SOFTWARE TESTING FINAL REVIEW

Fall Semester 2021-22

TEAM: Muskan Rastogi - 18BIT0287

Akshat Gupta – 18BIT0330

Prateek Devashetti - 19BIT0229

SUBMITTED TO: Prof. Prasanna M

COURSE CODE: ITE2004

COURSE TITLE: Software Testing

PROJECT: Facebook Testing

GITHUB LINK:

https://github.com/akshatvg/Software-Testing--Black-Box--Facebook.com

Title: Testing the Facebook Website (facebook.com)

SRS Document:

Table of contents:

1. INTRODUCTION
This document is prepared in order to determine a software requirement specification for Facebook. Facebook is a social network on which people can add their friends, share videos and photos ,send and receive messages , comment and like on the links etc. In order to gain an overview about the report firstly , the purpose and scope of this document will be given , then an overall description of Facebook system is followed. In addition to these , system features such as uploading photo, sharing videos, adding friends , etc. are described deeply.

1.1 PURPOSE
The SRS is needed to evolve as the development of the software product processes. The purpose of this document is to give a complete description about how Facebook social network system can be developed. This document is to provide information about what the software product is to do to customers and establish an agreement between customers and suppliers and also become helpful for development. In addition to these, it provide a basis for validation and verification.

1.2 SCOPE
The name of the software product is Facebook. Facebook is a social network that connects people. The aim of Facebook is to provide information to the users about the events and the people whom they know. The users of Facebook can add friends, share videos which they want their friends watch; upload photos, comment on their friends' sharing's, chatting with their friends and become informed about their friends. Moreover, people can create social groups for such as university clubs, football clubs or for social awareness. People can be informed about the events by the help of these groups or their friends.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS
When the user logins Facebook, they can see their home page, which is named as "News Feed" that provide users to see what their friends share, what their friends write their status. Moreover, at the left of this page, the user can see the event invitations and the birthdays of their friends. Therefore, News Feed is the main page which combines daily friend interactions.

1.4 REFERENCES
1. www.facebook.com
2. www.wikihow.com

2. OVERALL DESCRIPTION
In this section, background information about what type of requirements the system should have will be provided briefly.

2.1 PRODUCT PERSPECTIVE
Facebook is an independent and world-wide social network website. Every person can use it online without a fee. The Facebook is not a part of a larger system, it is an independent system. People from different regions of the world can connect to it and exchange information with other people. In order to control the contents of the sharing's and comments done by the other people, Facebook has also a control mechanism. People can deliver their complaints about any part of the Facebook to the "Facebook Administrators". Then,

"Facebook Administrators" might take appropriate actions according to the complained situation which is against the rules.

2.2 PRODUCT FUNCTIONS
After creating an account and starting to use the Facebook, first thing he or she will make is searching for friends. The user will search people by their names and can send an invitation to them to add as a friend and to be able to see their shared items on Facebook. If the person accepts the invitation, these two persons become friends on Facebook and can interact more closely such as sending messages to each other. Any user can share his/her status like whatever he is thinking, wherever he is or his current mode. Friends of this person can make a comment on that. Furthermore, if a user shared a photo, video, link or anything, any friend of that user can share that shared item also. Users can upload photo and video to their profiles and create an album. Anyone can create a group and invite people to join in the group. Similarly, people can attend the activities where they are invited.

2.3 USER CHARACTERISTICS
Facebook does not require any specific computer knowledge to use it except the developers and administrators of it. Standard users are thought to be from any age, any gender
and from any nationality who can use just computer's browser. On the other hand,
administrators and potential developers need a high level of expertise to understand web technologies.

2.4 DESIGN & IMPLEMENTATION CONSTRAINTS
Being a social network website, the software should ensure the safety of information given by the user and provide some privacy settings options to the user. Firstly, Facebook provides people the right to choose the category of
people who will be able to view their shared items. Some users may not desire the access of some people to their shared items and information. If this is the case, users can set their privacy settings to prevent some people's access to their information. Secondly, Facebook cannot sell the private information of users to someone else.

3. External Interface Requirement

3.1 Interface Requirement
Various interfaces for the Facebook could be
1. Login Page
2. Home Page
3. There will be a screen displaying news feeds, friend request, suggestion,etc.

3.2 Hardware Interface
The System must run over the internet, all the hardware shall require to connect internet
will be hardware interface for the system. As for e.g. Modem,
WAN
- LAN, Ethernet Cross-Cable.

3.3 Software Interface

The system is on server so it requires the any scripting language like PHP, VBScript etc. The system require Data Base also for the store the any transaction of the system like MYSQL etc. system also require DNS (domain name space) for the naming on the internet. At the last user need web browser for interact with the system.

4. SPECIFIC REQUIREMENTS

In this section, all software requirements will be explain some information of the user.
d in detail. All requirements are divided into two groups as functional and non-functional.

4.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component.
F.R. 1 - Create Account
DESCRIPTION- If user is new and does not have Facebook account.
INPUT-Request for first name, last name , email id/phone no. and password.
PROCESSING-Retrieves the provided information and makes a new account for user.
OUTPUT- Displays created account.

F.R. 2- Searching Friends
DESCRIPTION- User searches for friends to add in friend list.
INPUT- Click on search and enter friend name.
PROCESSING- When user tap on 'search', search and show their name.
OUTPUT- Name of friend is showed.

F.R. 3- Sending Request
DESCRIPTION- User can send request to a person he/she wants to add in his/her friend list.
INPUT- Tap on 'add friend',
PROCESSING- When user tap on 'add friend', friend request sent displayed.
OUTPUT-Friend request sent.

F.R. 4 - Accepting Friend Request
DESCRIPTION- When someone sends friend request to user , user gets notification whether he wants to 'accept' or 'delete'
INPUT-Click on 'accept' button.
PROCESSING- User clicks on 'accept' and friend added to his/her friend list.
OUTPUT-Friend added in friend list.

F.R. 5 - Creating Groups
DESCRIPTION- One can create groups on facebook and add some peoples from his/her friend list.
INPUT- Click 'profile' at the top, click 'groups' ,click 'see all', tap 'create group' ,name group, add some people,add description , select privacy and click 'save'.
PROCESSING- When someone click on 'groups' transferred to 'see all'

and adds the name. people.description and selects privacy transferred to 'save'
OUTPUT-Group created.

F.R. 6 - Uploading Photos
DESCRIPTION- User can add photos on facebook to update his activities.
INPUT- Tap 'photo' ,select photo to upload ,tap done'.
PROCESSING- user clicks on 'photo' transferred to 'select', photo selected by user then transferred to tap 'done'.
OUTPUT-Photo uploaded successfully.

F.R. 7 - Creating Albums
DESCRIPTION- User can add facebook photos and create album
INPUT-Go to 'update status',create 'photo album',choose order of photos, choose album cover, choose privacy and post.
PROCESSING-
OUTPUT-Album Created.

F.R. 8 - Sharing Status
DESCRIPTION- When someone clicks Share below a post, they are able to share your photos, videos or status updates through Facebook.
INPUT-Search post to share, tap 'share' , tap 'write post' and tap 'share now.
PROCESSING- User clicks on share, transferred to write post and then clicks on 'share now' to share.
OUTPUT- Post shared by the user.

F.R. 9 - Create New Page
DESCRIPTION- One can create a new page on facebook to add his/her activities and connect to friends.
INPUT- Tap on 'pages',tap 'create page',tap 'get started', select name,select categories ,add cover photo,profile picture for page tap 'visit page.
PROCESSING- User tap on 'pages' ,transferred to 'create page' and then clicks 'get started' ,transferred to add name ,cover photo profile photo and then tap on 'visit page' to see the created page.
OUTPUT- A new page created

F.R. 10 - Sending Message
DESCRIPTION- User should be able to send instant message to any contact on his/her contact list. User should be notified when message is successfully delivered to recipient by displaying a tick sign next to message sent.
INPUT- Message typed.
PROCESSING- Message send to other user.
OUTPUT- Tick on Message

F.R. 11 - Send Attachments
DESCRIPTION- User should be able to send audio, video and images as attachments.
INPUT- File attached.

PROCESSING-Send to other side user.
OUTPUT-Tick on file.

F.R. 12 - Commenting
DESCRIPTION- Facebook comments are key to understanding how users engage with one's content.
INPUT- Click the comment link, type comment, press enter to publish.
PROCESSING- Post the comment on user's attachment.
OUTPUT- Comment attached.

F.R. 13 - Uploading Videos
DESCRIPTION- User can upload video in his/her account
INPUT- Click 'add video' button choose file.add description and post.
PROCESSING- User clicks on 'add video'. transferred to choose files and then clicks on 'post' to upload and then video uploaded.
OUTPUT-Video uploaded successfully.

F.R. 14 - Notes
DESCRIPTION- User can add notes
INPUT- Select 'more' at right of profile picture, click 'notes', click 'add notes' , drag file,create note and 'publish'.
PROCESSING- User clicks on 'more'
, choose notes and clicks to add
and then attaches file then click to create and 'publish'
OUTPUT- Note published.

F.R. 15 - Videos
DESCRIPTION- User can watch uploaded videos on Facebook.
INPUT-Search videos, click to watch.
PROCESSING- User searches for a video in 'search' and a list of videos related to search displayed and then click to watch.
OUTPUT- video watched by user.

F.R. 16 - Notifications
DESCRIPTION- Notifications are updates about activity on facebook.
INPUT-Tap the globe icon, click see all.
PROCESSING- user click on globe icon then list of notifications displayed.
OUTPUT-List of notifications displayed.

F.R. 17 - Edit Profile
DESCRIPTION- Asks to upload photo of user, add a nickname , a birth name , relationship, about you, etc. and modifies the profile.
INPUT- Upload photo,nickname,birth name , relationship ,about you, etc.
PROCESSING- Check the information and processes the request.
OUTPUT- Displays the information provided by user.

F.R. 18 - Update Profile
DESCRIPTION- User can update his profile picture, can add birthday nickname from function 'update profile'
INPUT- Asks to update user's profile picture, cover picture.education.etc.

PROCESSING- Check the information and processes the request.
OUTPUT- Updates profile as provided by user.

F.R. 19- Creating an Event
DESCRIPTION- User can Create a Facebook Event through his Page to connect to his audience..
INPUT- Click on 'create event' tab, choose name, add location and time choose which friends to invite.
PROCESSING- Check the information and process the request.
OUTPUT-Event created.

F.R. 20 - Privacy Setting
DESCRIPTION- User can properly manage Facebook privacy setting so that he/she known who is/isn't seeing his/her updates photos and more.
INPUT- Asks to establish checks like who can view profile, friend list
PROCESSING- Check the information and processes the request and applies to change user's account.
OUTPUT- Applied changes are reflected on user's account.

5. NON-FUNCTIONAL REQUIREMENTS

5.1 Security
The system use SSL (secured socket layer) in all transactions that include any other confidential passenger information. The system must automatically log out in all customers after a period of inactivity.
The system should not leave any cookies on the customer's computer containing the user's password, system's back-end servers shall only be accessible to authenticated administrators. Sensitive data will be encrypted before being sent over insecure connections like the internet.

5.2 Reliability
The system provides storage of all databases on redundant computers with automatic switch over. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

5.3 Availability
The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

5.4 Maintainability
A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that its maintainability can be done efficiently.

5.5 Portability
The application is HTML and scripting language based. So That end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is use this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops, and PDA etc.

Test Plan:

Software testing tools are often used to assure firmness, thoroughness and performance in testing software products. Unit testing and subsequent integration testing can be performed by software testing tools. These tools are used to fulfil all the requirements of planned testing activities. These tools also work as commercial software testing tools. The quality of the software is evaluated by software testers with the help of various testing tools. We intend to use Selenium as our means of automated testing. We will perform manual testing on iOS, Android, Safari, Chrome, Brave and Microsoft Edge to ensure the platform works fine on all OS and Browsers.

Introduction

- We are all beginners to both Software Testing so we will try to learn and incorporate everything onto our tests.

References:

- www.facebook.com
- https://ieeexplore.ieee.org/abstract/document/6530393
- https://ieeexplore.ieee.org/document/6918309
- https://ieeexplore.ieee.org/document/6612231
- https://ieeexplore.ieee.org/document/9152626
- https://ieeexplore.ieee.org/document/6518896
- https://ieeexplore.ieee.org/document/7740370
- https://ieeexplore.ieee.org/abstract/document/6258316
- https://ieeexplore.ieee.org/document/6595803

Test Items:

- Facebook website made on React and PHP.

Features to be Tested:

- All the features mentioned in the 4.1 Section of the SRS document will be tested.

Features Not to Be Tested:

- Facebook Marketplace.
- Payments on Facebook.

Approach:

- We will follow a Lean testing model.

Item Pass / Fail Criteria:

- If the UX is not user-friendly, test case will fail.
- If there is a usability issue, test case will fail.
- If there is a security vulnerability, test case will fail.
- If there is a logical vulnerability, test case will fail.
- In all normal conditions, test cases will pass.

Suspension Criteria and Resumption Requirements:

- Testing will pause during college exams.

Test Environment:

- All the requirements mentioned in the 3 Section of the SRS document will be used.

Schedule:

- We will follow the deadlines scheduled by our professor to decide the schedule. There will be 3 major project reviews for the same.

Responsibilities:

- Each of us will learn, work on documentation and implement our learnings for smooth testing.

Risks:

- Security risks have been potentially thought of.
- Usability issues have been found.

Approvals:

- Our faculty must approve the plan.

For Review 2 and 3:
The various testing and their test cases are as follows:

<u>LOGIN/ SIGN-UP FLOW</u>

What is CSRF?

Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform. It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other.

What is the impact of a CSRF attack?

In a successful CSRF attack, the attacker causes the victim user to carry out an action unintentionally. For example, this might be to change the email address on their account, to change their password, or to make a funds transfer. Depending on the nature of the action, the attacker might be able to gain full control over the user's account. If the compromised user has a privileged role within the application, then the attacker might be able to take full control of all the application's data and functionality.

How to construct a CSRF attack

Manually creating the HTML needed for a CSRF exploit can be cumbersome, particularly where the desired request contains a large number of parameters, or there are other quirks in the request. The easiest way to construct a CSRF exploit is using the CSRF PoC generator that is built in to Burp Suite Professional:

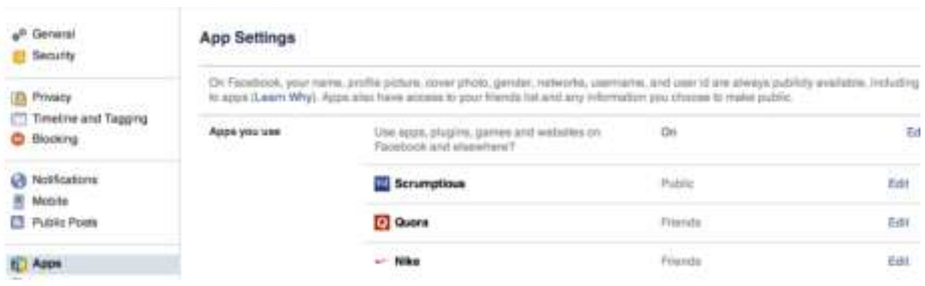
- Select a request anywhere in Burp Suite Professional that you want to test or exploit.
- From the right-click context menu, select Engagement tools / Generate CSRF PoC.
- Burp Suite will generate some HTML that will trigger the selected request (minus cookies, which will be added automatically by the victim's browser).
- You can tweak various options in the CSRF PoC generator to fine-tune aspects of the attack. You might need to do this in some unusual situations to deal with quirky features of requests.
- Copy the generated HTML into a web page, view it in a browser that is logged in to the vulnerable web site, and test whether the intended request is issued successfully and the desired action occurs.

How does facebook handle this issue

So earlier facebook did not have CSRF tokens due to which they had to pay $25,000 as a penalty, due to which a lot of users data was retraced and being hacked.

But the current system has CSRF tokens because of which we can't use xss to login through iframes or scripts from other sites
POST /[business_id]? fields=... **Facebook's Graph API is protected against CSRF by requiring a valid access token from the user**. Without this token, the request is un-authenticated.

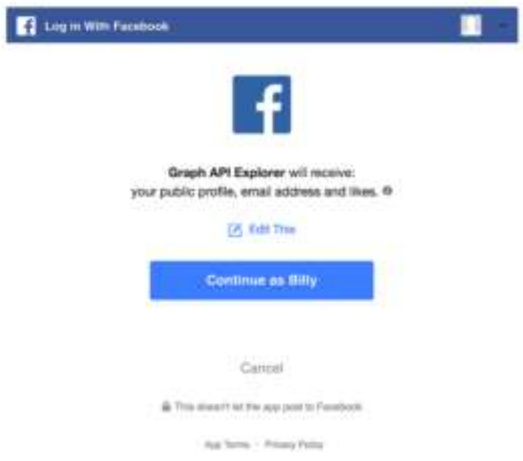
Test cases for Login Flow



App Settings

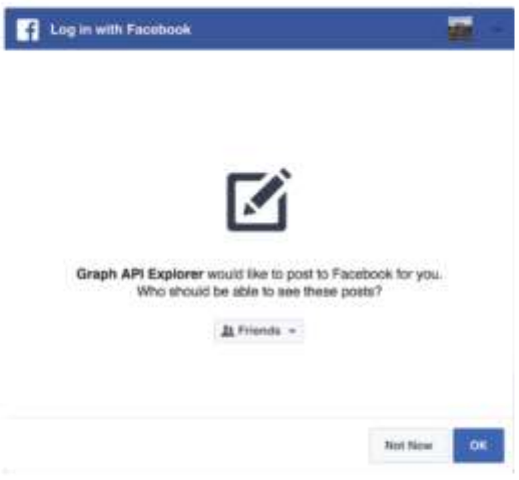
On Facebook, your name, profile picture, cover photo, gender, networks, username, and user id are always publicly available, including to apps (Learn Why). Apps also have access to your friends list and any information you choose to make public.

Apps you use	Use apps, plugins, games and websites on Facebook and elsewhere?	On	Ed
	Scrumptious	Public	Edit
	Quora	Friends	Edit
	Nike	Friends	Edit



Graph API Explorer will receive:
your public profile, email address and likes.

This doesn't let the app post to Facebook



Graph API Explorer would like to post to Facebook for you.
Who should be able to see these posts?

It’s important to test and verify that your Facebook Login flow works well under a variety of conditions. To test your Login flow, first create a separate Facebook user account:

1. Create a new test user account with Facebook
2. Log into Facebook with your test user credentials

1. Someone new to your app logs in with Facebook

1. Go to your app and tap on the **Log in with Facebook** button
2. Tap OK to accept the read permissions
3. Click OK again to accept write permissions if applicable
4. Go to app settings and verify that the granted permissions are there

2. Someone logs in with Facebook after previously logging in via a non-Facebook flow with the same email address

1. Go to your app and login using your email address
2. Log out of your app and tap on the "Log in with Facebook" button
3. Tap OK to accept the read permissions (and OK again to accept write permissions where applicable)
4. Go to app settings on Facebook and verify that the granted permissions are there

3. Someone who has logged into your app with Facebook in the past logs back in

1. Go back to your app and tap on the “Log in with Facebook” button
2. Tap OK to accept the read permissions (and OK again to accept write permissions where applicable)
3. Uninstall then re-install your app
4. Open your app and tap on the “Log in with Facebook” button
5. Verify that you can log in without seeing any permission dialogs

4. Someone cancels log in with Facebook and tries to log in again

1. Go to your app and tap on the "Log in with Facebook” button
2. Verify that the read permissions are shown and tap “Cancel”
3. Open your app and tap on the "Log in with Facebook” button
4. Verify that the read permissions are shown again

5. Someone removes your app from Facebook via app settings and revisits your app. Your app should detect this and prompt the person to log back in.

1. Go to your app and tap on the "Log in with Facebook” button

2. Tap OK to accept the read permissions (and OK again to accept write permissions where applicable)
3. Go to app settings on Facebook and remove your app
4. Repeat steps 1-2 and verify that Facebook Login works

6. Someone changes the Facebook password after logging in with Facebook to your app

In this case, your token will be invalid and you should notify users that their Facebook session has expired and ask them to log in again.

1. Change your Facebook password and select “Log me out of other devices”
2. Go to your app and tap on the "Log in with Facebook” button
3. Tap OK to accept the read permissions (and OK again to accept write permissions where applicable)
4. Go to app settings on Facebook and verify that the granted permissions are there

7. Someone disabled Facebook platform via app settings and logs in to your app

In this case, you should make sure your app detects the error so that it can notify users and redirect them to the non-iOS integrated version of Facebook Login.

1. Turn off platform for your test user via app settings
2. Go to your app and tap on the "Log in with Facebook” button
3. Tap OK to accept the read permissions (and OK again to accept write permissions where applicable)
4. Verify that platform is now turned on and the app is added to your test user profile with correct privacy

8. Someone revisits your app when your app token has expired.

Please read our guide on handling token expiration

9. For games that want to sync their status across multiple devices, test your syncing state

1. Login with Facebook on your app and play your app’s game until you reach a certain level X
2. Login with Facebook on a different device via the same or different operating systems, and test that level X remains

Functional Test Cases

Sr.No	TestCase_ID	TestCase_Objective

1	Login_01	To verify Login functionality with valid email id and valid password.
2	Login_02	To verify Login functionality with valid email address and invalid password.
3	Login_03	To verify Login functionality with invalid email address and valid password.
4	Login_04	To verify Login functionality with invalid email address and invalid password.
5	Login_05	To verify Login functionality with blank email address and valid password.
6	Login_06	To check that Login functionality with valid email id and blank password.
7	Login_07	To verify Login functionality with blank email address and blank password.
8	Login_08	To check that Login functionality with valid phone number and valid password.
9	Login_09	To check that Login functionality with valid phone number and invalid password.
10	Login_10	To check that Login functionality with invalid phone number and valid password.
11	Login_11	To check that Login functionality with invalid phone number and invalid password.
12	Login_12	To check that Login functionality with blank phone number and valid password.
13	Login_13	To verify that length of email address field and password field.

14	Login_14	To verify that error message display when any field is left blank.
15	Login_15	To verify Tab key functionality on the Login page.
16	Login_16	To verify that remember me checkbox functionality
20	Login_20	To verify that entered multiple times incorrect passwords.
21	Login_21	To verify that welcome message after successfully login into application.
22	Login_22	To verify that Forgotten Password functionality.
23	Login_23	To verify if password text format is encrypted or not into password field.
24	Login_24	To verify that back button of the browser after log out.
25	Login_25	To check that message for entered invalid inputs.
26	Login_26	To verify that Login button with click and Enter key events.

UI Testing

Generally, in UI testing, we are testing the layouts, text fields, radio buttons, checkbox, and drop-down list. The Login page should have an easy interface for users. So users can easily interact with the application.

1	Login_UI_1	To check Login page layout is as per specification or not.
2	Login_UI_2	To check that placeholders are displayed properly or not.
3	Login_UI_3	To check that red (*) mark is properly displayed for mandatory fields or not.

4	Login_UI_4	To check that text fields,buttons,checkbox are displayed as per specification or not.
5	Login_UI_5	To check that the Login page is responsive or not.

Security Testing

1	Login_Security_01	To verify that SSL certificate is implemented or not.
2	Login_Security_02	To verify that “Back” button of the browser after successfully logged out from the application.
3	Login_Security_03	To verify that a user is able to login by directly entering url in the browser or not.
4	Login_Security_04	To verify that login session timeout functionality.
5	Login_Security_05	To verify that password format should be encrypted or not.
6	Login_Security_06	To verify that a user is able to enter more than characters as per specified into Email and Password fields or not.

Sign-up Test Cases

SR.No	**Testcase_ID**	**Testcase_Objective**
1	SignUp_01	To verify that all required fields are present in the form.
2	SignUp_02	To verify that all required text fields have a valid placeholder.
3	SignUp_03	To verify that all required fields have a red * mark.
4	SignUp_04	To verify tab key functionality on the sign-up page.

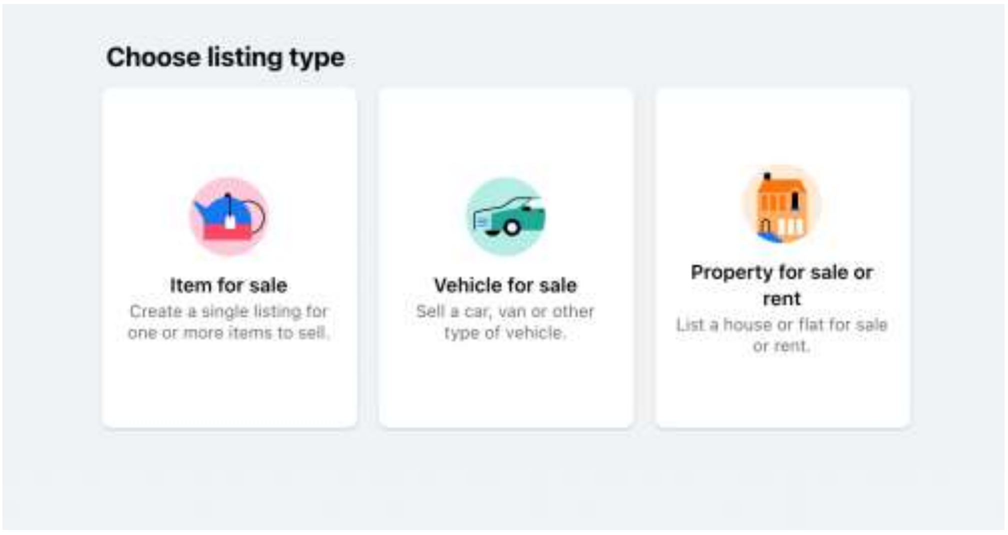
5	SignUp_05	To verify Enter Key functionality for sign up button.
6	SignUp_06	To verify sign-up page load with none of the data into the text field.
7	SignUp_07	To verify the lower and upper limit of the Firstname text field.
8	SignUp_08	To verify the Firstname field without any data.
9	SignUp_09	To verify the lower and upper limits of the Lastname text field.
10	SignUp_10	To verify the Last name field without any data.
11	SignUp_11	To verify lower and upper limit numbers of Mobile number fields.
12	SignUp_12	To verify alpha characters allow in the Mobile number field.
13	SignUp_13	To verify special characters allowed in the Mobile number field.
14	SignUp_14	To verify the Mobile number field without any data.
15	SignUp_15	To verify the mobile number with the area code.
16	SignUp_16	To verify validation message for incorrect mobile number
17	SignUp_17	To verify email id text without @ Symbol
18	SignUp_18	To verify the email id field with Special characters.
19	SignUp_19	To verify email id field with garbage email id.
20	SignUp_20	To verify email id field with already registered email id.

21	SignUp_21	To verify validation message for incorrect email id
22	SignUp_22	To verify password field with lower and upper characters length.
23	SignUp_23	To verify password field with lower and upper characters length.
24	SignUp_24	To verify Password field with only strings
25	SignUp_25	To verify Password field with only enter numbers
26	SignUp_26	To verify Password field with only enter special characters
27	SignUp_27	To verify the Password field with alphanumeric characters.
28	SignUp_28	To verify Password field with alpha and special characters
29	SignUp_29	To verify the Password field with numeric and special characters.
30	SignUp_30	To verify the Password field without data.
31	SignUp_31	To verify validation message for incorrect password format
32	SignUp_32	To verify confirm password field with different password.
33	SignUp_33	To verify confirm password field with the same password
34	SignUp_34	To verify confirm password field without data.
35	SignUp_35	To verify validation messages for different passwords.
36	SignUp_36	To verify that select month from month drop downlist from DOB

37	SignUp_37	To verify that select date from Date drop downlist from DOB
38	SignUp_38	To verify that select Year from Year drop downlist from DOB
39	SignUp_39	To verify that select the Female radio button from gender.
40	SignUp_40	To verify that select the Male radio button from gender.
41	SignUp_41	To verify that select the Custom radio button from gender.
42	SignUp_42	To verify the sign-up button without filling up any data on the sign-up page.
43	SignUp_43	To verify the sign-up button by entering the key and click the event.
44	SignUp_44	To verify user gets a successful signup message or not.
45	SignUp_45	To verify sign up button with multiple click events

Usability Testing

1. While creating a post for facebook market place, we don't get to categories out the product which makes it super difficult to filter and find the right product.

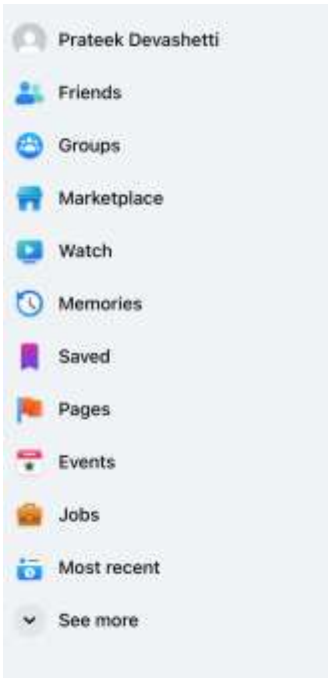

Choose listing type

Item for sale
Create a single listing for one or more items to sell.

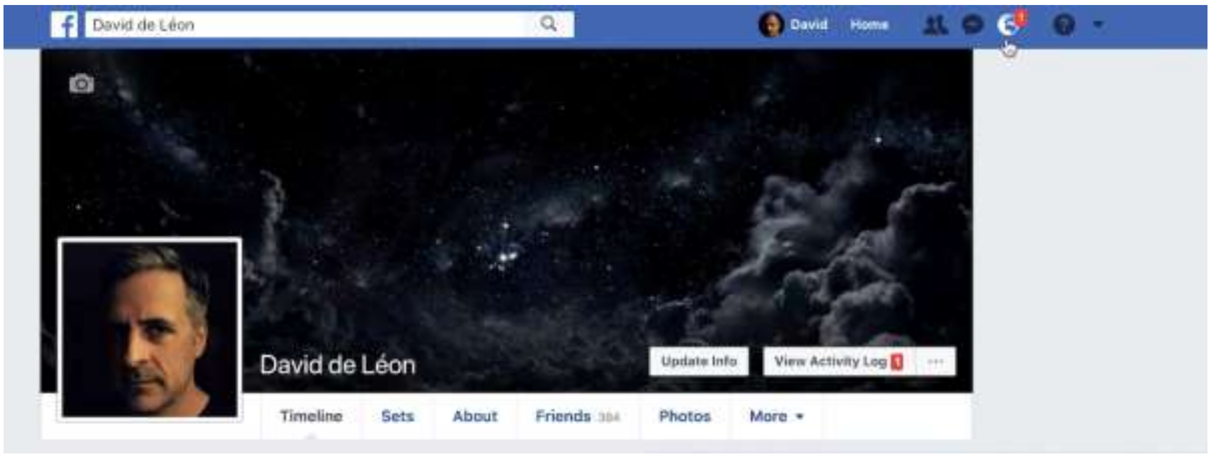
Vehicle for sale
Sell a car, van or other type of vehicle.

Property for sale or rent
List a house or flat for sale or rent.

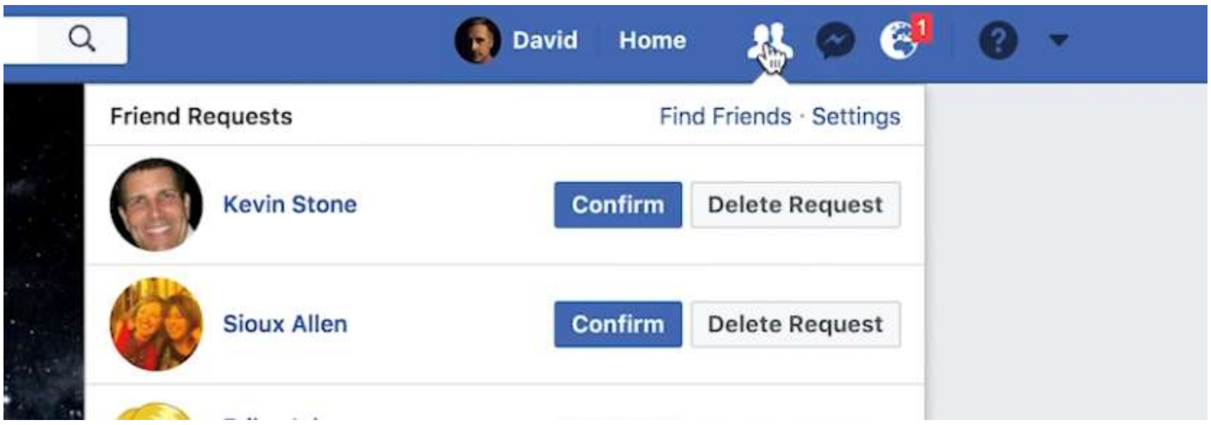
2. The navigation drawer has design flaws due to which it creates a cognitive load on the users due to which there are chances that the user might not be able to find they are looking forward to.



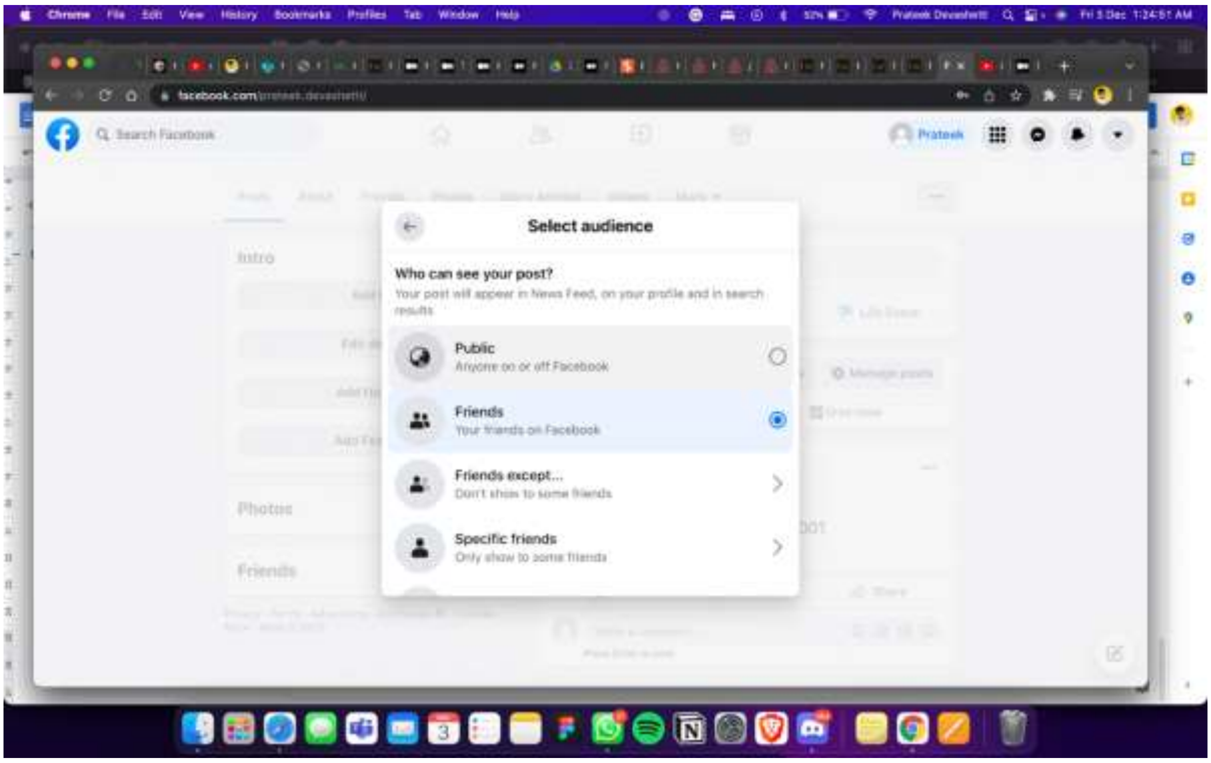
3. Not settings visible in the navigation bar therefore users isnt sure where he can visti to change account settings and all`



4. Settings option is not put in the right information architecture, which creates an impression of a not well defined site-mapping



5. The users are not asked for this pop-up condition before posting any sort of content onto the screen.



Select audience

Who can see your post?

Your post will appear in News Feed, on your profile and in search results

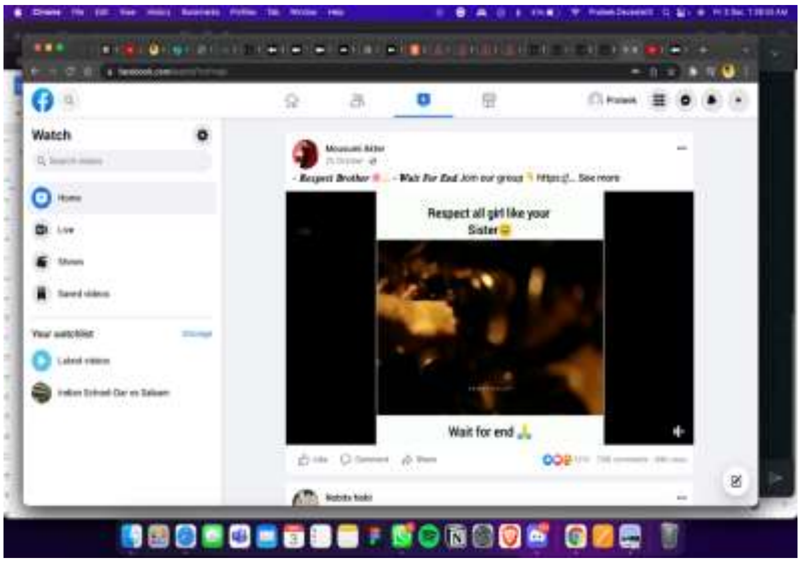
Public
Anyone on or off Facebook

[✓ Friends
Your friends on Facebook]

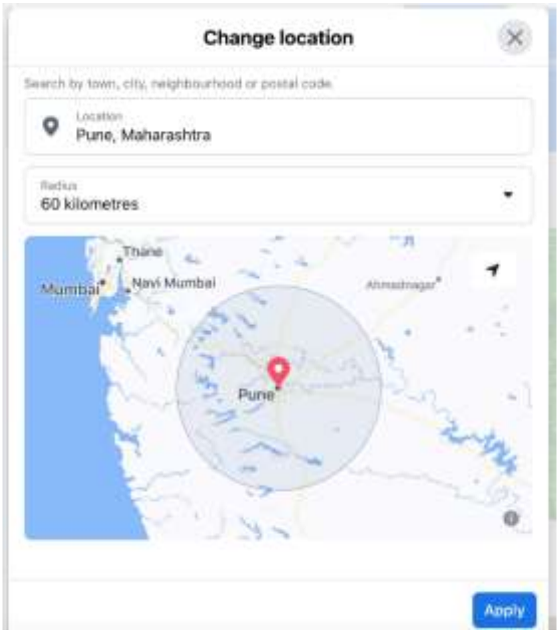
Friends except...
Don't show to some friends

Specific friends
Only show to some friends

6. Facebook does not allow the users to choice the type of content they would like to watch instead it just provides us with the most popular video on platform.



7. As marketplace is the main selling point of facebook, still location services wasn't asked to us when we joined in.

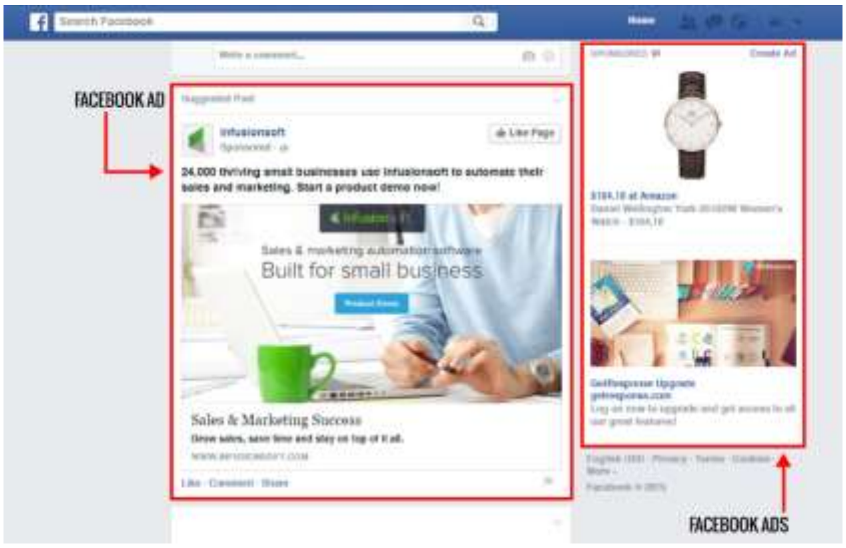

Change location

Search by town, city, neighbourhood or postal code

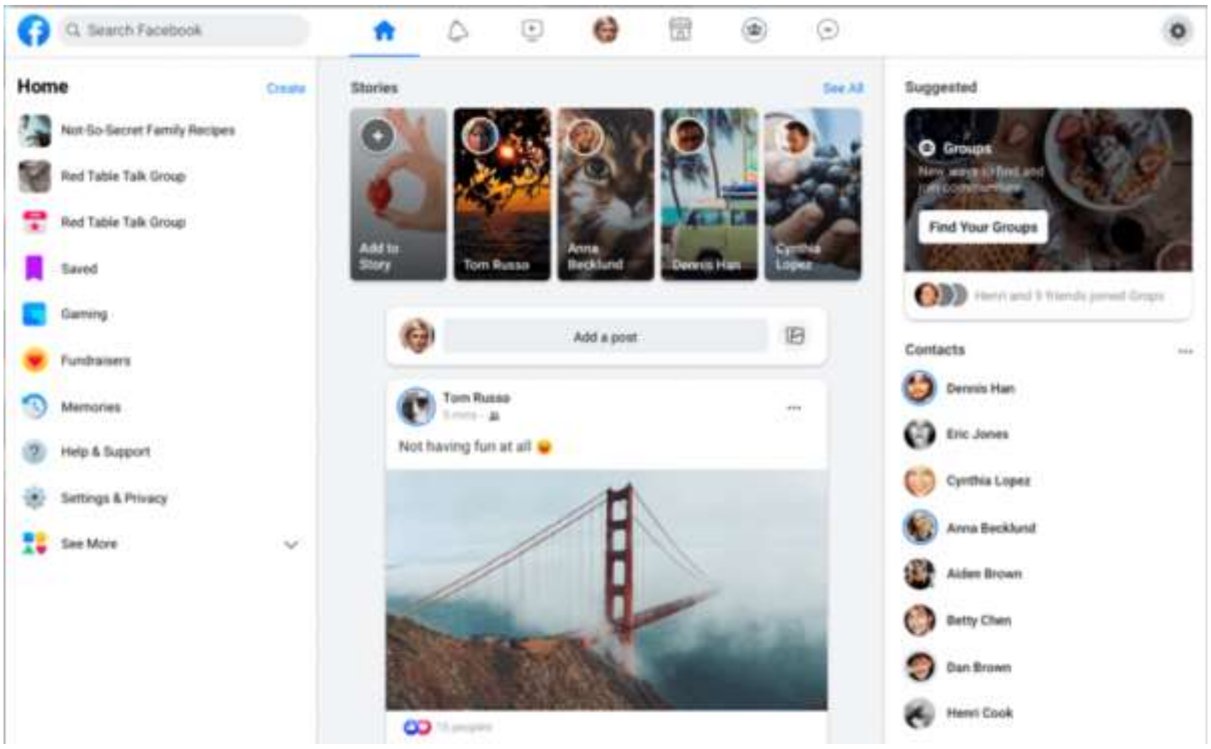
Location
Pune, Maharashtra

Radius
60 kilometres

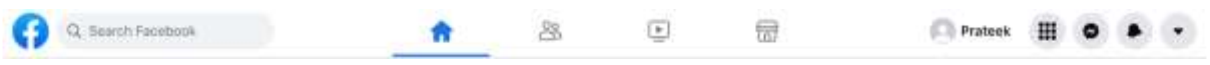
8. Facebook ads cover-up majority of the screen space due to which the users might get distracted.


FACEBOOK AD
Infusionsoft
24,000 thriving small businesses use Infusionsoft to automate their sales and marketing. Start a product demo now!
Sales & marketing automation software
Built for small business
Sales & Marketing Success
Like · Comment · Share
$184.18 at Amazon
FACEBOOK ADS

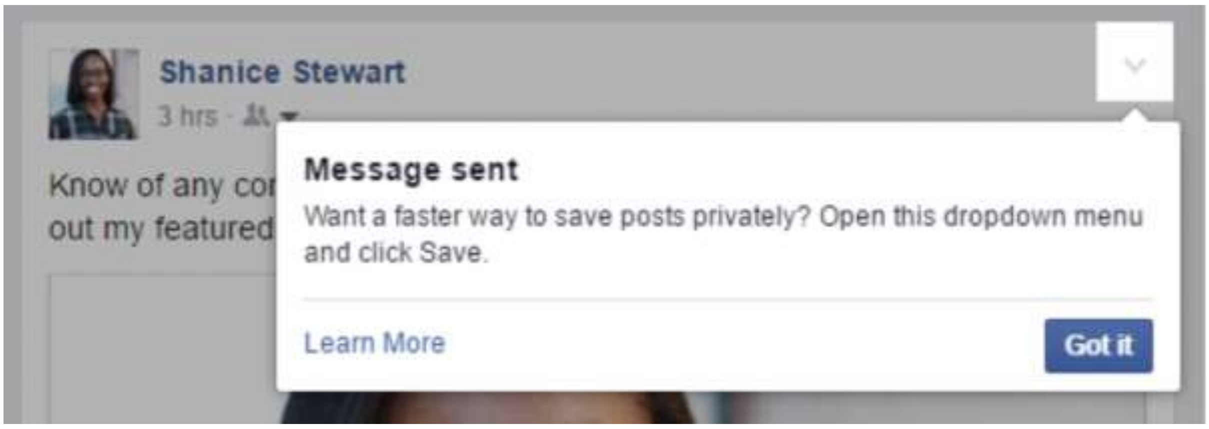
9. Iconography of the website does not really make sense when it come to segregation



10. Conventional websites design suggest that the search bar should be on the top left so that the users does not have to provide extra effort and it has more psychological effect than the visual aspect.



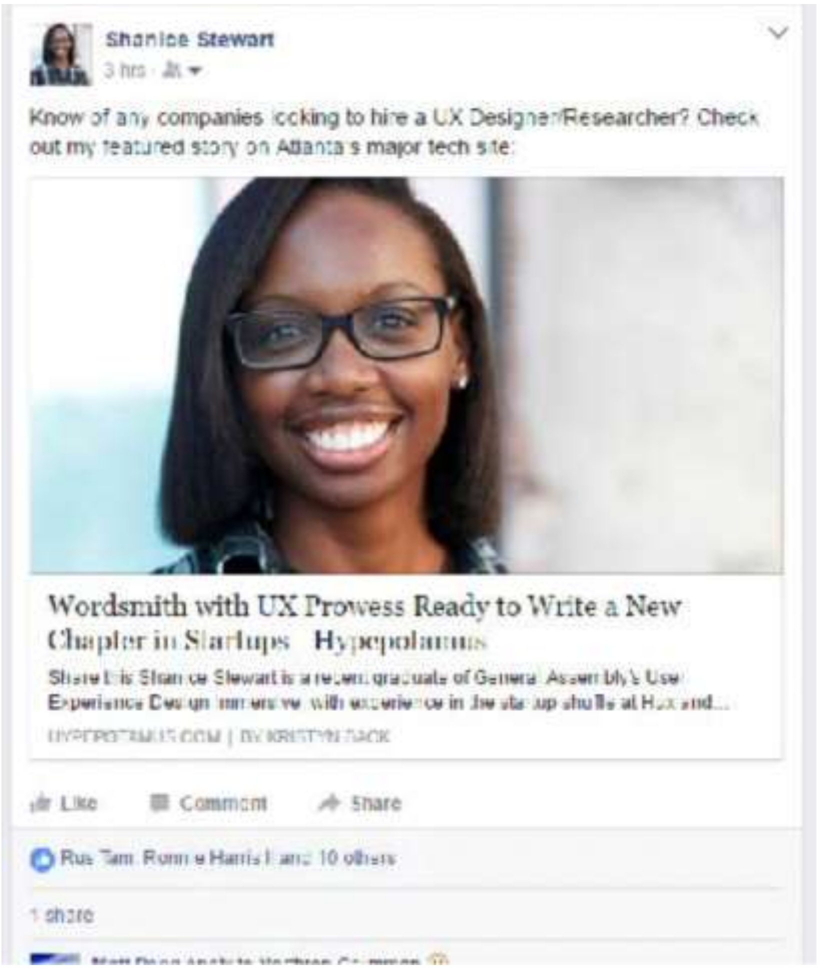
11. Quick access icons for faster and better engagement, as in the current system it is more time consuming and requires more interaction



Message sent

Want a faster way to save posts privately? Open this dropdown menu and click Save.

12. Save and like on the same level as it would help with faster accessibility.



Shanice Stewart
3 hrs

Know of any companies looking to hire a UX Designer/Researcher? Check out my featured story on Atlanta's major tech site:

Wordsmith with UX Prowess Ready to Write a New Chapter in Startups   Hypepotamus

Share this Shanice Stewart is a recent graduate of General Assembly's User Experience Design immersive with experience in the startup shuttle at Hux and...

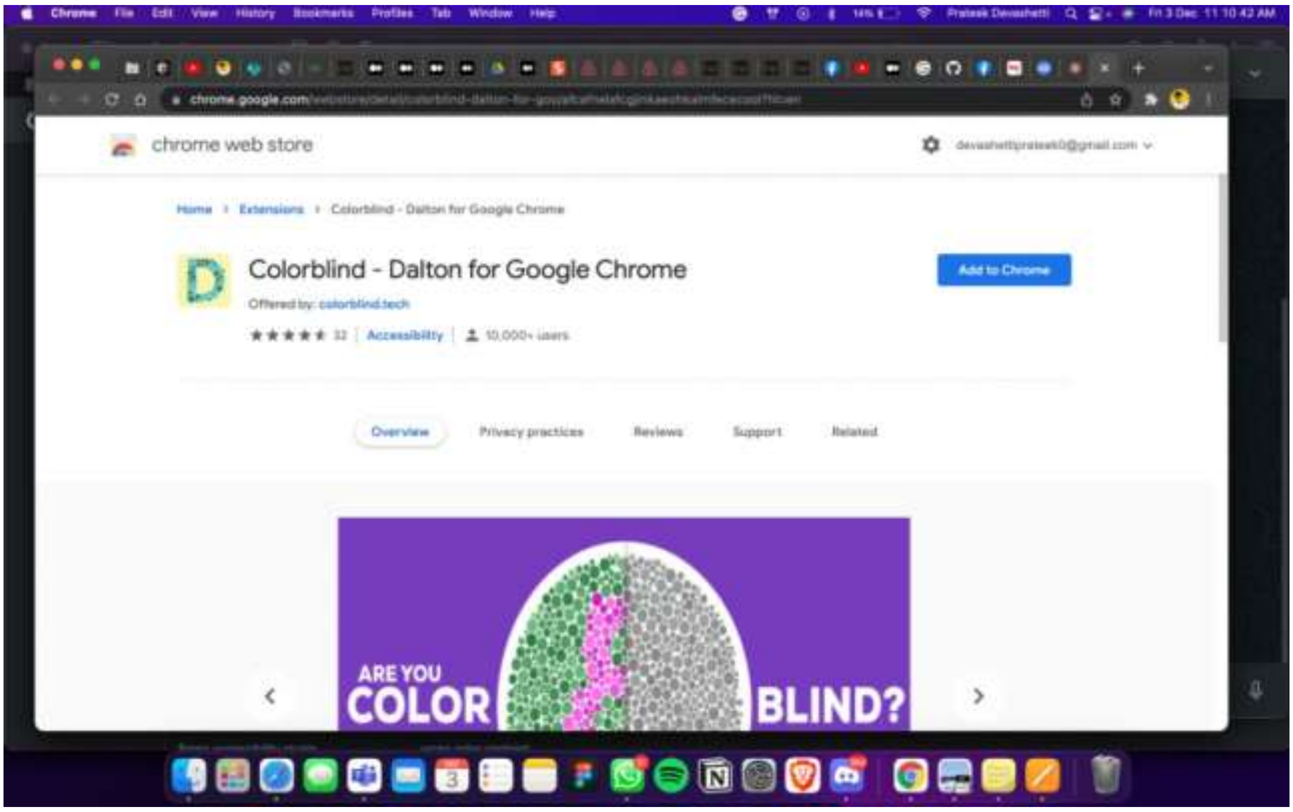
HYPEPOTAMUS.COM | BY KRISTYN DACK

Rus Tam, Ronnie Harris II and 10 others

1 share

13. Facebook currently aims for an inclusive audience which can be made better by using a fixed set of icons and colors as we can use different set of plugins to convert such websites so that it is accessible for a bigger audience.

14. Save a post using Facebook's save feature**.** At first, no one knew how to do this. The first sentiment 75% of the time was "I don't think you can do that." One tester went through the flow for *sharing* the post with themselves. One figured it out by clicking the faint grey arrow in the top right hand corner of the post. Others stared blankly at the screen as if to say "Help." I would then ask them to touch the same arrow. From there, they would lead themselves through to save the post.


chrome web store
Home › Extensions › Colorblind - Dalton for Google Chrome
Colorblind - Dalton for Google Chrome
Offered by: colorblind.tech
Add to Chrome
Accessibility
10,000+ users
Overview
Privacy practices
Reviews
Support
Related
ARE YOU COLOR BLIND?

15. The current system has multiple constraint layouts and auto scroll due to which the cognitive load shifts in fraction of seconds and this creates a rather odd interaction of the user with the platform.

<u>Bug Bounty</u>

Issue No. 1

Impact

It leads to page admin disclosure which is a serious issue.

A page admin will post a story to their personal account instead of the Page when messaging from the Page inbox on FBLite and hitting "Add to Story"

Steps

1.UserA sends a photo to UserB through the page.

2. UserA clicks on Add to story option

3. The photo is added to the UserA's story instead of Page's story which is leading to page admin disclosure.

Issue No. 2

Steps :

1. User A goes to his PageX's inbox through fblite and sees UserB's message thread

2. UserA messages to User B

3.User B receives the text message done by UserA through page's id

4. UserA now sends photo to UserB through the page inbox.

5. UserB receives the photo message through UserA's personal profile id instead of the page id which leads to page admin disclosure.

Noticed Facebook bugs:

Bug 1

My friend had started a Facebook page to post funny videos.

One video was very funny. I knew his fb id and also that he is the admin of the page.

Example admin id- xxxx

While viewing a video, I simply right clicked, View Page Source, searched xxxx.

Boom! One result found.

The page source was leaking the id of the person who was the content owner.

Bug 2

Users could have made a messenger call to the victim's account and then receive the call from the victim's locked Android phone to use the 'Watch Together' feature from the call screen without unlocking the phone thus allowing the intruder to get access to all of the saved videos & Watch History of the Facebook user. So, basically; the vulnerability here was that Facebook was allowing users to use such a sensitive feature like Watch Together even from a locked state of the device.

Facebook patched this one along with similar such vulnerabilities by asking first to unlock the phone before using such sensitive features from a locked Android phone.

Bug 3

We can create fundraisers for nonprofits and personal causes on Facebook. We can add our friends as organizers in the fundraisers we created. They need to approve our invitation to become an organizer. Once they approved the invitation an attacker was able to block the victim and thus victim was unable access the fundraiser and remove themself as an organizer from the fundraiser. An attacker could use this for their personal benefits.

Steps:

1)Attacker creates a fundraiser and invites victim as an organizer

2) Once victim accepts the invitation, attacker blocks the victim..

3) Victim is now unable to access the fundraiser but others can still see the victim as an organizer of the fundraiser.

Bug 4

Facebook paid a $25,000 bounty for a critical cross-site request forgery (CSRF) vulnerability that could have been exploited to hijack accounts simply by tricking users into clicki on a link.

"This bug could have allowed malicious users to send requests with CSRF tokens to arbitrary endpoints on Facebook which could lead to takeover of victims accounts. In order for this attack to be effective, an attacker would have to trick the target into clicking on a link."

Bug 5

Suppose there are 3 users A, B, and C. Here, A and B are friends on Facebook. The user B has blocked user C. Now user A wants to know whether user B has blocked user C or not. But how? Let's see the below 2 case possibilities.

1. If user A is accessing Facebook using mobile site: Here, he needs to make a post.

- The user A will go to the profile of user B. For example, User A will go to *m.facebook.com/b*
- Then user A will post the URL of the user C (For example, “https://www.facebook.com/c”) to the B’s profile *as a post* via mobile site.

Bug 6

If user A is accessing Facebook using a computer site: Here, he doesn’t need to make a post.

- The user A will go to the profile of user B. For example, User A will go to *“https://www.facebook.com/b”*
- The user A will paste the URL of the user C (For example, “https://www.facebook.com/c”) to the B’s profile in the box where we post status.
- Now he/she(user A) just needs to hit the preview button. Now user A will not be able to see the preview of that one.

But if B doesn’t block C then user A will able to see the preview.

VARIOUS OTHER TESTING TOOLS AND TEST CASES

Black box testing using different tools for the Facebook website to find bugs and make test cases.

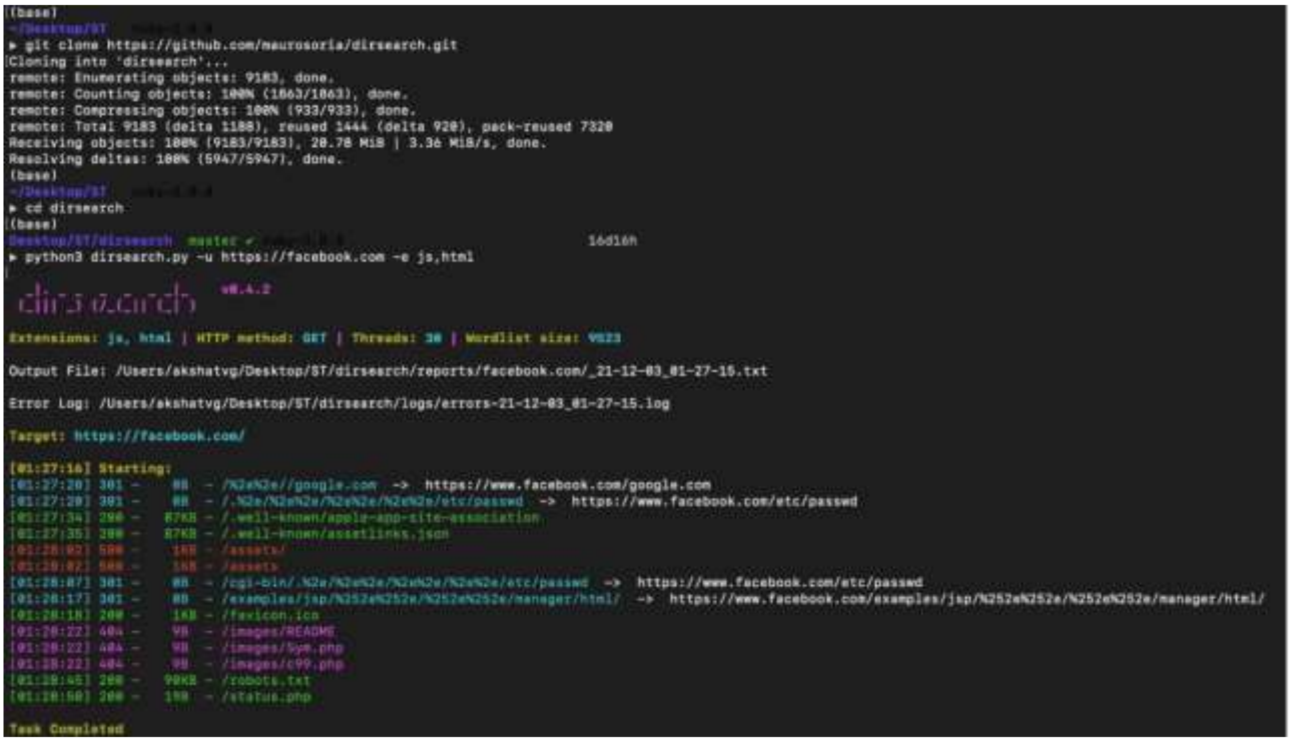
White box testing techniques analyse the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

Since we do not have access to the code of Facebook, we are only conducting black box (functionality) testing.

To check out the following testing, please check the github link:
https://github.com/akshatvg/Software-Testing--Black-Box--Facebook.com

Dirsearch

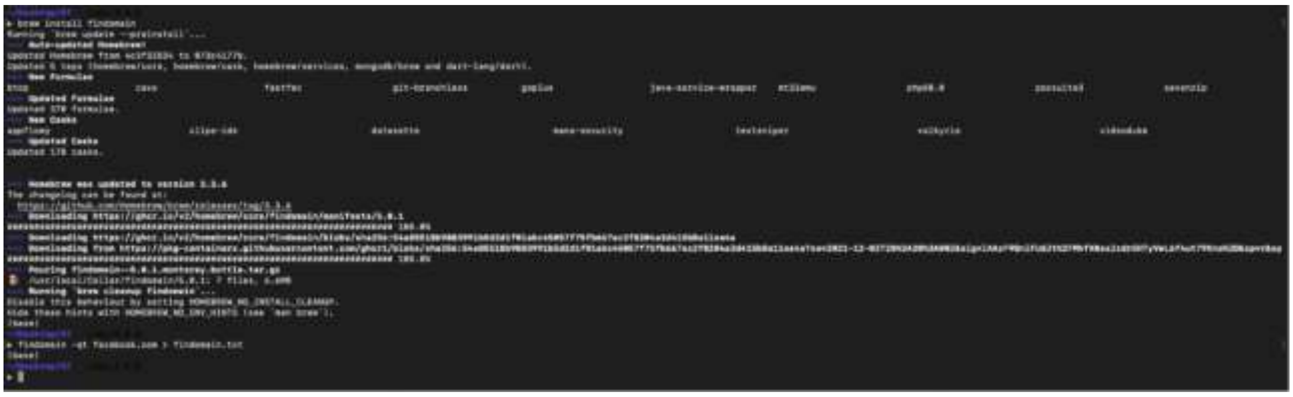
```
git clone https://github.com/maurosoria/dirsearch.git
cd dirsearch
python3 dirsearch.py -u https://facebook.com -e js,html
```
OUTPUT: Output File: /Users/akshatvg/Desktop/ST/dirsearch/reports/facebook.com/_21-12-03_01-27-15.txt



```
(base)
~/Desktop/ST
▶ git clone https://github.com/maurosoria/dirsearch.git
Cloning into 'dirsearch'...
remote: Enumerating objects: 9183, done.
remote: Counting objects: 100% (1863/1863), done.
remote: Compressing objects: 100% (933/933), done.
remote: Total 9183 (delta 1188), reused 1444 (delta 920), pack-reused 7320
Receiving objects: 100% (9183/9183), 20.78 MiB | 3.36 MiB/s, done.
Resolving deltas: 100% (5947/5947), done.
(base)
~/Desktop/ST
▶ cd dirsearch
(base)
Desktop/ST/dirsearch  master ✓                                   16d16h
▶ python3 dirsearch.py -u https://facebook.com -e js,html

  dirsearch  v0.4.2

Extensions: js, html | HTTP method: GET | Threads: 30 | Wordlist size: 9523

Output File: /Users/akshatvg/Desktop/ST/dirsearch/reports/facebook.com/_21-12-03_01-27-15.txt

Error Log: /Users/akshatvg/Desktop/ST/dirsearch/logs/errors-21-12-03_01-27-15.log

Target: https://facebook.com/

[01:27:16] Starting:
[01:27:20] 301 -    0B  - /%2e%2e//google.com  ->  https://www.facebook.com/google.com
[01:27:20] 301 -    0B  - /.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd  ->  https://www.facebook.com/etc/passwd
[01:27:34] 200 -   87KB - /.well-known/apple-app-site-association
[01:27:35] 200 -   87KB - /.well-known/assetlinks.json
[01:28:02] 500 -    1KB - /assets/
[01:28:02] 500 -    1KB - /assets
[01:28:07] 301 -    0B  - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd  ->  https://www.facebook.com/etc/passwd
[01:28:17] 301 -    0B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/  ->  https://www.facebook.com/examples/jsp/%252e%252e/%252e%252e/manager/html/
[01:28:18] 200 -    1KB - /favicon.ico
[01:28:22] 404 -    9B  - /images/README
[01:28:22] 404 -    9B  - /images/Sym.php
[01:28:22] 404 -    9B  - /images/c99.php
[01:28:45] 200 -   90KB - /robots.txt
[01:28:50] 200 -   19B  - /status.php

Task Completed
```

Findomain

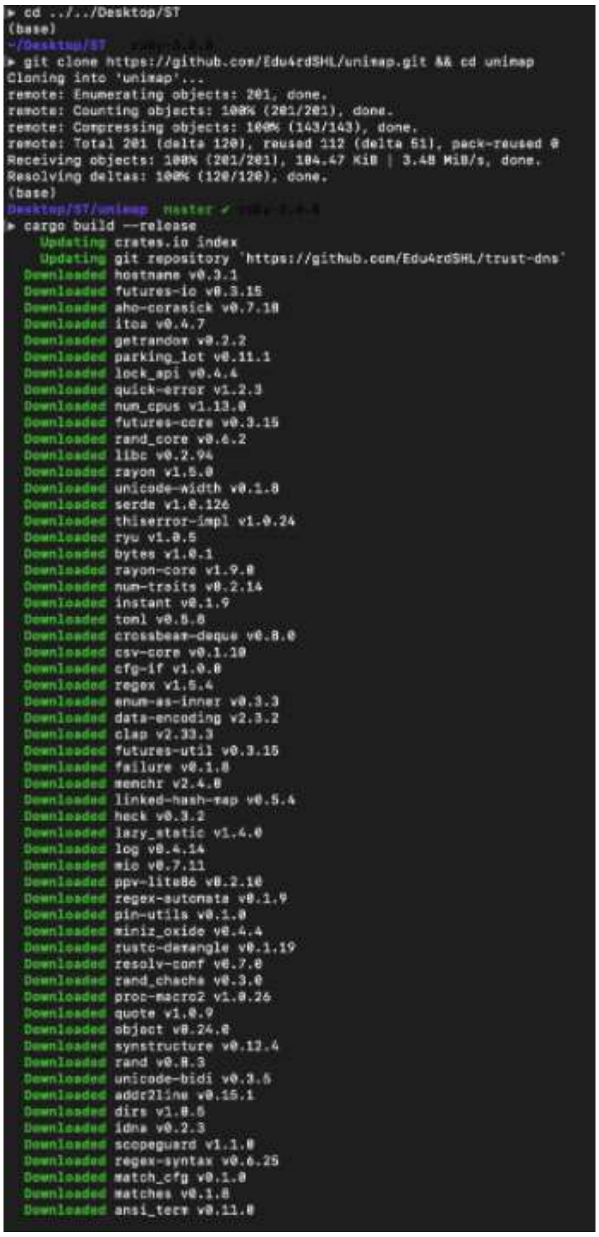
brew install findomain
findomain -qt facebook.com > findomain.txt



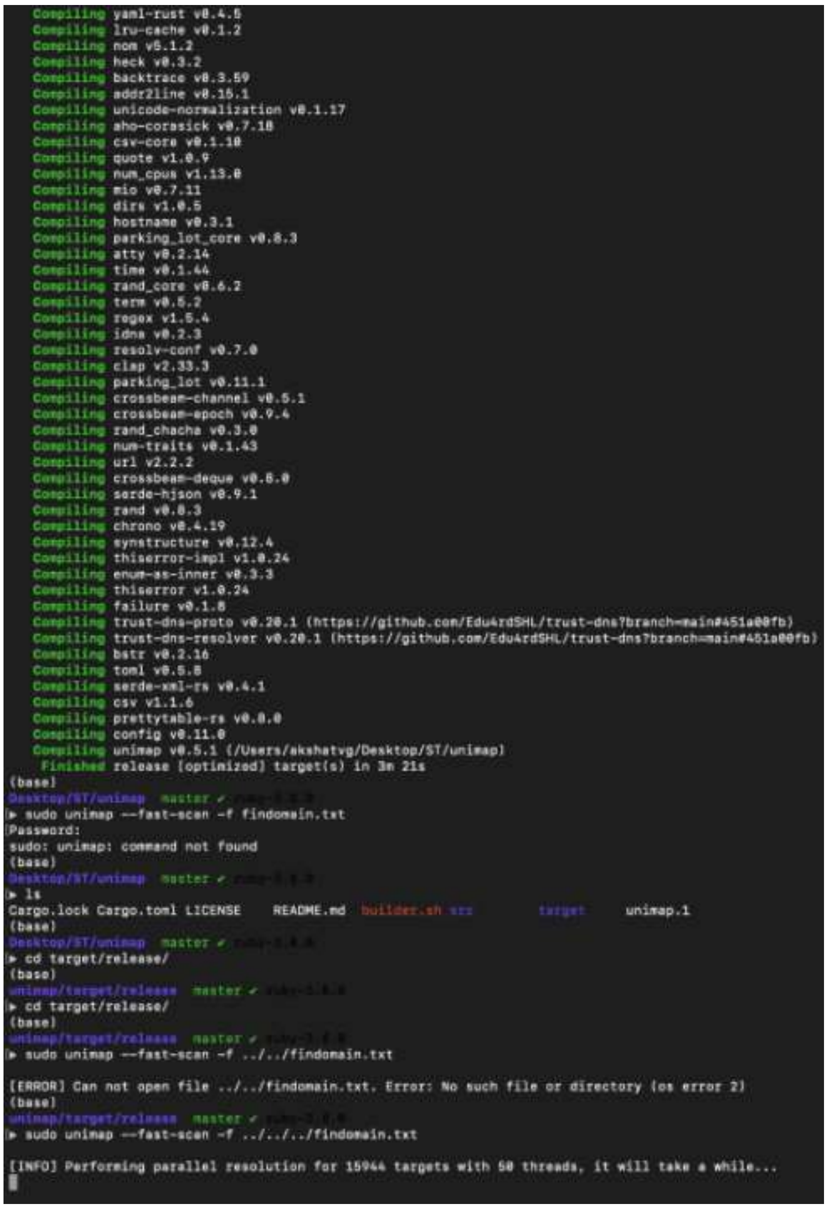
```
~/Desktop/ST
▶ brew install findomain
Running `brew update --preinstall`...
==> Auto-updated Homebrew!
Updated Homebrew from [illegible] to [illegible].
Updated 5 taps (homebrew/core, homebrew/cask, homebrew/services, mongodb/brew and dart-lang/dart).
==> New Formulae
[illegible]
==> Updated Formulae
Updated 178 formulae.
==> New Casks
[illegible]
==> Updated Casks
Updated 178 casks.

==> Homebrew was updated to version 3.3.6
The changelog can be found at:
  https://github.com/Homebrew/brew/releases/tag/3.3.6
==> Downloading https://ghcr.io/v2/homebrew/core/findomain/manifests/5.0.1
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/findomain/blobs/sha256:[illegible]
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:[illegible]
######################################################################## 100.0%
==> Pouring findomain--5.0.1.monterey.bottle.tar.gz
🍺  /usr/local/Cellar/findomain/5.0.1: 7 files, [illegible]
==> Running `brew cleanup findomain`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
(base)
~/Desktop/ST
▶ findomain -qt facebook.com > findomain.txt
(base)
~/Desktop/ST
▶
```

Unimap

git clone https://github.com/Edu4rdSHL/unimap && cd unimap
cargo build --release
cd target/release
sudo unimap --fast-scan -f ../../../findomain.txt
NOTE: It will take a lot of time for this because number of subdomains that we got are in lakhs.



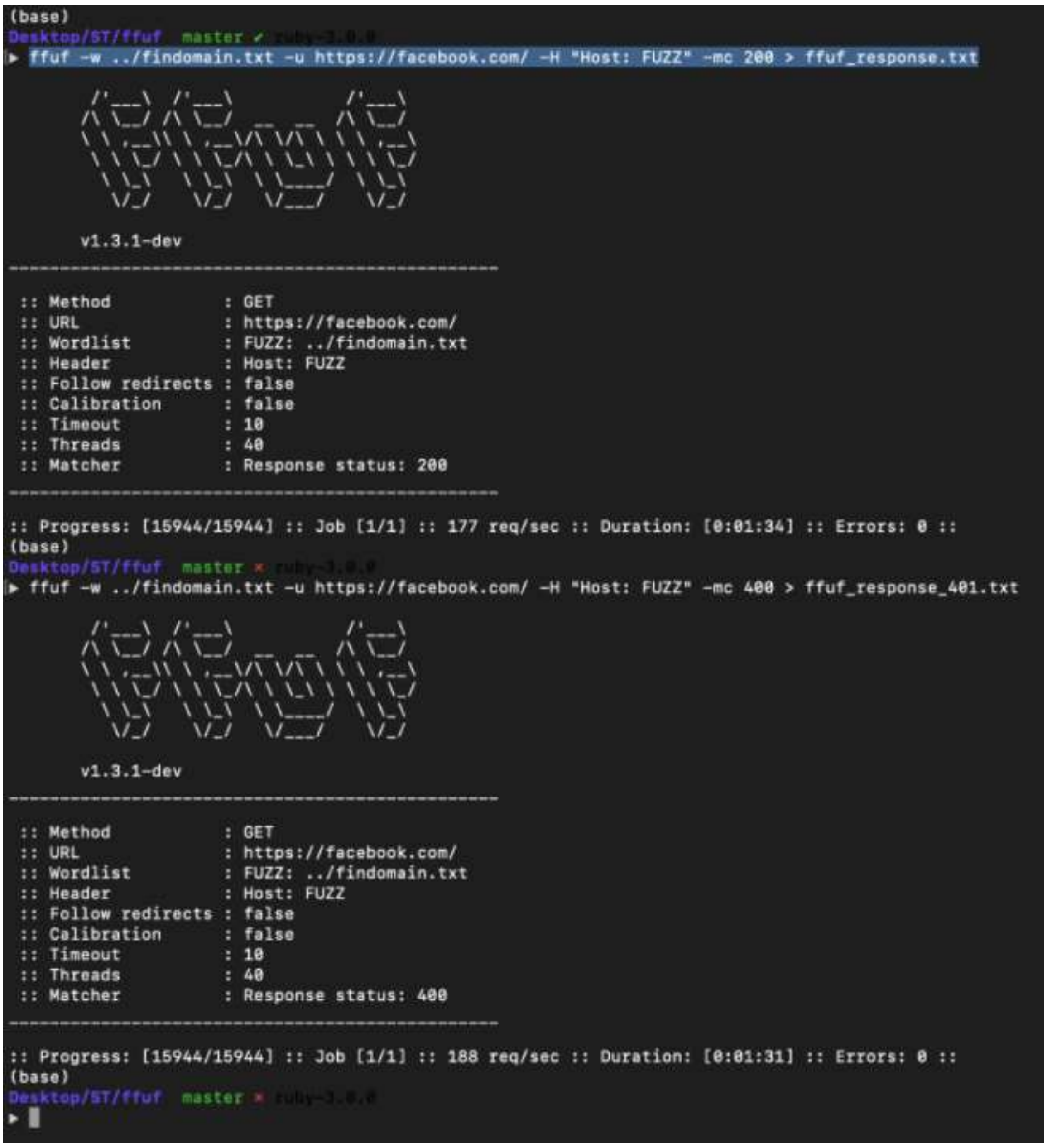
```
▶ cd ../../Desktop/ST
(base)
~/Desktop/ST  [illegible]
▶ git clone https://github.com/Edu4rdSHL/unimap.git && cd unimap
Cloning into 'unimap'...
remote: Enumerating objects: 201, done.
remote: Counting objects: 100% (201/201), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 201 (delta 120), reused 112 (delta 51), pack-reused 0
Receiving objects: 100% (201/201), 104.47 KiB | 3.48 MiB/s, done.
Resolving deltas: 100% (120/120), done.
(base)
Desktop/ST/unimap  master ✔ [illegible]
▶ cargo build --release
    Updating crates.io index
    Updating git repository `https://github.com/Edu4rdSHL/trust-dns`
  Downloaded hostname v0.3.1
  Downloaded futures-io v0.3.15
  Downloaded aho-corasick v0.7.18
  Downloaded itoa v0.4.7
  Downloaded getrandom v0.2.2
  Downloaded parking_lot v0.11.1
  Downloaded lock_api v0.4.4
  Downloaded quick-error v1.2.3
  Downloaded num_cpus v1.13.0
  Downloaded futures-core v0.3.15
  Downloaded rand_core v0.6.2
  Downloaded libc v0.2.94
  Downloaded rayon v1.5.0
  Downloaded unicode-width v0.1.8
  Downloaded serde v1.0.126
  Downloaded thiserror-impl v1.0.24
  Downloaded ryu v1.0.5
  Downloaded bytes v1.0.1
  Downloaded rayon-core v1.9.0
  Downloaded num-traits v0.2.14
  Downloaded instant v0.1.9
  Downloaded toml v0.5.8
  Downloaded crossbeam-deque v0.8.0
  Downloaded csv-core v0.1.10
  Downloaded cfg-if v1.0.0
  Downloaded regex v1.5.4
  Downloaded enum-as-inner v0.3.3
  Downloaded data-encoding v2.3.2
  Downloaded clap v2.33.3
  Downloaded futures-util v0.3.15
  Downloaded failure v0.1.8
  Downloaded memchr v2.4.0
  Downloaded linked-hash-map v0.5.4
  Downloaded heck v0.3.2
  Downloaded lazy_static v1.4.0
  Downloaded log v0.4.14
  Downloaded mio v0.7.11
  Downloaded ppv-lite86 v0.2.10
  Downloaded regex-automata v0.1.9
  Downloaded pin-utils v0.1.0
  Downloaded miniz_oxide v0.4.4
  Downloaded rustc-demangle v0.1.19
  Downloaded resolv-conf v0.7.0
  Downloaded rand_chacha v0.3.0
  Downloaded proc-macro2 v1.0.26
  Downloaded quote v1.0.9
  Downloaded object v0.24.0
  Downloaded synstructure v0.12.4
  Downloaded rand v0.8.3
  Downloaded unicode-bidi v0.3.5
  Downloaded addr2line v0.15.1
  Downloaded dirs v1.0.5
  Downloaded idna v0.2.3
  Downloaded scopeguard v1.1.0
  Downloaded regex-syntax v0.6.25
  Downloaded match_cfg v0.1.0
  Downloaded matches v0.1.8
  Downloaded ansi_term v0.11.0
```



```
   Compiling yaml-rust v0.4.5
   Compiling lru-cache v0.1.2
   Compiling nom v5.1.2
   Compiling heck v0.3.2
   Compiling backtrace v0.3.59
   Compiling addr2line v0.15.1
   Compiling unicode-normalization v0.1.17
   Compiling aho-corasick v0.7.18
   Compiling csv-core v0.1.10
   Compiling quote v1.0.9
   Compiling num_cpus v1.13.0
   Compiling mio v0.7.11
   Compiling dirs v1.0.5
   Compiling hostname v0.3.1
   Compiling parking_lot_core v0.8.3
   Compiling atty v0.2.14
   Compiling time v0.1.44
   Compiling rand_core v0.6.2
   Compiling term v0.5.2
   Compiling regex v1.5.4
   Compiling idna v0.2.3
   Compiling resolv-conf v0.7.0
   Compiling clap v2.33.3
   Compiling parking_lot v0.11.1
   Compiling crossbeam-channel v0.5.1
   Compiling crossbeam-epoch v0.9.4
   Compiling rand_chacha v0.3.0
   Compiling num-traits v0.1.43
   Compiling url v2.2.2
   Compiling crossbeam-deque v0.8.0
   Compiling serde-hjson v0.9.1
   Compiling rand v0.8.3
   Compiling chrono v0.4.19
   Compiling synstructure v0.12.4
   Compiling thiserror-impl v1.0.24
   Compiling enum-as-inner v0.3.3
   Compiling thiserror v1.0.24
   Compiling failure v0.1.8
   Compiling trust-dns-proto v0.20.1 (https://github.com/Edu4rdSHL/trust-dns?branch=main#451a00fb)
   Compiling trust-dns-resolver v0.20.1 (https://github.com/Edu4rdSHL/trust-dns?branch=main#451a00fb)
   Compiling bstr v0.2.16
   Compiling toml v0.5.8
   Compiling serde-xml-rs v0.4.1
   Compiling csv v1.1.6
   Compiling prettytable-rs v0.8.0
   Compiling config v0.11.0
   Compiling unimap v0.5.1 (/Users/akshatvg/Desktop/ST/unimap)
    Finished release [optimized] target(s) in 3m 21s
(base)
Desktop/ST/unimap  master ✓ [illegible]
▶ sudo unimap --fast-scan -f findomain.txt
Password:
sudo: unimap: command not found
(base)
Desktop/ST/unimap  master ✓ [illegible]
▶ ls
Cargo.lock Cargo.toml LICENSE    README.md  builder.sh src        target     unimap.1
(base)
Desktop/ST/unimap  master ✓ [illegible]
▶ cd target/release/
(base)
unimap/target/release  master ✓ [illegible]
▶ cd target/release/
(base)
unimap/target/release  master ✓ [illegible]
▶ sudo unimap --fast-scan -f ../../findomain.txt

[ERROR] Can not open file ../../findomain.txt. Error: No such file or directory (os error 2)
(base)
unimap/target/release  master ✓ [illegible]
▶ sudo unimap --fast-scan -f ../../../findomain.txt

[INFO] Performing parallel resolution for 15944 targets with 50 threads, it will take a while...
█
```

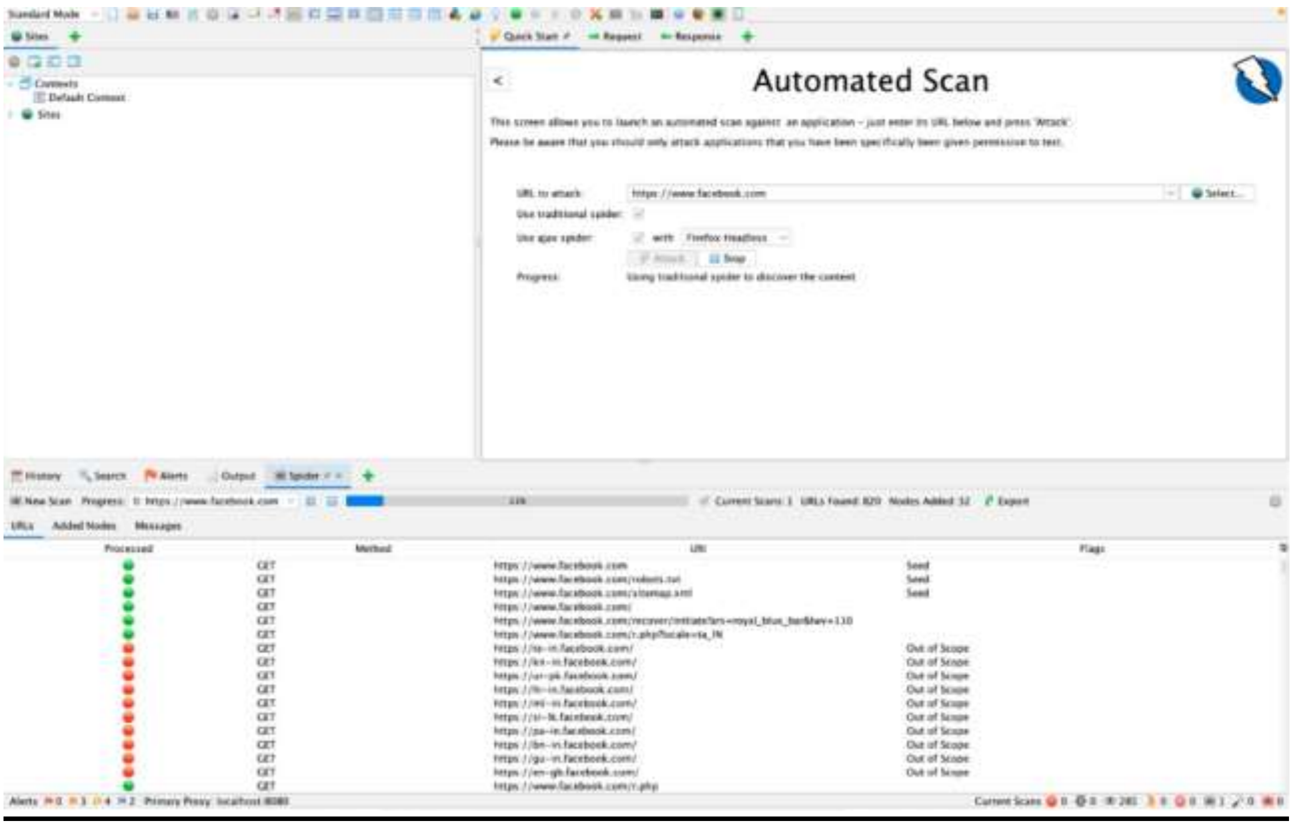
ffug- parameter fuzzing using brute force

git clone https://github.com/ffuf/ffuf && cd ffuf && go get && go build
ffuf -w ../findomain.txt -u https://facebook.com/ -H "Host: FUZZ" -mc 200 > ffuf_response.txt
ffuf -w ../findomain.txt -u https://facebook.com/ -H "Host: FUZZ" -mc 400 > ffuf_response_400.txt



```
(base)
Desktop/ST/ffuf  master ✔ ruby-3.0.0
▶ ffuf -w ../findomain.txt -u https://facebook.com/ -H "Host: FUZZ" -mc 200 > ffuf_response.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://facebook.com/
 :: Wordlist         : FUZZ: ../findomain.txt
 :: Header           : Host: FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200
________________________________________________

:: Progress: [15944/15944] :: Job [1/1] :: 177 req/sec :: Duration: [0:01:34] :: Errors: 0 ::
(base)
Desktop/ST/ffuf  master ✘ ruby-3.0.0
▶ ffuf -w ../findomain.txt -u https://facebook.com/ -H "Host: FUZZ" -mc 400 > ffuf_response_401.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://facebook.com/
 :: Wordlist         : FUZZ: ../findomain.txt
 :: Header           : Host: FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 400
________________________________________________

:: Progress: [15944/15944] :: Job [1/1] :: 188 req/sec :: Duration: [0:01:31] :: Errors: 0 ::
(base)
Desktop/ST/ffuf  master ✘ ruby-3.0.0
▶ █
```

<u>OWASP-ZAP</u>



Automated Scan

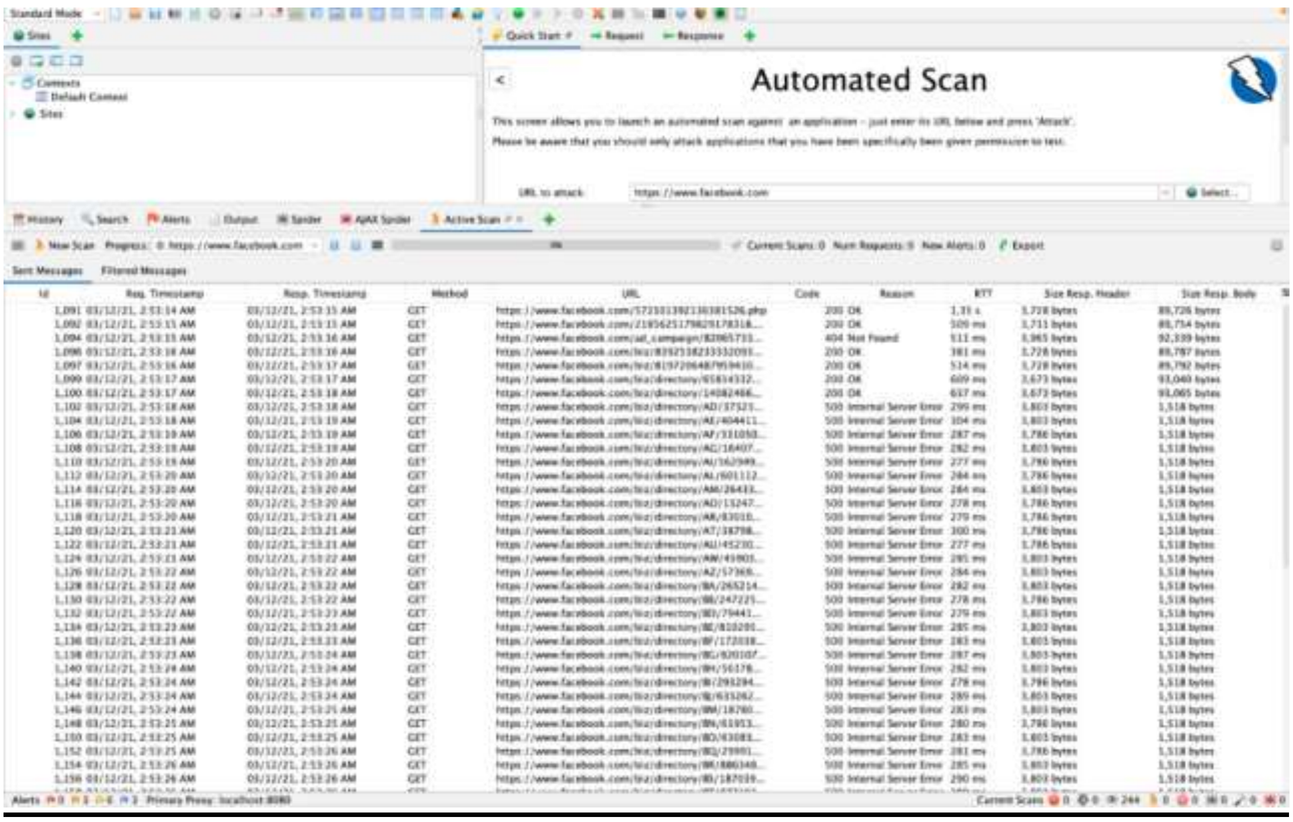
This screen allows you to launch an automated scan against an application – just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: https://www.facebook.com
Use traditional spider: ✓
Use ajax spider: ✓ with Firefox Headless
Attack | Stop
Progress: Using traditional spider to discover the content

Progress: 0: https://www.facebook.com 13% Current Scans: 1 URLs Found: 820 Nodes Added: 32

Processed	Method	URI	Flags
●	GET	https://www.facebook.com	Seed
●	GET	https://www.facebook.com/robots.txt	Seed
●	GET	https://www.facebook.com/sitemap.xml	Seed
●	GET	https://www.facebook.com/	
●	GET	https://www.facebook.com/recover/initiate?ars=royal_blue_bar&lwv=110	
●	GET	https://www.facebook.com/r.php?locale=ta_IN	
●	GET	https://ta-in.facebook.com/	Out of Scope
●	GET	https://kn-in.facebook.com/	Out of Scope
●	GET	https://ur-pk.facebook.com/	Out of Scope
●	GET	https://hi-in.facebook.com/	Out of Scope
●	GET	https://ml-in.facebook.com/	Out of Scope
●	GET	https://si-lk.facebook.com/	Out of Scope
●	GET	https://pa-in.facebook.com/	Out of Scope
●	GET	https://bn-in.facebook.com/	Out of Scope
●	GET	https://gu-in.facebook.com/	Out of Scope
●	GET	https://en-gb.facebook.com/	Out of Scope
●	GET	https://www.facebook.com/r.php	

Alerts 0 1 4 2 Primary Proxy: localhost:8080 Current Scans 0 0 281 0 0 1 0 0



Automated Scan

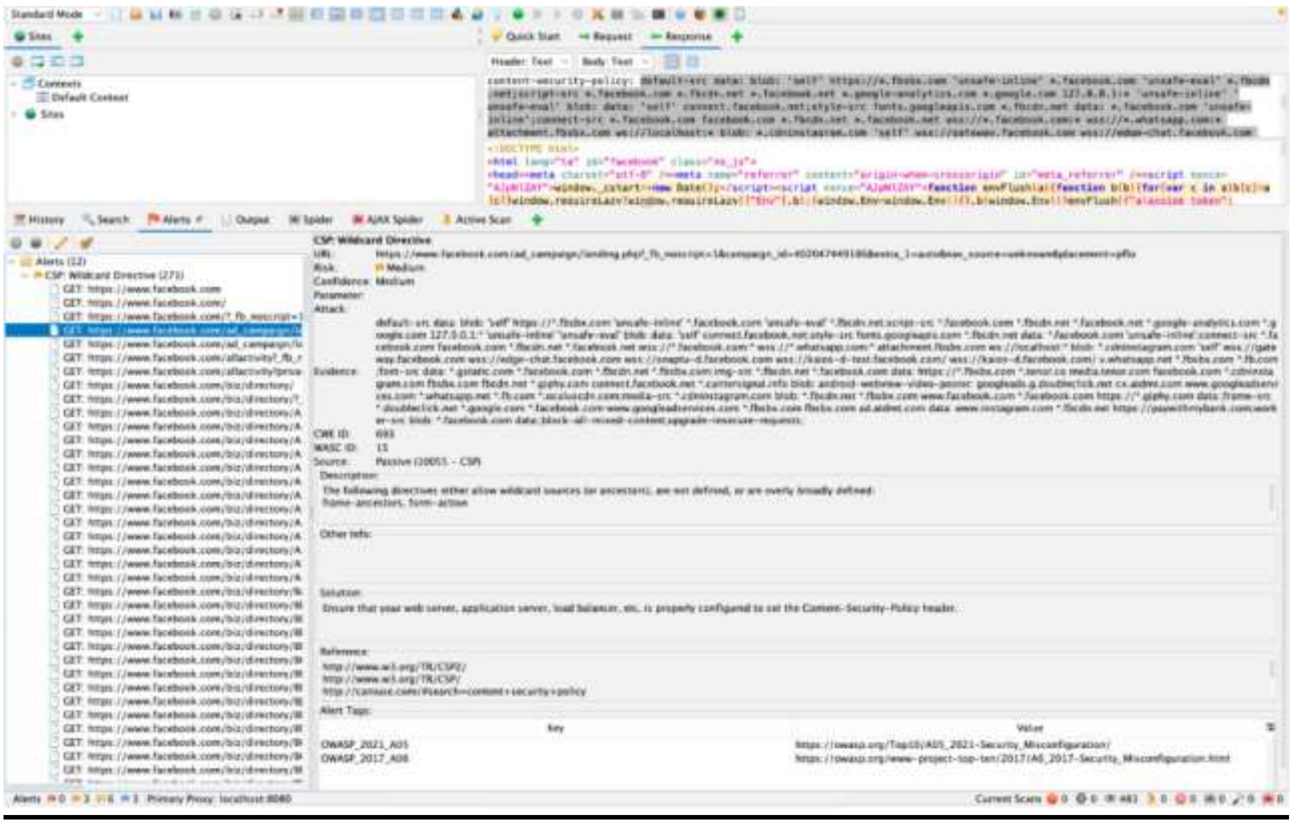
This screen allows you to launch an automated scan against an application – just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: https://www.facebook.com

Progress: 0: https://www.facebook.com 8% Current Scans: 0 Num Requests: 0 New Alerts: 0

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
1,091	03/12/21, 2:53:14 AM	03/12/21, 2:53:15 AM	GET	https://www.facebook.com/572101392136381526.php	200	OK	1.35 s	1,728 bytes	89,726 bytes
1,092	03/12/21, 2:53:15 AM	03/12/21, 2:53:15 AM	GET	https://www.facebook.com/2185625179829178318...	200	OK	509 ms	1,711 bytes	89,754 bytes
1,094	03/12/21, 2:53:15 AM	03/12/21, 2:53:16 AM	GET	https://www.facebook.com/ad_campaign/82065731...	404	Not Found	511 ms	3,965 bytes	92,339 bytes
1,096	03/12/21, 2:53:16 AM	03/12/21, 2:53:16 AM	GET	https://www.facebook.com/biz/8392518233332091...	200	OK	381 ms	1,728 bytes	89,787 bytes
1,097	03/12/21, 2:53:16 AM	03/12/21, 2:53:17 AM	GET	https://www.facebook.com/biz/8197206487959430...	200	OK	514 ms	1,728 bytes	89,792 bytes
1,099	03/12/21, 2:53:17 AM	03/12/21, 2:53:17 AM	GET	https://www.facebook.com/biz/directory/65814332...	200	OK	609 ms	3,673 bytes	93,040 bytes
1,100	03/12/21, 2:53:17 AM	03/12/21, 2:53:18 AM	GET	https://www.facebook.com/biz/directory/14082466...	200	OK	617 ms	3,673 bytes	93,065 bytes
1,102	03/12/21, 2:53:18 AM	03/12/21, 2:53:18 AM	GET	https://www.facebook.com/biz/directory/AD/37121...	500	Internal Server Error	299 ms	1,803 bytes	1,518 bytes
1,104	03/12/21, 2:53:18 AM	03/12/21, 2:53:19 AM	GET	https://www.facebook.com/biz/directory/AE/404411...	500	Internal Server Error	304 ms	1,803 bytes	1,518 bytes
1,106	03/12/21, 2:53:19 AM	03/12/21, 2:53:19 AM	GET	https://www.facebook.com/biz/directory/AF/331050...	500	Internal Server Error	287 ms	1,786 bytes	1,518 bytes
1,108	03/12/21, 2:53:19 AM	03/12/21, 2:53:19 AM	GET	https://www.facebook.com/biz/directory/AG/16407...	500	Internal Server Error	282 ms	1,803 bytes	1,518 bytes
1,110	03/12/21, 2:53:19 AM	03/12/21, 2:53:20 AM	GET	https://www.facebook.com/biz/directory/AI/162949...	500	Internal Server Error	277 ms	1,786 bytes	1,518 bytes
1,112	03/12/21, 2:53:20 AM	03/12/21, 2:53:20 AM	GET	https://www.facebook.com/biz/directory/AL/601112...	500	Internal Server Error	284 ms	1,786 bytes	1,518 bytes
1,114	03/12/21, 2:53:20 AM	03/12/21, 2:53:20 AM	GET	https://www.facebook.com/biz/directory/AM/26433...	500	Internal Server Error	284 ms	1,803 bytes	1,518 bytes
1,116	03/12/21, 2:53:20 AM	03/12/21, 2:53:20 AM	GET	https://www.facebook.com/biz/directory/AO/13247...	500	Internal Server Error	278 ms	1,786 bytes	1,518 bytes
1,118	03/12/21, 2:53:20 AM	03/12/21, 2:53:21 AM	GET	https://www.facebook.com/biz/directory/AR/83010...	500	Internal Server Error	279 ms	1,786 bytes	1,518 bytes
1,120	03/12/21, 2:53:21 AM	03/12/21, 2:53:21 AM	GET	https://www.facebook.com/biz/directory/AT/38798...	500	Internal Server Error	300 ms	1,786 bytes	1,518 bytes
1,122	03/12/21, 2:53:21 AM	03/12/21, 2:53:21 AM	GET	https://www.facebook.com/biz/directory/AU/45230...	500	Internal Server Error	277 ms	1,786 bytes	1,518 bytes
1,124	03/12/21, 2:53:21 AM	03/12/21, 2:53:22 AM	GET	https://www.facebook.com/biz/directory/AW/45903...	500	Internal Server Error	285 ms	1,803 bytes	1,518 bytes
1,126	03/12/21, 2:53:22 AM	03/12/21, 2:53:22 AM	GET	https://www.facebook.com/biz/directory/AZ/57369...	500	Internal Server Error	284 ms	1,803 bytes	1,518 bytes
1,128	03/12/21, 2:53:22 AM	03/12/21, 2:53:22 AM	GET	https://www.facebook.com/biz/directory/BA/265214...	500	Internal Server Error	282 ms	1,803 bytes	1,518 bytes
1,130	03/12/21, 2:53:22 AM	03/12/21, 2:53:22 AM	GET	https://www.facebook.com/biz/directory/BB/247225...	500	Internal Server Error	278 ms	1,786 bytes	1,518 bytes
1,132	03/12/21, 2:53:22 AM	03/12/21, 2:53:23 AM	GET	https://www.facebook.com/biz/directory/BD/79441...	500	Internal Server Error	279 ms	1,803 bytes	1,518 bytes
1,134	03/12/21, 2:53:23 AM	03/12/21, 2:53:23 AM	GET	https://www.facebook.com/biz/directory/BE/810201...	500	Internal Server Error	285 ms	1,803 bytes	1,518 bytes
1,136	03/12/21, 2:53:23 AM	03/12/21, 2:53:23 AM	GET	https://www.facebook.com/biz/directory/BF/172018...	500	Internal Server Error	283 ms	1,803 bytes	1,518 bytes
1,138	03/12/21, 2:53:23 AM	03/12/21, 2:53:24 AM	GET	https://www.facebook.com/biz/directory/BG/820107...	500	Internal Server Error	287 ms	1,803 bytes	1,518 bytes
1,140	03/12/21, 2:53:24 AM	03/12/21, 2:53:24 AM	GET	https://www.facebook.com/biz/directory/BH/56178...	500	Internal Server Error	282 ms	1,803 bytes	1,518 bytes
1,142	03/12/21, 2:53:24 AM	03/12/21, 2:53:24 AM	GET	https://www.facebook.com/biz/directory/BI/293284...	500	Internal Server Error	278 ms	1,786 bytes	1,518 bytes
1,144	03/12/21, 2:53:24 AM	03/12/21, 2:53:24 AM	GET	https://www.facebook.com/biz/directory/BJ/635262...	500	Internal Server Error	289 ms	1,803 bytes	1,518 bytes
1,146	03/12/21, 2:53:24 AM	03/12/21, 2:53:25 AM	GET	https://www.facebook.com/biz/directory/BM/18780...	500	Internal Server Error	283 ms	1,803 bytes	1,518 bytes
1,148	03/12/21, 2:53:25 AM	03/12/21, 2:53:25 AM	GET	https://www.facebook.com/biz/directory/BN/61953...	500	Internal Server Error	280 ms	1,786 bytes	1,518 bytes
1,150	03/12/21, 2:53:25 AM	03/12/21, 2:53:25 AM	GET	https://www.facebook.com/biz/directory/BO/43083...	500	Internal Server Error	283 ms	1,803 bytes	1,518 bytes
1,152	03/12/21, 2:53:25 AM	03/12/21, 2:53:26 AM	GET	https://www.facebook.com/biz/directory/BQ/29991...	500	Internal Server Error	281 ms	1,786 bytes	1,518 bytes
1,154	03/12/21, 2:53:26 AM	03/12/21, 2:53:26 AM	GET	https://www.facebook.com/biz/directory/BR/886348...	500	Internal Server Error	285 ms	1,803 bytes	1,518 bytes
1,156	03/12/21, 2:53:26 AM	03/12/21, 2:53:26 AM	GET	https://www.facebook.com/biz/directory/BS/187059...	500	Internal Server Error	290 ms	1,803 bytes	1,518 bytes

Alerts 0 1 6 2 Primary Proxy: localhost:8080 Current Scans 0 0 244 0 0 0 0 0



CSP: Wildcard Directive

URL: https://www.facebook.com/ad_campaign/landing.php?_fb_noscript=1&campaign_id=[illegible]&extra_1=auto&aw_source=unknown&placement=pflo

Risk: Medium

Confidence: Medium

Parameter:

Attack:

Evidence: default-src data: blob: 'self' https://*.fbsbx.com 'unsafe-inline' *.facebook.com 'unsafe-eval' *.fbcdn.net;script-src *.facebook.com *.fbcdn.net *.facebook.net *.google-analytics.com *.google.com 127.0.0.1:* 'unsafe-inline' 'unsafe-eval' blob: data: 'self' connect.facebook.net;style-src fonts.googleapis.com *.fbcdn.net data: *.facebook.com 'unsafe-inline';connect-src *.facebook.com facebook.com *.fbcdn.net *.facebook.net wss://*.facebook.com:* wss://*.whatsapp.com:* attachment.fbsbx.com ws://localhost:* blob: *.cdninstagram.com 'self' wss://gateway.facebook.com wss://edge-chat.facebook.com wss://snaptu-d.facebook.com wss://kaios-d-test.facebook.com/ wss://kaios-d.facebook.com/ v.whatsapp.net *.fbsbx.com *.fb.com;font-src data: *.gstatic.com *.facebook.com *.fbcdn.net *.fbsbx.com;img-src *.fbcdn.net *.facebook.com data: https://*.fbsbx.com *.tenor.co media.tenor.com facebook.com *.cdninstagram.com fbsbx.com fbcdn.net *.giphy.com connect.facebook.net *.carrierzigmal.info blob: android-webview-video-poster: googleads.g.doubleclick.net cx.atdmt.com www.googleadservices.com *.whatsapp.net *.fb.com *.oculuscdn.com;media-src *.cdninstagram.com blob: *.fbcdn.net *.fbsbx.com www.facebook.com *.facebook.com https://*.giphy.com data:;frame-src *.doubleclick.net *.google.com *.facebook.com www.googleadservices.com *.fbsbx.com fbsbx.com ad.atdmt.com data: www.instagram.com *.fbcdn.net https://paywithmybank.com;worker-src blob: *.facebook.com data:;block-all-mixed-content;upgrade-insecure-requests;

CWE ID: 693

WASC ID: 15

Source: Passive (10055 - CSP)

Description:

The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined:
frame-ancestors, form-action

Other Info:

Solution:

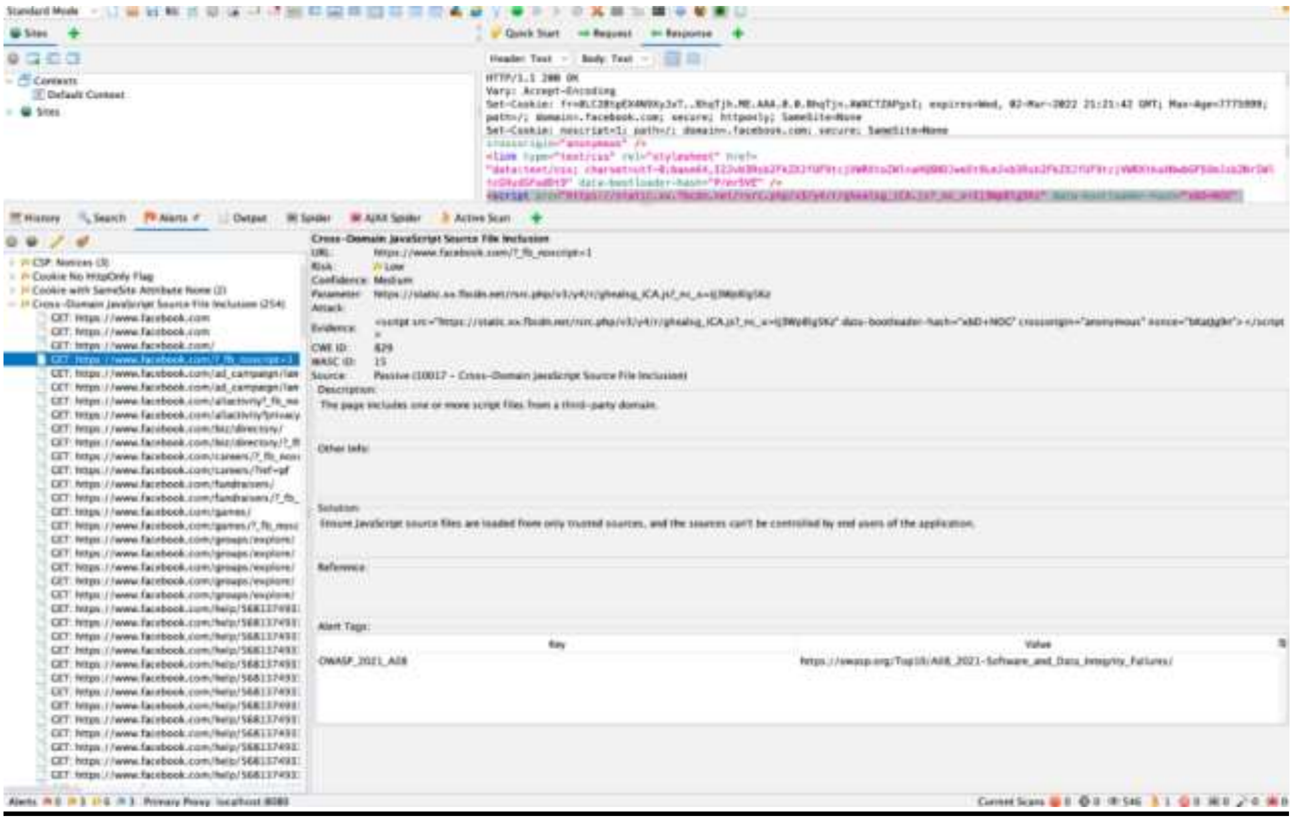
Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

Reference:

http://www.w3.org/TR/CSP2/
http://www.w3.org/TR/CSP/
http://caniuse.com/#search=content+security+policy

Alert Tags:

Key	Value
OWASP_2021_A05	https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
OWASP_2017_A06	https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html



Cross-Domain JavaScript Source File Inclusion

URL: https://www.facebook.com/?_fb_noscript=1

Risk: Low

Confidence: Medium

Parameter: https://static.xx.fbcdn.net/rsrc.php/v3/y4/r/ghealsg_ICA.js?_nc_x=IjI3WpRlgSKz

Attack:

Evidence: <script src="https://static.xx.fbcdn.net/rsrc.php/v3/y4/r/ghealsg_ICA.js?_nc_x=IjI3WpRlgSKz" data-bootloader-hash="xbD+NOC" crossorigin="anonymous" nonce="bKatJgkr"></script>

CWE ID: 829

WASC ID: 15

Source: Passive (10017 - Cross-Domain JavaScript Source File Inclusion)

Description:

The page includes one or more script files from a third-party domain.

Other Info:

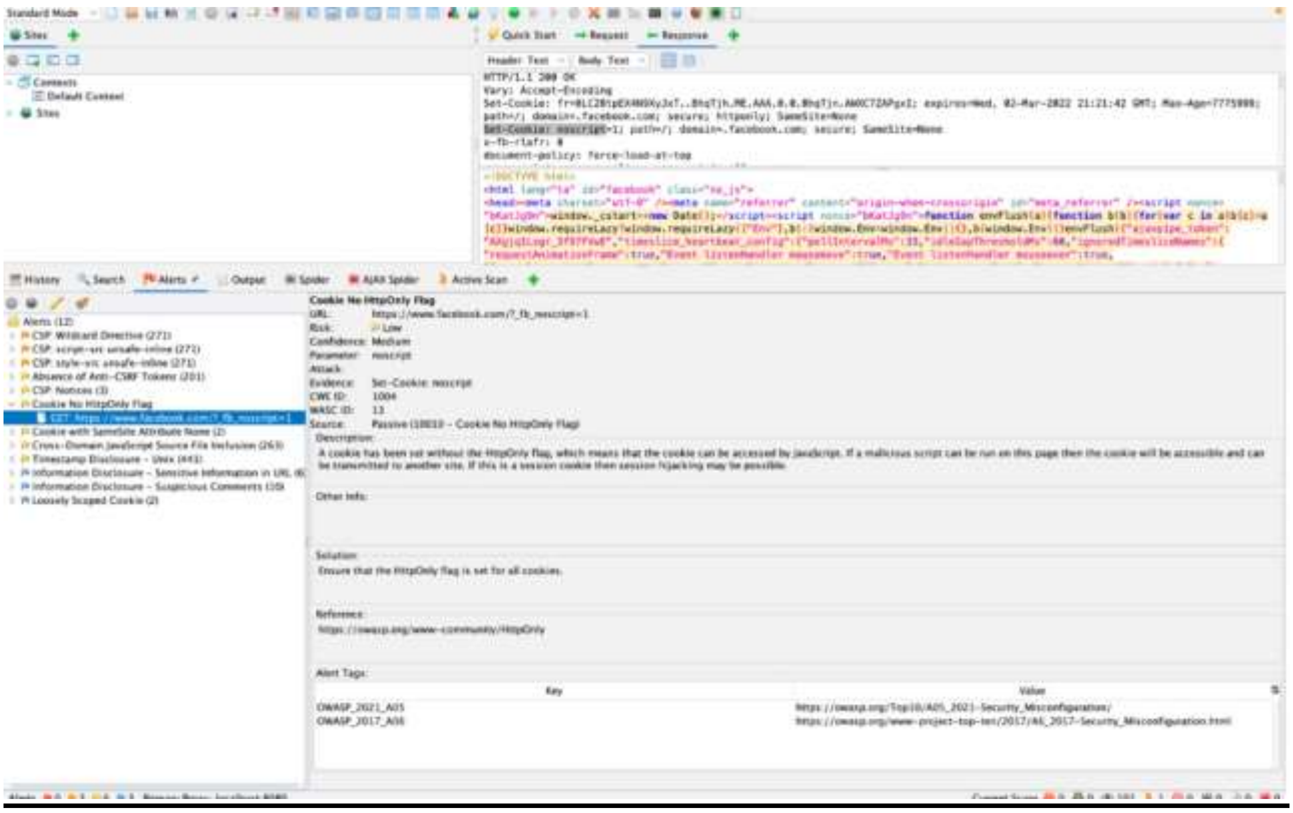
Solution:

Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Reference:

Alert Tags:

Key	Value
OWASP_2021_A08	https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/



```
HTTP/1.1 200 OK
Vary: Accept-Encoding
Set-Cookie: fr=0LC2BtpEX4N9Xy3xT..BhqTjh.ME.AAA.0.0.BhqTjn.AWXC7ZAPgxI; expires=Wed, 02-Mar-2022 21:21:42 GMT; Max-Age=7775999;
path=/; domain=.facebook.com; secure; httponly; SameSite=None
Set-Cookie: noscript=1; path=/; domain=.facebook.com; secure; SameSite=None
x-fb-rlafr: 0
document-policy: force-load-at-top
```

```
<!DOCTYPE html>
<html lang="la" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="referrer" content="origin-when-crossorigin" id="meta_referrer" /><script nonce=
"bKatJg0n">window._cstart=+new Date();</script><script nonce="bKatJg0n">function envFlush(a){function b(b){for(var c in a)b[c]=a
[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{},b(window.Env))}envFlush({"ajaxpipe_token":
"AXgjqILcgr_3F97FVw8","timeslice_heartbeat_config":{"pollIntervalMs":33,"idleGapThresholdMs":60,"ignoredTimesliceNames":{
"requestAnimationFrame":true,"Event listenHandler mousemove":true,"Event listenHandler mouseover":true,
```

Alerts (12)
- CSP: Wildcard Directive (271)
- CSP: script-src unsafe-inline (271)
- CSP: style-src unsafe-inline (271)
- Absence of Anti-CSRF Tokens (201)
- CSP: Notices (3)
- Cookie No HttpOnly Flag
  - GET: https://www.facebook.com/?_fb_noscript=1
- Cookie with SameSite Attribute None (2)
- Cross-Domain JavaScript Source File Inclusion (263)
- Timestamp Disclosure - Unix (443)
- Information Disclosure - Sensitive Information in URL (6)
- Information Disclosure - Suspicious Comments (10)
- Loosely Scoped Cookie (2)

Cookie No HttpOnly Flag

URL:	https://www.facebook.com/?_fb_noscript=1
Risk:	Low
Confidence:	Medium
Parameter:	noscript
Attack:	
Evidence:	Set-Cookie: noscript
CWE ID:	1004
WASC ID:	13
Source:	Passive (10010 - Cookie No HttpOnly Flag)

Description:
A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

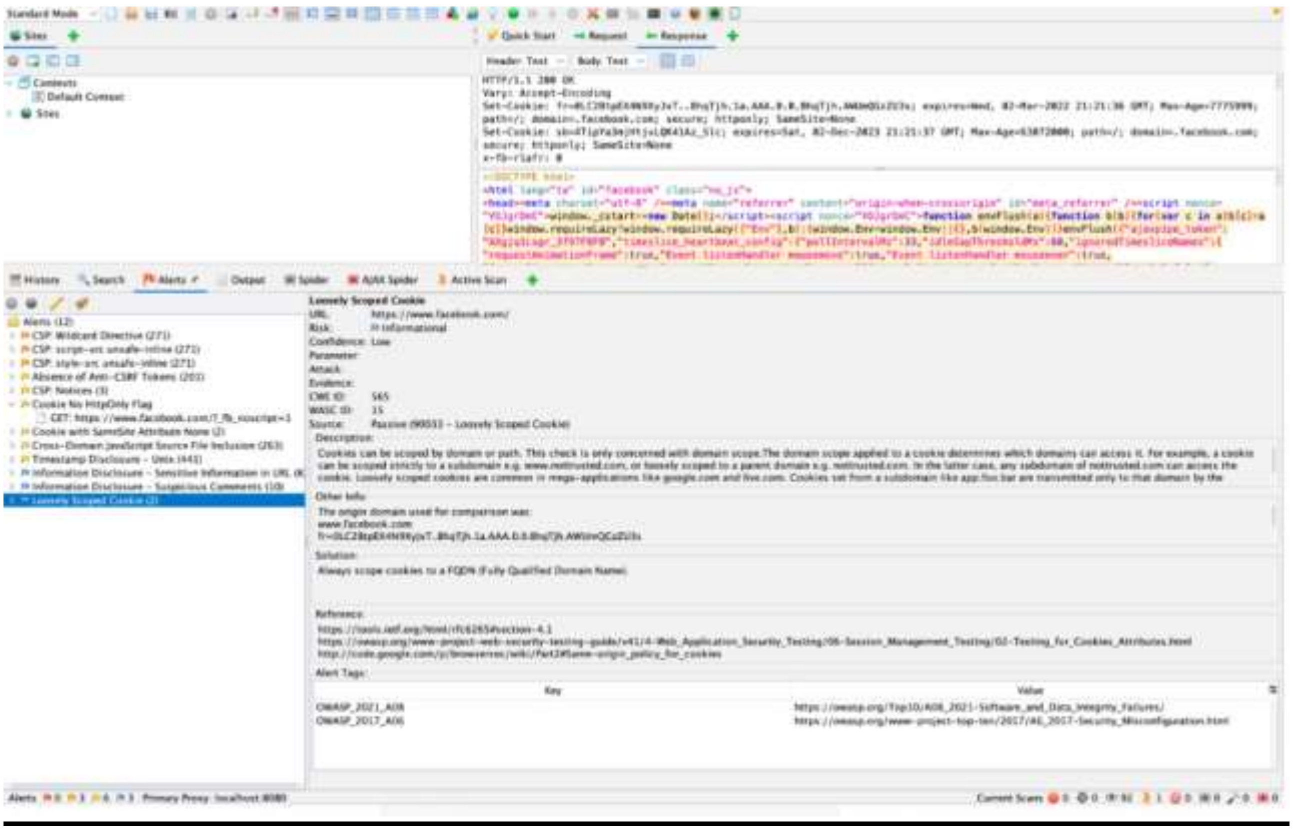
Other Info:

Solution:
Ensure that the HttpOnly flag is set for all cookies.

Reference:
https://owasp.org/www-community/HttpOnly

Alert Tags:

Key	Value
OWASP_2021_A05	https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
OWASP_2017_A06	https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html



```
HTTP/1.1 200 OK
Vary: Accept-Encoding
Set-Cookie: fr=0LC2BtpEX4N9XyJxT..BhqTjh.1a.AAA.0.0.BhqTjh.AWUmQCuZU3s; expires=Wed, 02-Mar-2022 21:21:36 GMT; Max-Age=7775999;
path=/; domain=.facebook.com; secure; httponly; SameSite=None
Set-Cookie: sb=4TipYa3mjHtjsLQK41Az_Sl(; expires=Sat, 02-Dec-2023 21:21:37 GMT; Max-Age=63072000; path=/; domain=.facebook.com;
secure; httponly; SameSite=None
x-fb-rlafr: 0
```

```
<!DOCTYPE html>
<html lang="la" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="referrer" content="origin-when-crossorigin" id="meta_referrer" /><script nonce=
"YGlgrbeC">window._cstart=+new Date();</script><script nonce="YGlgrbeC">function envFlush(a){function b(b){for(var c in a)b[c]=a
[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{},b(window.Env))}envFlush({"ajaxpipe_token":
"AXgjqILcgr_3F97F0P8","timeslice_heartbeat_config":{"pollIntervalMs":33,"idleGapThresholdMs":60,"ignoredTimesliceNames":{
"requestAnimationFrame":true,"Event listenHandler mousemove":true,"Event listenHandler mouseover":true,
```

Alerts (12)
- CSP: Wildcard Directive (271)
- CSP: script-src unsafe-inline (271)
- CSP: style-src unsafe-inline (271)
- Absence of Anti-CSRF Tokens (201)
- CSP: Notices (3)
- Cookie No HttpOnly Flag
  - GET: https://www.facebook.com/?_fb_noscript=1
- Cookie with SameSite Attribute None (2)
- Cross-Domain JavaScript Source File Inclusion (263)
- Timestamp Disclosure - Unix (443)
- Information Disclosure - Sensitive Information in URL (6)
- Information Disclosure - Suspicious Comments (10)
- Loosely Scoped Cookie (2)

Loosely Scoped Cookie

URL:	https://www.facebook.com/
Risk:	Informational
Confidence:	Low
Parameter:	
Attack:	
Evidence:	
CWE ID:	565
WASC ID:	15
Source:	Passive (90033 - Loosely Scoped Cookie)

Description:
Cookies can be scoped by domain or path. This check is only concerned with domain scope.The domain scope applied to a cookie determines which domains can access it. For example, a cookie can be scoped strictly to a subdomain e.g. www.nottrusted.com, or loosely scoped to a parent domain e.g. nottrusted.com. In the latter case, any subdomain of nottrusted.com can access the cookie. Loosely scoped cookies are common in mega-applications like google.com and live.com. Cookies set from a subdomain like app.foo.bar are transmitted only to that domain by the

Other Info:
The origin domain used for comparison was:
www.facebook.com
fr=0LC2BtpEX4N9XyJxT..BhqTjh.1a.AAA.0.0.BhqTjh.AWUmQCuZU3s

Solution:
Always scope cookies to a FQDN (Fully Qualified Domain Name).

Reference:
https://tools.ietf.org/html/rfc6265#section-4.1
https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
http://code.google.com/p/browsersec/wiki/Part2#Same-origin_policy_for_cookies

Alert Tags:

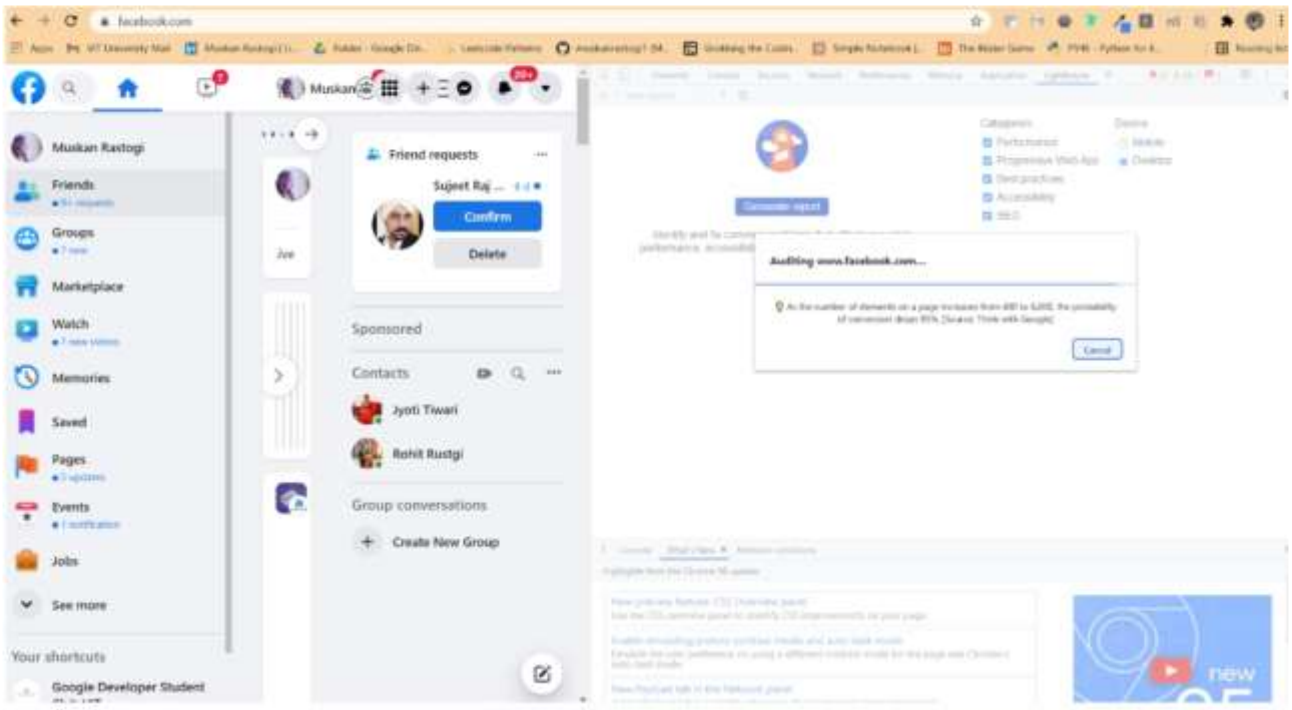
Key	Value
OWASP_2021_A08	https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/
OWASP_2017_A06	https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html

Lighthouse:

Lighthouse is an open-source, automated tool for improving the quality of web pages. You can run it against any web page, public or requiring authentication. It has audits for performance, accessibility, progressive web apps, SEO and more.

You can run Lighthouse in Chrome DevTools, from the command line, or as a Node module. You give Lighthouse a URL to audit, it runs a series of audits against the page, and then it generates a report on how well the page did. From there, use the failing audits as indicators on how to improve the page. Each audit has a reference doc explaining why the audit is important, as well as how to fix it.

Lighthouse in action:



To get a detailed format of the test results and audit check:
https://github.com/akshatvg/Software-Testing--Black-Box--Facebook.com/blob/master/Lighthouse.pdf

NESSUS

Nessus is a proprietary vulnerability scanner developed by Tenable, Inc. Tenable.io is a subscription-based service. Tenable also contains what was previously known as Nessus Cloud, which used to be Tenable's Software-as-a-Service solution. Nessus is an open-source network vulnerability scanner that uses the Common Vulnerabilities and Exposures architecture for easy cross-linking between compliant security tools. In fact, Nessus is one of the many vulnerability scanners used during vulnerability assessments and penetration testing engagements, including malicious attacks. Nessus is a tool that checks computers to find vulnerabilities that hackers COULD exploit.

Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack.

Nessus can scan these vulnerabilities and exposures:

Vulnerabilities that could allow unauthorized control or access to sensitive data on a system
Misconfiguration (e.g. open mail relay)
Denials of service (Dos) vulnerabilities
Default passwords, a few common passwords, and blank/absent passwords on some system accounts.
Software flaws, missing patches, malware and misconfiguration errors across a wide range of operating systems, devices and applications are dealt with by Nessus.

We performed an advanced scan and a web vulnerability scan, to check out the reports of them please visit:

https://github.com/akshatvg/Software-Testing--Black-Box--Facebook.com/blob/master/Nessus_Advanced-Scan.pdf

https://github.com/akshatvg/Software-Testing--Black-Box--Facebook.com/blob/master/Nessus_Advanced-Web-Scan.pdf

Future Scope

- Rate limiting: Vegeta to send 100 requests for login to test if the APIs rate limit a user.
- Stress testing: K6 to check with thousands of users. The servers should auto-scale.