Sécurité des Applications

Partie WEB

Stéphane Küng

November 17, 2020

OWASP Foundation Projects

OWASP Projects (OWASP)

Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software.

- OWASP Top 10 Proactive Security Controls For Software Developers to Build Secure Software
- OWASP Web Security Testing Guide
- OWASP Top 10

OWASP Top 10 Proactive Security Controls

The Top 10 Proactive Controls

The list is ordered by importance with list item number 1 being the most important:

C1: Define Security Requirements

C2: Leverage Security Frameworks and Libraries

C3: Secure Database Access

C4: Encode and Escape Data

C5: Validate All Inputs

C6: Implement Digital Identity

C7: Enforce Access Controls

C8: Protect Data Everywhere

C9: Implement Security Logging and Monitoring

C10: Handle All Errors and Exceptions

OWASP Web Security Testing Guide

4. Web Application Security Testing

OWASP Top 10

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Rappel

Escaping vs Encoding

Encoding is transforming data from one format into another format.

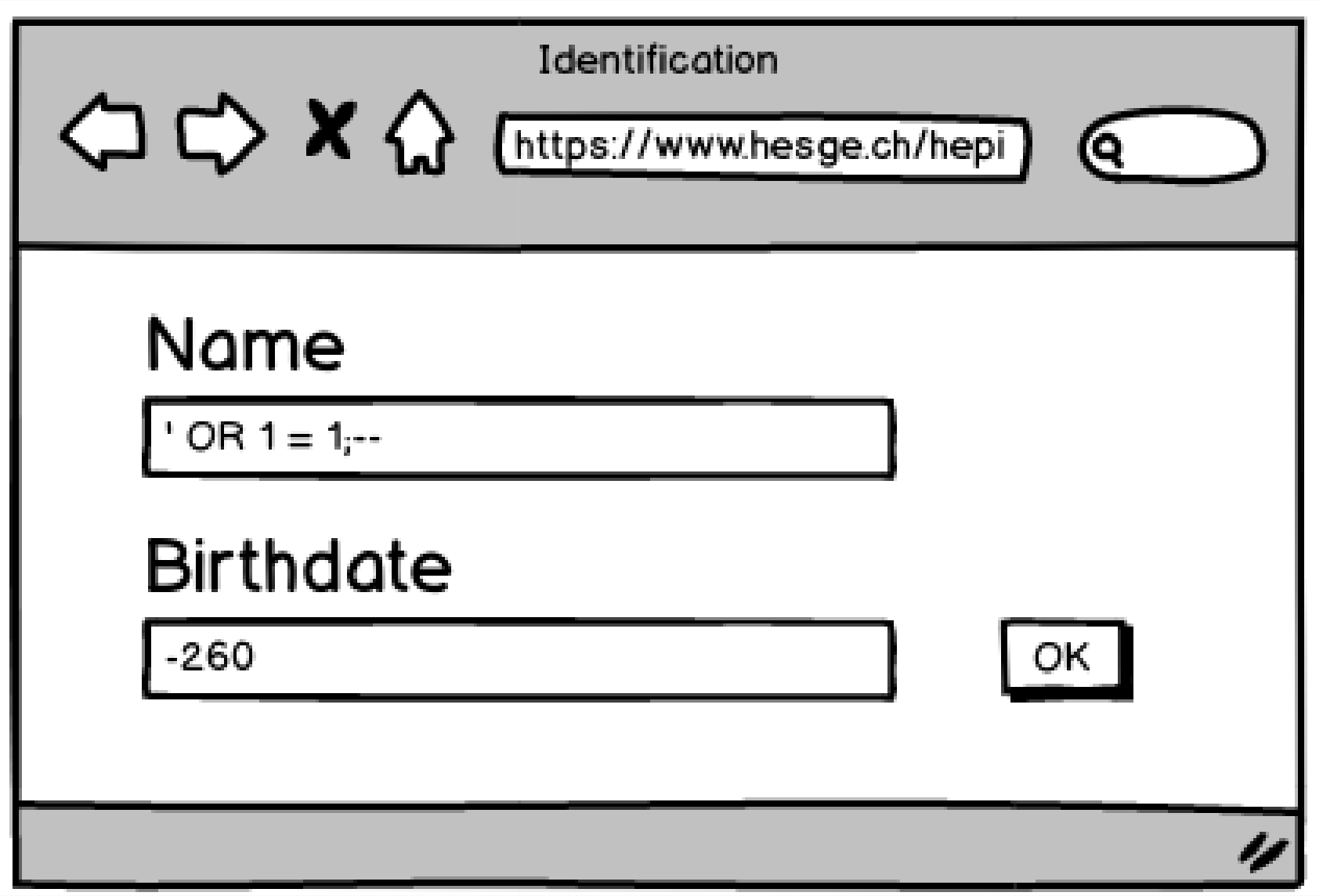
- **HTML**: `<h1>Hello</h1>`
- **UTF8, Unicode**
- **Base 64, URL Encoding, ...**

Escaping : Changing the interpreation of the following characterer by adding an escape character (Subset of encoding)

- **SQL**: `SELECT * FROM a WHERE name='Sam\'s`
- **Python/Javascript**: `"\"hello\"","""hello"""`,
- **Shell**: `cd My\ Folder`

Injection

Input Validation



Input Validation 2

Where the data comes from ?

- Form (maybe unintentional)
- Cookie
- URL
- Headers
- Another web site (ajax)
- ...

Input Validation 3

How to perform data validation ?[1]

- Data Type Validation (cast/parse)
- Range and constraint validation (min/max/neg)
- Code and Cross-reference validation (Rules, Object exist,)
- Structured validation (JSON/XML Schema)
- Whitelist : Array of allowd values (Days, Country)
- Regular Expression
- Blacklist : Need to be sure

[1] Input Validation Cheat Sheet, Data validation

Input Validation 4

Server Side

Input Validation 5

File Upload:

- Size
- Filename
- Extension
- Mime Type
- Magic number
- polymorph images

Input Validation 6

Misdone, **Input** Data Validation can lead to :

- Corruption

Output Data Validation 1

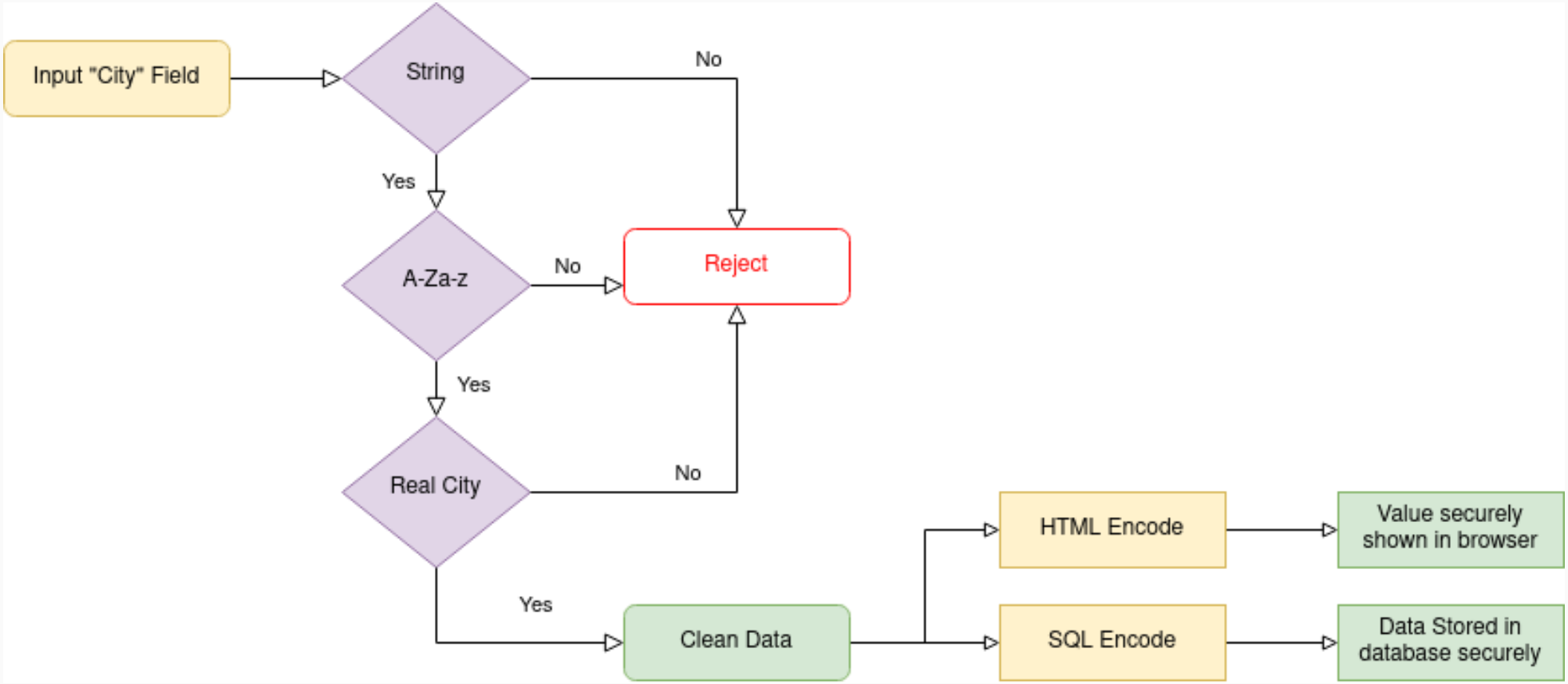
Disable any **active** content by using appropriate **encoding** before transmitting it to the targeted interpreter.

Output Data Validation 2

What is the destination / targeted system ?

- SQL interpreter : `'` `"` `;`
- HTML browser : `<` `>`
- URL : `:` `/` `@` `&` `=`
- XML file : `<` `>`
- Command Line : `;` `&`
- **Use the correct Encoding**
- **Use SQL Prepared Statements**

Output data Validation 3



Output data Validation 4

Misdone, **Output** Data Validation can lead to :

- Corruption
- Injection
- RCE
- Gain information
- Gain privileges
- XSS

Injection

https://owasp.org/www-project-top-ten/2017/A1_2017-Injection

SQL Injection 1

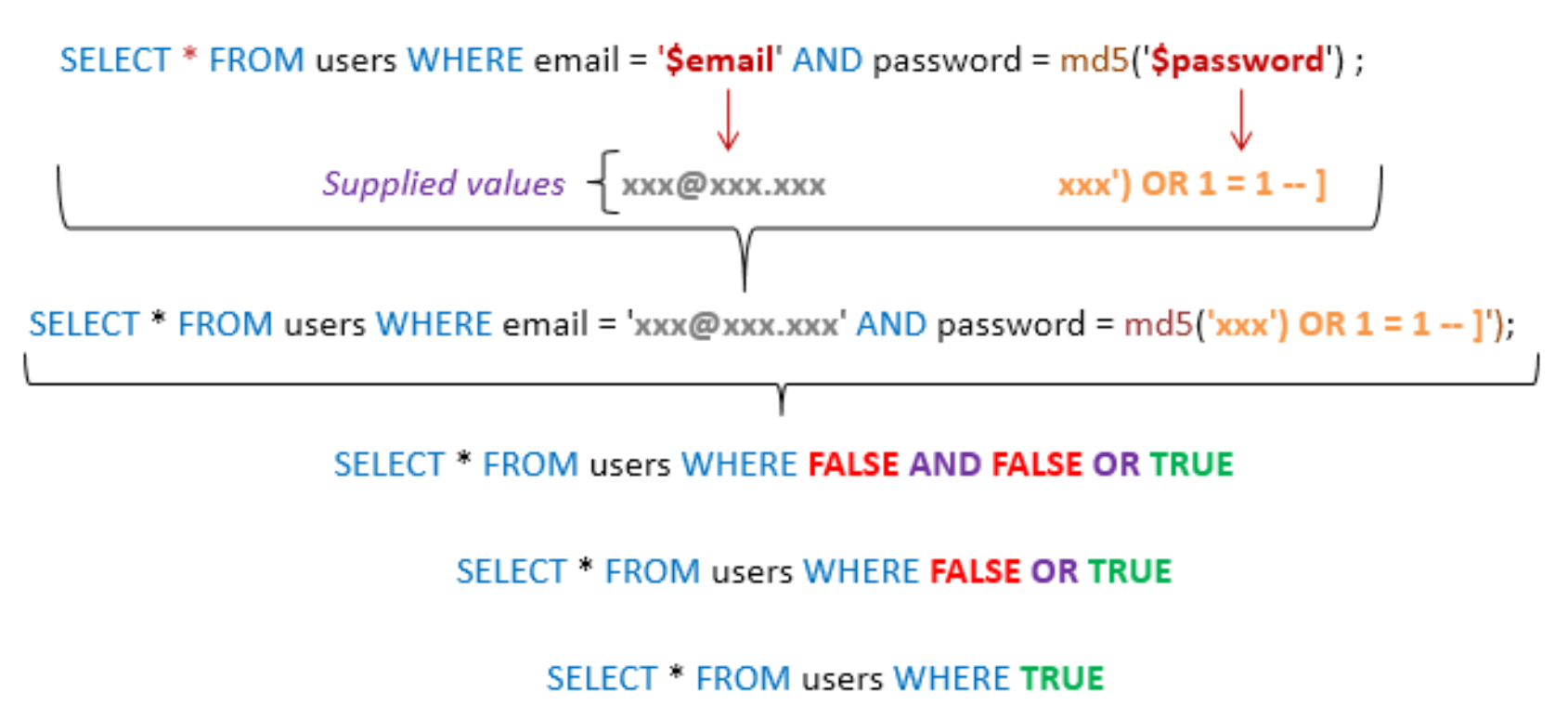


Figure 1: guru99.com

- Gain privileges, Corruption, ...

LDAP Injection

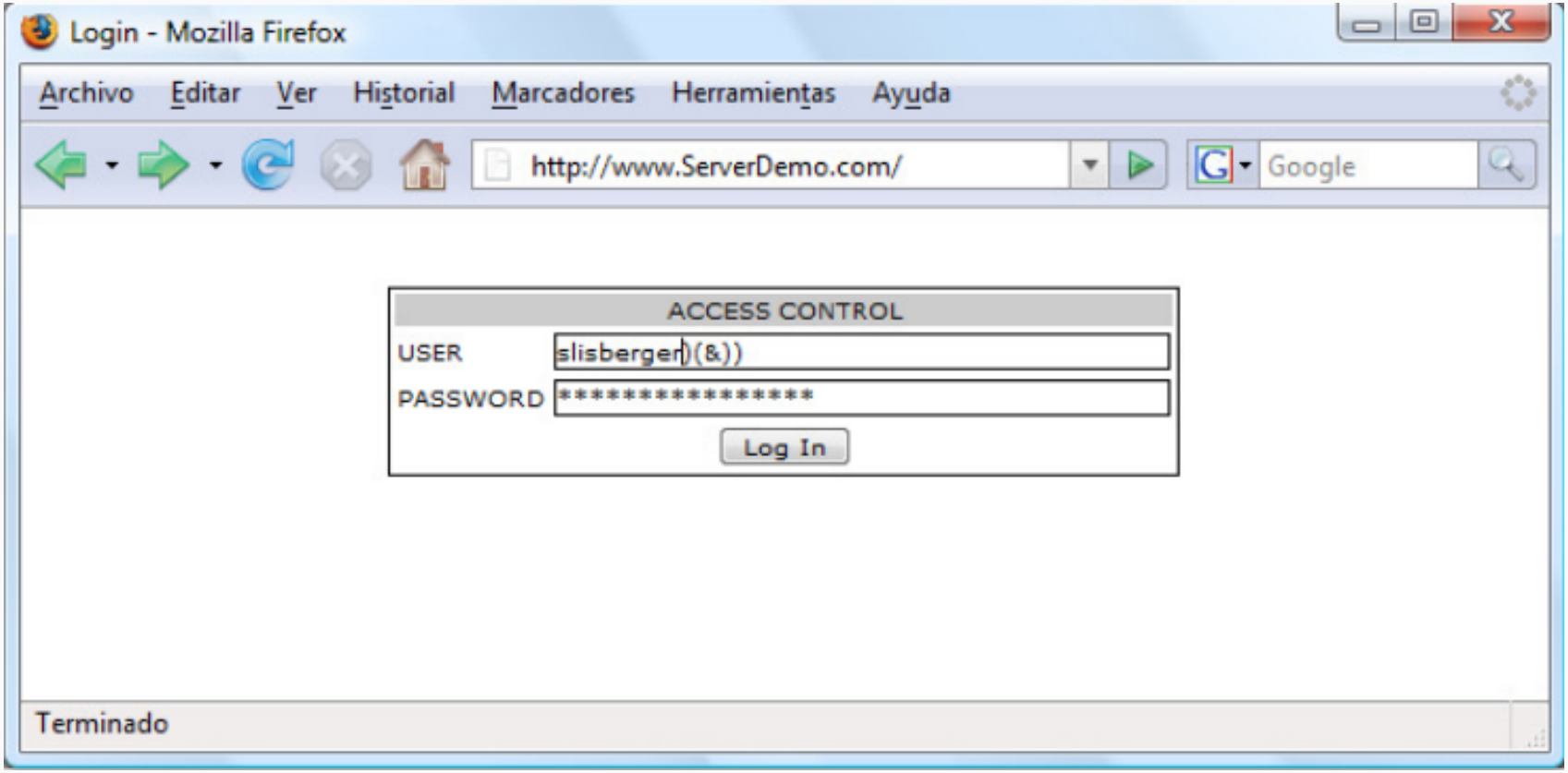


Figure 2: repo.zenk-security.com

```
(& (USER=slisberger)(&))(PASSWORD=Pwd))
```

- Gain privileges

CMD Injection

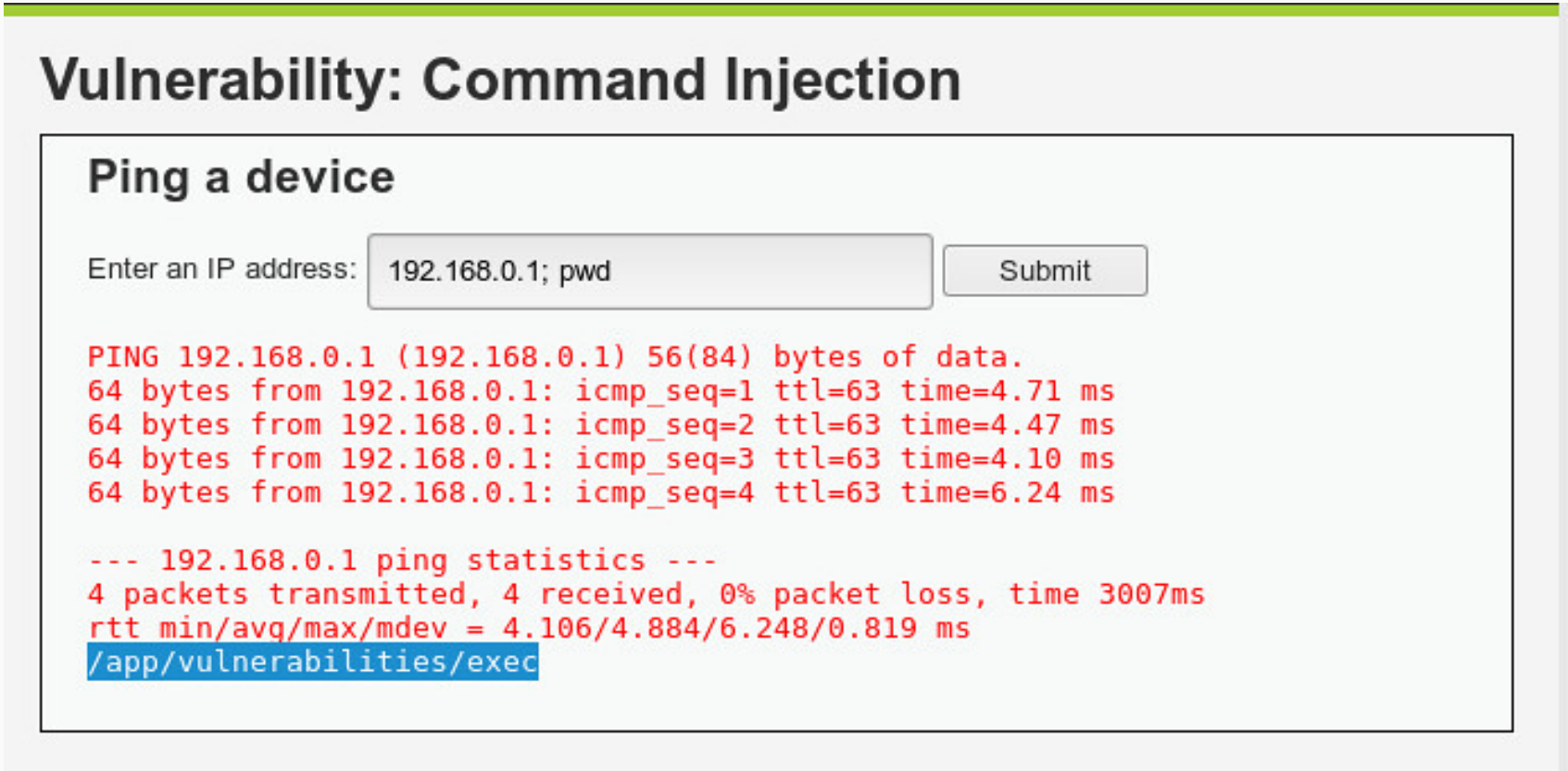


Figure 3: chris-young.net

- RCE, Gain information, Gain privileges