Name: Jun Wang 000791814

Lab 2: Bypassing ClientSide Controls

Web Application Security

Marks: \_\_\_\_\_\_\_\_

Bonus: \_\_\_\_\_\_\_\_

Table of Contents

[Lab Outcome 2](#_Toc20167180)

[Background Reading 2](#_Toc20167181)

[Introduction 2](#_Toc20167182)

[Architecture Diagram 2](#_Toc20167183)

[Required Hardware/Software 3](#_Toc20167184)

[1.0 Hidden Fields 3](#_Toc20167185)

[2.0 Restrictions in HTML Fields 3](#_Toc20167186)

[3.0 Script-Based Validations 3](#_Toc20167187)

[4.0 Looking at HTML Code 3](#_Toc20167188)

[5.0 Collect user data from HTML forms 3](#_Toc20167189)

[6.0 Compare methods to handle client data securely 4](#_Toc20167190)

[7.0 Sign-Off – Lab 2: Bypassing Client-Side Controls 5](#_Toc20167191)

Web Application Security

Lab 2: Bypassing Client-Side Controls

# Lab Outcome

Find and exploit client-side controls.

Background Reading

Read the textbook sections listed in the Course Schedule.

# Introduction

Many web applications rely on validation on the client side to control the data that is submitted to the server. If these measures aren’t replicated on the server side, the server can easily be bypassed and a user can gain access to private or sensitive information on the server.

Architecture Diagram

|  |
| --- |
|  |

Required Hardware/Software

* VM Ubuntu 18.04 – 2 CPU, 4GB Ram, 20 GB hard disk
  + Docker
    - WebGoat v7.1
* VM Kali
  + Burp or other Web proxy (scanner)

# 1.0 Hidden Fields

In WebGoat, complete the **Parameter Tampering > Exploit Hidden Fields** lesson.

|  |
| --- |
|  |

# 2.0 Restrictions in HTML Fields

In WebGoat, complete the **Parameter Tampering > Bypass HTML Field Restrictions** lesson.

|  |
| --- |
| Insert evidence here. |

# 3.0 Script-Based Validations

In WebGoat, complete the **Parameter Tampering > Bypass Client Side JavaScript Validation** lesson.

|  |
| --- |
| STEP1.intercept the forms submission.  STEP2. keep pushing step button till this page shows up. Delete the function validate() and onclick=’validate();’    STEP3.Go back to browser, fill the form with random characters and submit |

# 4.0 Looking at HTML Code

In WebGoat, complete the **Code Quality > Discover Clues in the HTML** lesson.

|  |
| --- |
| Insert evidence here. |

# 5.0 Collect user data from HTML forms

Take clues from <https://sabe.io/classes/html/forms-inputs>, what is the name of the field “Enter your name:” and “Go” button

“Enter your name:”name=”person”

The name of “Go” button is “SUBMIT”.

|  |
| --- |
|  |

# 6.0 Compare methods to handle client data securely

Take clues from https://stormpath.com/blog/secure-your-rest-api-right-way, what is a common method used to secure website authentication

## 1.Traditional username/password authentication

## Use the Right API Security Protocol

3 most common used protocols are:

1)Basic API Authentication w/ TLS, it offers the lowest security options of the common protocols, a username and password is Base64 encoded.

2)OAuth1.0a, OAuth1 is a widely-used, tested, secure, signature-based protocol. This is the only of the three protocols that can be safely used without SSL.

3)OAuth2, OAuth2’s current specification removes signatures, OAuth2’s default settings are less secure than OAuth1. All the encryption is now handled by TLS.

### Custom Protocols

Custom API authentication protocols should be avoided, OAuth1.0a is a solid alternative.

2. Use API Keys

1)API keys/secrets are usually a long series of random characters that are difficult to guess.

2)API access would fail every time the password is changed.

3)Unique API keys authentication skips the hashing step and therefore speeds up your calls.

4)storing the API key/secret in a file only readable by the owner

5)“Url62” generates a globally unique byte array encoded using 62 ‘url- safe’ characters for each UUID.

6)In addition to avoiding the overhead of a session cluster (Database, Memcache, etc…), you can just add additional machines to your API cluster in order to grow with your user base.

# 7.0 Sign-Off – Lab 2: Bypassing Client-Side Controls

Detach this page and submit it to your instructor to show you have completed each of the sections.

Name:

Student ID:

|  |  |
| --- | --- |
| **Section** | **Instructor Initials** |
| 1.0 Hidden Fields |  |
| 2.0 Restrictions in HTML Fields |  |
| 3.0 Script-Based Validations |  |
| 4.0 Looking at HTML Code |  |
| 5.0 Collect user data from HTML forms |  |
| 6.0 Compare methods to handle client data securely |  |