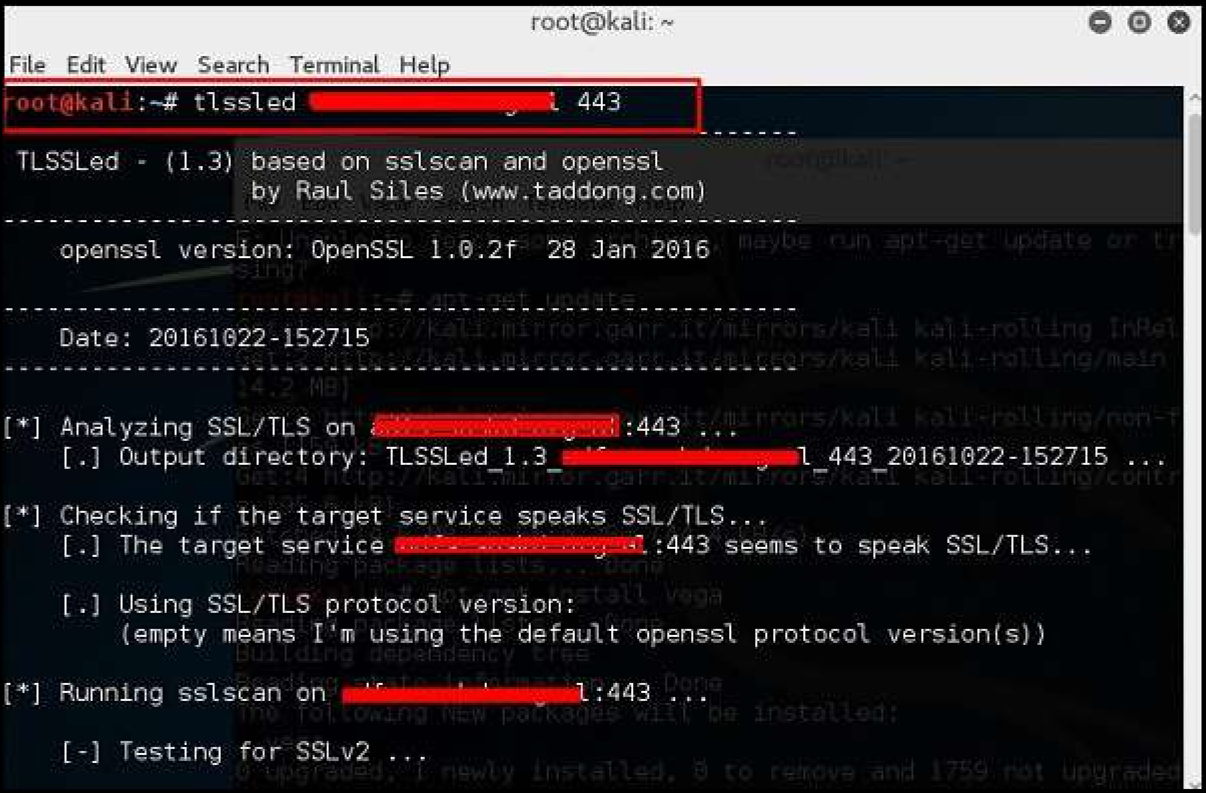
Website Penetration Testing

SSL Scanning Tools

TLSSLed is a Linux shell script used to evaluate the security of a target SSL/TLS (HTTPS) web server implementation. It is based on sslscan, a thorough SSL/TLS scanner that is based on the openssl library, and on the "**openssl s_client**" command line tool.

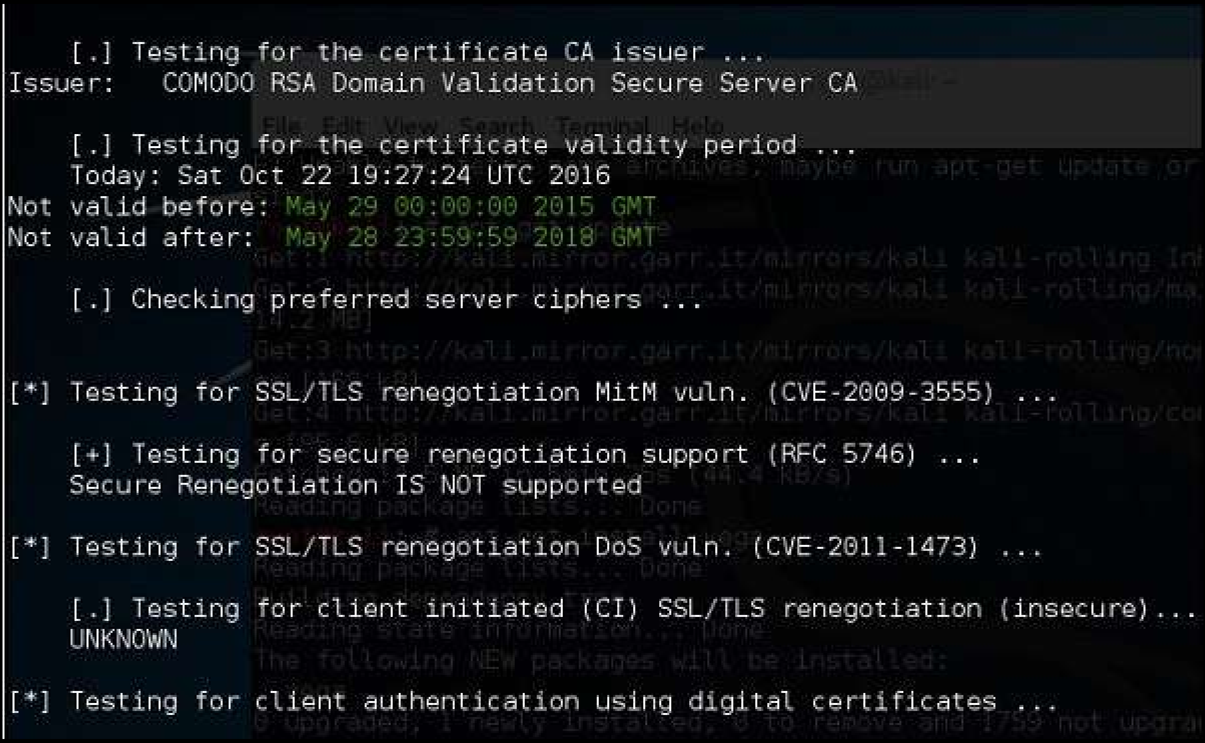
The current tests include checking if the target supports the SSLv2 protocol, the NULL cipher, weak ciphers based on their key length (40 or 56 bits), the availability of strong ciphers (like AES), if the digital certificate is MD5 signed, and the current SSL/TLS renegotiation capabilities.

To start testing, open a terminal and type "**tlssled URL port**". It will start to test the certificate to find data.



```
oot@kali:~# tlssled [redacted] 443
------------------------------------------------------
 TLSSLed - (1.3) based on sslscan and openssl
                 by Raul Siles (www.taddong.com)
------------------------------------------------------
    openssl version: OpenSSL 1.0.2f  28 Jan 2016
------------------------------------------------------
    Date: 20161022-152715
------------------------------------------------------

[*] Analyzing SSL/TLS on [redacted]:443 ...
    [.] Output directory: TLSSLed_1.3_[redacted]_443_20161022-152715 ...

[*] Checking if the target service speaks SSL/TLS...
    [.] The target service [redacted]:443 seems to speak SSL/TLS...

    [.] Using SSL/TLS protocol version:
        (empty means I'm using the default openssl protocol version(s))

[*] Running sslscan on [redacted]:443 ...

    [-] Testing for SSLv2 ...
```

You can see from the finding that the certificate is valid until 2018 as shown in green in the following screenshot.



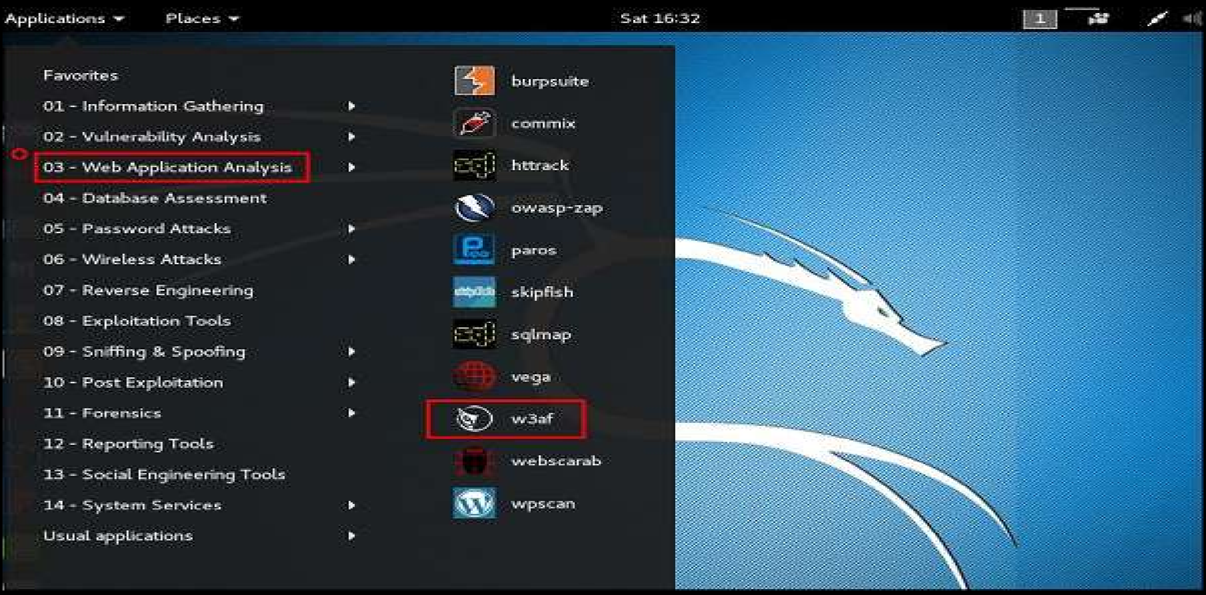
```
    [.] Testing for the certificate CA issuer ...
Issuer:   COMODO RSA Domain Validation Secure Server CA

    [.] Testing for the certificate validity period ...
    Today: Sat Oct 22 19:27:24 UTC 2016
Not valid before: May 29 00:00:00 2015 GMT
Not valid after:  May 28 23:59:59 2018 GMT

    [.] Checking preferred server ciphers ...


[*] Testing for SSL/TLS renegotiation MitM vuln. (CVE-2009-3555) ...

    [+] Testing for secure renegotiation support (RFC 5746) ...
    Secure Renegotiation IS NOT supported

[*] Testing for SSL/TLS renegotiation DoS vuln. (CVE-2011-1473) ...

    [.] Testing for client initiated (CI) SSL/TLS renegotiation (insecure)...
    UNKNOWN

[*] Testing for client authentication using digital certificates ...
```

w3af

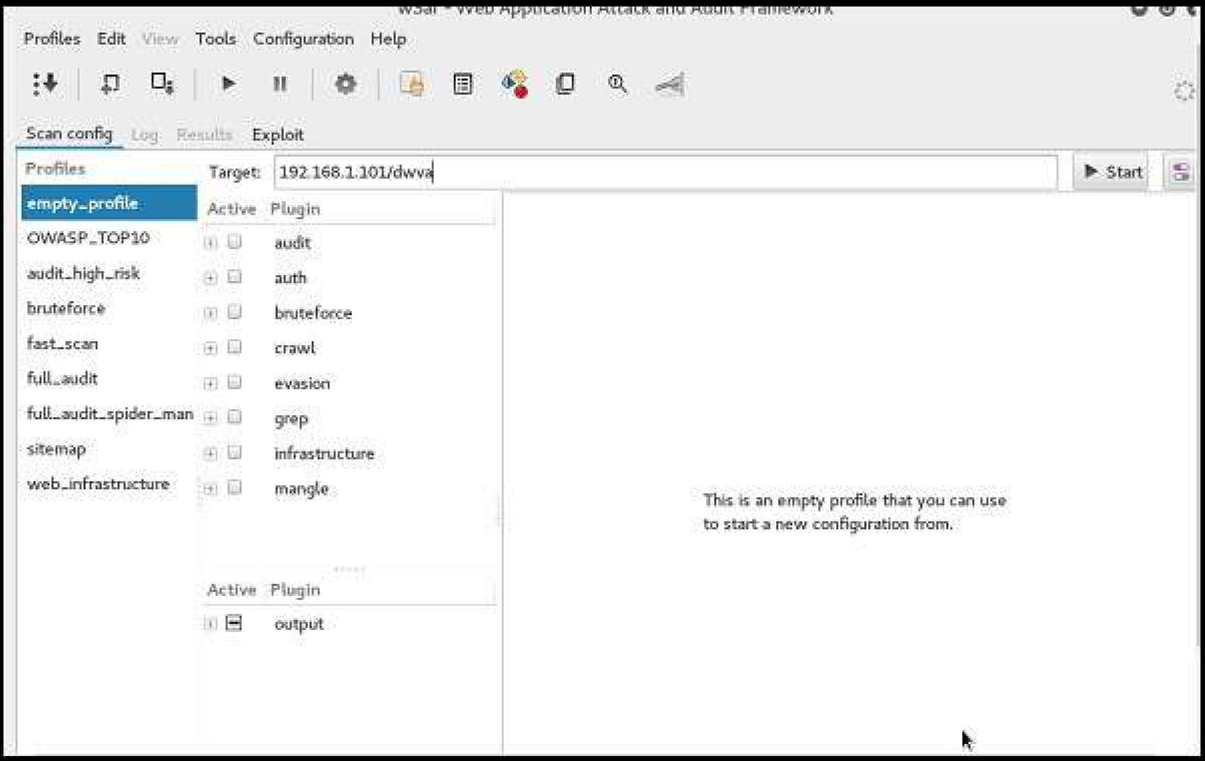
w3af is a Web Application Attack and Audit Framework which aims to identify and exploit all web application vulnerabilities. This package provides a Graphical User Interface (GUI) for the framework. If you want a command-line application only, install w3af-console.

The framework has been called the "metasploit for the web", but it's actually much more as it also discovers the web application vulnerabilities using black-box scanning techniques. The w3af core and its plugins are fully written in Python. The project has more than 130 plugins, which identify and exploit SQL injection, cross-site scripting (XSS), remote file inclusion and more.

Step 1 – To open it, go to Applications → 03-Web Application Analysis → Click w3af.



Step 2 – On the "Target" enter the URL of victim which in this case will be metasploitable web address.


w3af - Web Application Attack and Audit Framework
Profiles Edit View Tools Configuration Help
Scan config Log Results Exploit
Start

Target: 192.168.1.101/dwva

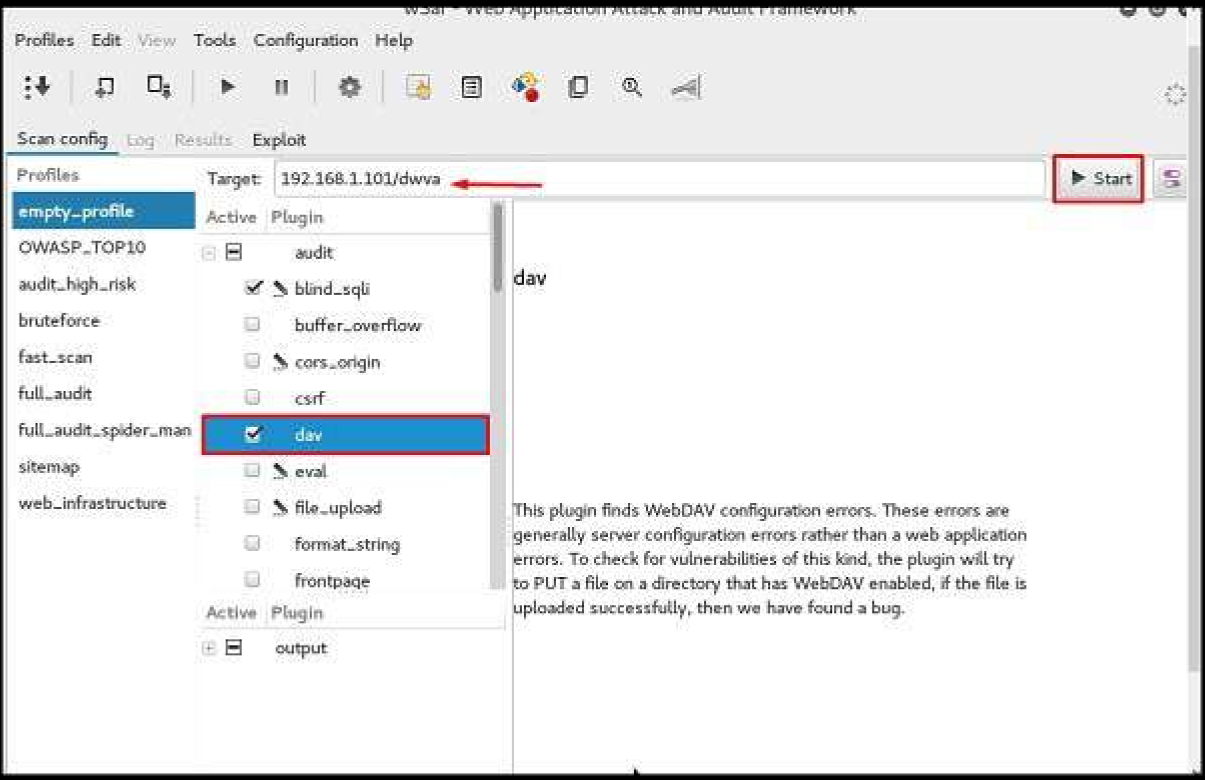
Profiles
empty_profile
OWASP_TOP10
audit_high_risk
bruteforce
fast_scan
full_audit
full_audit_spider_man
sitemap
web_infrastructure

Active	Plugin
□	audit
□	auth
□	bruteforce
□	crawl
□	evasion
□	grep
□	infrastructure
□	mangle

Active	Plugin
⊟	output

This is an empty profile that you can use to start a new configuration from.

Step 3 – Select the profile → Click "Start".


w3af - Web Application Attack and Audit Framework
Profiles Edit View Tools Configuration Help
Scan config Log Results Exploit
Start

Target: 192.168.1.101/dwva

Profiles
empty_profile
OWASP_TOP10
audit_high_risk
bruteforce
fast_scan
full_audit
full_audit_spider_man
sitemap
web_infrastructure

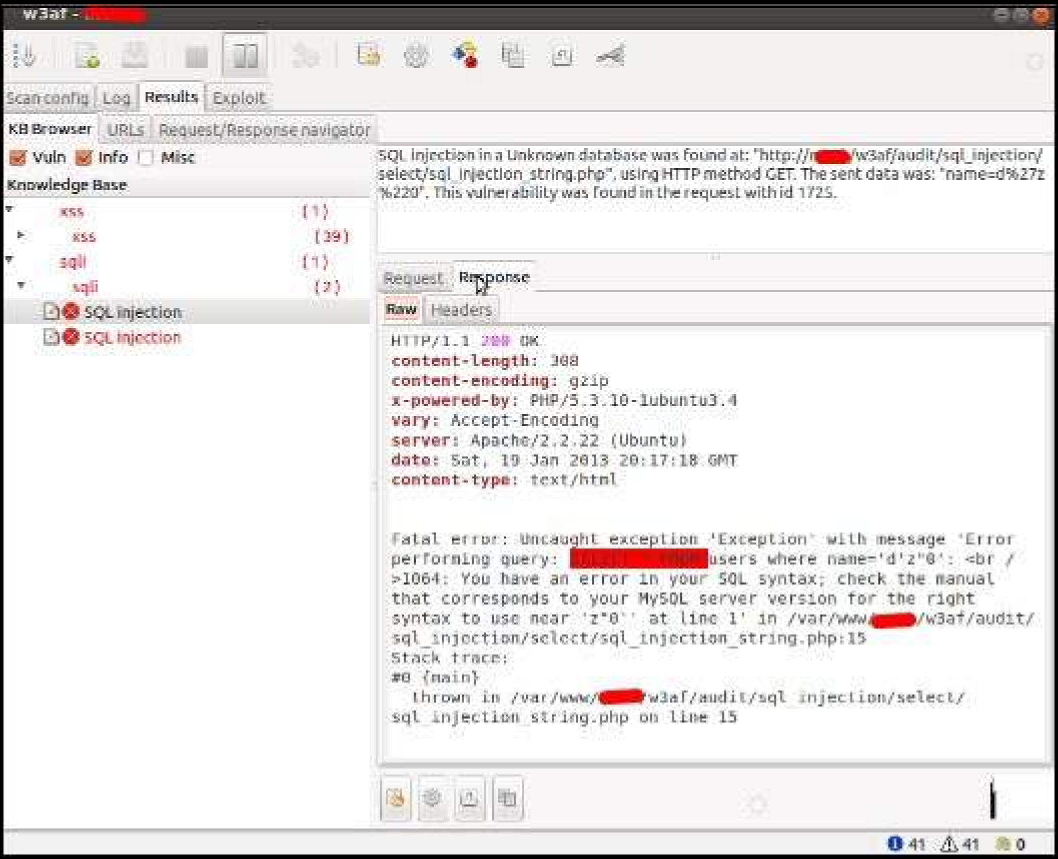
Active	Plugin
⊟	audit
✓	blind_sqli
□	buffer_overflow
□	cors_origin
□	csrf
☑	dav
□	eval
□	file_upload
□	format_string
□	frontpage

Active	Plugin
⊟	output

dav

This plugin finds WebDAV configuration errors. These errors are generally server configuration errors rather than a web application errors. To check for vulnerabilities of this kind, the plugin will try to PUT a file on a directory that has WebDAV enabled, if the file is uploaded successfully, then we have found a bug.

Step 4 – Go to "Results" and you can see the finding with the details.



SQL injection in a Unknown database was found at: "http://[redacted]/w3af/audit/sql_injection/select/sql_injection_string.php", using HTTP method GET. The sent data was: "name=d%27z%220". This vulnerability was found in the request with id 1725.

```
HTTP/1.1 200 OK
content-length: 308
content-encoding: gzip
x-powered-by: PHP/5.3.10-1ubuntu3.4
vary: Accept-Encoding
server: Apache/2.2.22 (Ubuntu)
date: Sat, 19 Jan 2013 20:17:18 GMT
content-type: text/html


Fatal error: Uncaught exception 'Exception' with message 'Error
performing query: [redacted]users where name='d'z"0': <br /
>1064: You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right
syntax to use near 'z"0'' at line 1' in /var/www[redacted]/w3af/audit/
sql_injection/select/sql_injection_string.php:15
Stack trace:
#0 {main}
  thrown in /var/www/[redacted]w3af/audit/sql_injection/select/
sql_injection_string.php on line 15
```