): 3.6.3
[+] Found the following vulnerabilities:
        [20161002] - Core - Elevated Privileges
                Incorrect use of unfiltered data allows for users to register on a site w
ith elevated privileges.
                Severity: high
                CVE: CVE-2016-8869

        [20161203] - Core - Information Disclosure
                Inadequate ACL checks in the Beez3 com_content article layout override en
ables a user to view restricted content.
                Severity: low
                CVE: CVE-2016-9837

        [20170402] - Core - XSS Vulnerability
                Inadequate filtering leads to XSS in the template manager component.
                Severity: low
                CVE: CVE-2017-7984
```

JScanner is a vulnerability scanner that can analyze remote Joomla CMS installations using several different techniques. It is written in python and apart from identifying vulnerabilities, it can also perform user enumeration.

Here’s how to use JScanner to scan a remote Joomla powered site:

```
python jscanner.py analyze -u targetsite.com
```

We will then see the results printed on the console (sample report).

Apart from vulnerability scanning, JScanner can also perform user enumeration. This works, however, only if the user registration function is enabled on the target Joomla site.

Here’s how to do the user enumeration with JScanner:

```
python jscanner.py enumerate -u http://targetsite.com -U users.txt
```

JScanner will then list detected valid usernames on the console. Consequently, we can use this list to perform login attacks on the identified users.

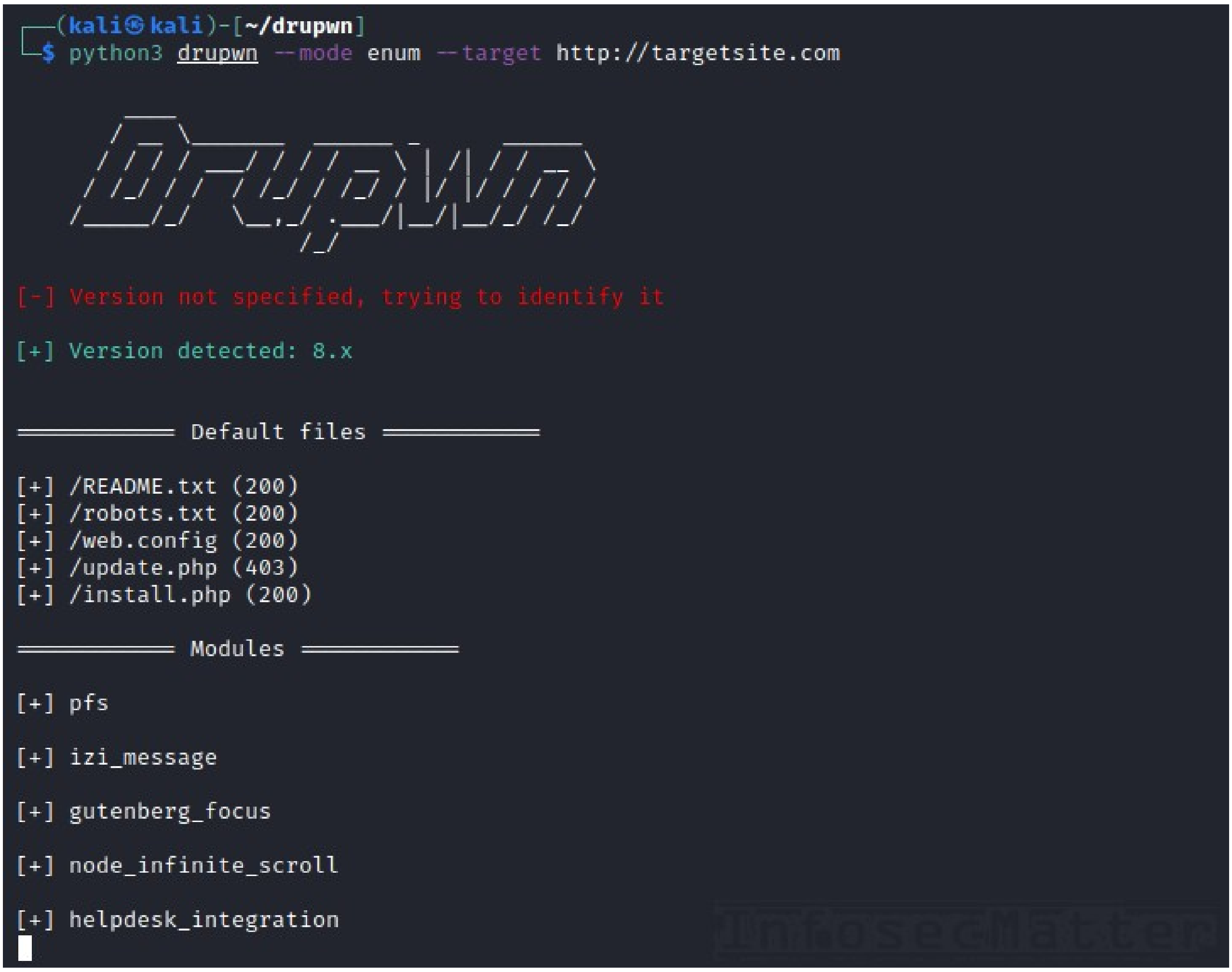
Go back to top.

Drupal vulnerability scanners

This section contains vulnerability scanners designed specifically for identifying vulnerabilities in Drupal CMS.

Drupwn

GitHub repository | Homepage | Sample report



```
┌──(kali㉿kali)-[~/drupwn]
└─$ python3 drupwn --mode enum --target http://targetsite.com

        ____
       / __ \_______  ______ _      ______
      / / / / ___/ / / / __ \ | /| / / __ \
     / /_/ / /  / /_/ / /_/ / |/ |/ / / / /
    /_____/_/   \__,_/ .___/|__/|__/_/ /_/
                    /_/

[-] Version not specified, trying to identify it

[+] Version detected: 8.x


============ Default files ============

[+] /README.txt (200)
[+] /robots.txt (200)
[+] /web.config (200)
[+] /update.php (403)
[+] /install.php (200)

============ Modules ============

[+] pfs

[+] izi_message

[+] gutenberg_focus

[+] node_infinite_scroll

[+] helpdesk_integration
```

Drupwn is a powerful Drupal enumeration and exploitation tool written in python. It runs in two different modes – **enum** and **exploit**.

In the **enum** mode, Drupwn will enumerate various Drupal components, namely:

- Users
- Nodes
- Default files
- Modules
- Themes

Here’s how to enumerate a remote Drupal site with Drupwn:

```
python3 drupwn --mode enum --target http://targetsite.com
```

In the **exploit** mode, Drupwn can check and exploit several recent Drupal RCE (Remote Command Execution) vulnerabilities.

To start Drupwn in the **exploit** mode, run it like this:

```
python3 drupwn --mode exploit --target http://targetsite.com
```

And then follow the options in the menu.

Online Drupal scanners

There are also 3rd party online services which offer a vulnerability scan of Drupal CMS:

- https://pentest-tools.com/cms-vulnerability-scanning/drupal-scanner
- https://hackertarget.com/drupal-security-scan/

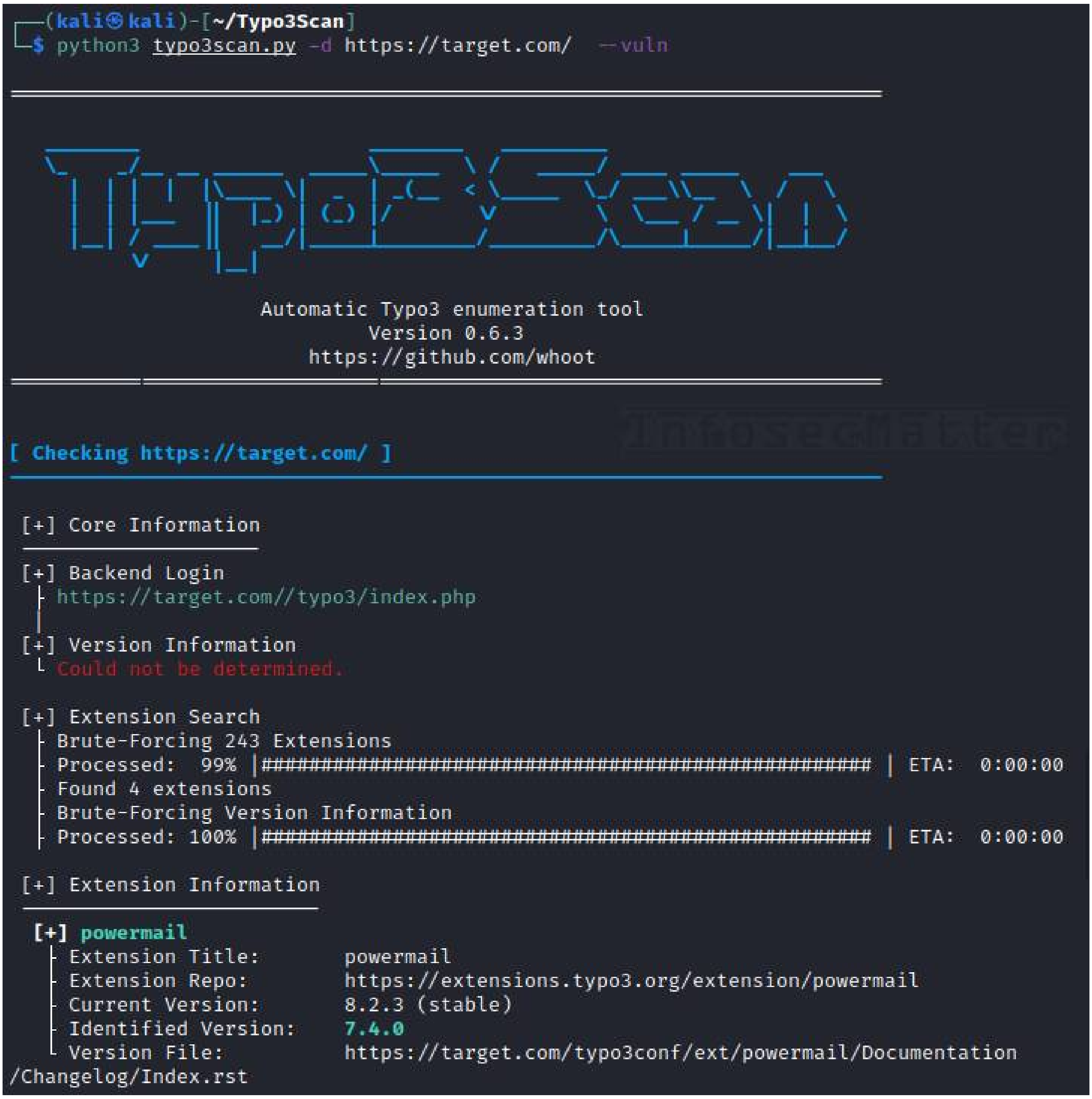
Go back to top.

Typo3 vulnerability scanners

This section contains vulnerability scanners designed specifically for identifying vulnerabilities in Typo3 CMS.

Typo3Scan

GitHub repository | Sample report



```
┌──(kali㉿kali)-[~/Typo3Scan]
└─$ python3 typo3scan.py -d https://target.com/  --vuln

═══════════════════════════════════════════════════════════════════════

                    Automatic Typo3 enumeration tool
                           Version 0.6.3
                       https://github.com/whoot
═══════════════════════════════════════════════════════════════════════

[ Checking https://target.com/ ]
───────────────────────────────────────────────────────────────────────

 [+] Core Information
 ────────────────────
 [+] Backend Login
  ├ https://target.com//typo3/index.php
  │
 [+] Version Information
  └ Could not be determined.

 [+] Extension Search
  ├ Brute-Forcing 243 Extensions
  ├ Processed:  99% |################################################## | ETA:  0:00:00
  ├ Found 4 extensions
  ├ Brute-Forcing Version Information
  ├ Processed: 100% |################################################## | ETA:  0:00:00

 [+] Extension Information
 ─────────────────────────
  [+] powermail
   ├ Extension Title:       powermail
   ├ Extension Repo:        https://extensions.typo3.org/extension/powermail
   ├ Current Version:       8.2.3 (stable)
   ├ Identified Version:    7.4.0
   └ Version File:          https://target.com/typo3conf/ext/powermail/Documentation
/Changelog/Index.rst
```

Typo3Scan is a penetration testing tool for enumerating of Typo3 powered CMS sites and installed extensions. It also has a database with known vulnerabilities for the Typo3 core and the extensions.

Before running it, make sure to update the database by running:

```
python typo3scan.py -u
```

To scan a remote Typo3 CMS site for vulnerabilities, run:

```
python typo3scan.py -d http://targetsite.com --vuln
```

The scanner will report all identified extensions on the console and highlight any outdated version (see sample report).

Note that the tool can also produce JSON output, useful for further machine processing.

Typo3Scan can also perform a detailed fingerprinting by trying to enumerate all known Typo3 extensions (currently more than 8,100 extensions) by running it like this:

```
python typo3scan.py -d http://targetsite.com
```

Such detailed enumeration will take at least 30 minutes to finish, but it will find many more deployed extensions, not just the ones that are known to be vulnerable.

Go back to top.

Conclusion

It’s incredible to see how many free and open-source vulnerability scanners exist out there. The infosec community just seems to never stop giving.

Hope you will find this collection useful sometimes during your penetration tests. Make sure to only scan systems with explicit consent of the owner to stay within the boundaries of ethical hacking.

Have I missed any other CMS scanning tool that is worth mentioning here? Please share in the comment section.

If you liked this collection and you would like more content like this, please subscribe to my mailing list and follow InfosecMatter on Twitter and Facebook to not miss any new additions!