

UNICESUMAR
EVILLIN CRISTINE TRAPP ROSA

TRABALHO PRÁTICO
MONTAGEM DE UM AMBIENTE VIRTUAL WEB VULNERÁVEL

CURITIBA

2023

UNICESUMAR
EVILLIN CRISTINE TRAPP ROSA

TRABALHO PRÁTICO
MONTAGEM DE UM AMBIENTE VIRTUAL WEB VULNERÁVEL

Trabalho apresentado à disciplina de Desafio profissional apresentada a disciplina de Desafio profissional III por solicitação da professora Ana Paula Costacurta.

CURITIBA

2023

Sumário

INTRODUÇÃO	4
1. Ambiente virtual	5
1.1. Instalação e configuração do VirtualBox	5
1.2. Instalação e configuração do Linux na máquina virtual	5
1.3. Instalação e configuração do WebGoat	5
2. Visão Geral WebGoat	5
2.1. Descrição e funcionalidades	5
2.2. Como acessar e navegar	5
3. Práticas comuns de segurança em aplicações Web	6
3.1. Conceitos básicos de segurança em aplicações web	6
3.2. Identificação de vulnerabilidades comuns em aplicações web	6
3.3. Boas práticas para mitigação de vulnerabilidades em aplicações web	7
3.4. SQL Injection	7
4. Próximos passos	7
5. Lições de introdução e telas	9
6. Relatório	11
CONCLUSÃO	14
REFERÊNCIAS	15

CURITIBA

2023

INTRODUÇÃO

Vulnerabilidade em sistemas é um tema muito presente no dia a dia do estudante a atuante da área de tecnologia, tendo em vista que diversas empresas possuem aplicações web diversificada e sólida. As seguranças nessas aplicações são essenciais e muito importantes por inúmeros sentidos, mas principalmente por conta da coleta de dados sensíveis. Esses dados são extremamente vulneráveis à ataques que causam danos para a empresa, mas principalmente para seus clientes que tem sua privacidade invadida e muitas das vezes não saem ilesos por conta da falta de segurança.

Por conta de sistemas Web serem acessados por via da internet, esse trabalho tem como objetivo a exploração de vulnerabilidades de aplicações Web, através do WebGoat executado em uma máquina virtual com sistema operacional Linux Kali.

Para isso, fiz a instalação da virtualBox 6.1 em minha máquina, e utilizei o sistema operacional indicado nas instruções passadas e instalado o Java Runtime Environment (JRE) no Linux. Para estudo e realização das próximas etapas, foi escolhido o (A2) Cryptographic Failures no WebGoat dentro da máquina virtual.

CURITIBA

2023

1. Ambiente virtual

1.1. Instalação e configuração do VirtualBox

A instalação e configuração foram feitas sem muitas dificuldades, apenas a versão que estava disponível no material de apoio que não rodou no meu notebook, assim, tive que instalar uma versão mais antiga, mas sem problemas em questão de configurações.

1.2. Instalação e configuração do Linux na máquina virtual

Durante a instalação, o link disponível no arquivo não levou a versão 2,7gb, e não tive sucesso na procura do link correto, fazendo assim com que eu entrasse em contato com meus colegas de sala para ter acesso ao link correta, mas instalação e configuração do Linux foram feitas sem dificuldades, apenas tive o esforço de identificar qual era o login e senha da aplicação.

1.3. Instalação e configuração do WebGoat

Instalação do WebGoat foi feita sem dificuldades, após acessar o site, realizei meu cadastro e tive acesso às vulnerabilidades.

2. Visão Geral WebGoat

2.1. Descrição e funcionalidades

O WebGoat é uma aplicação web de código aberto desenvolvida com o objetivo de proporcionar um ambiente seguro para o aprendizado e treinamento em segurança de aplicações web. Ele simula uma aplicação real com diversas vulnerabilidades intencionais incorporadas, permitindo aos usuários explorarem essas vulnerabilidades de maneira controlada.

Suas funcionalidades principais são: Simulação de Vulnerabilidades, Ambiente Interativo de Aprendizado, Tutoriais e Lições, Aprendizado através de Erros e entre diversas outras.

2.2. Como acessar e navegar

Para instalação, devemos baixar o WebGoat no site oficial ou repositório do GitHub e seguir passo a passo de instalações específicas presente em sua documentação.

Para inicialização, após a instalação, inicie o servidor WebGoat de acordo com as instruções do sistema operacional. Normalmente, isso é feito através de um comando específico e aguardar até que o servidor WebGoat seja inicializado e exiba uma mensagem informando que está pronto para receber conexões.

Para acesso, devemos abrir o link <http://localhost:8080/WebGoat/> no navegador web da VirtualBox.

Para navegação, ao entrar no WebGoat, você será recebido com a página inicial do aplicativo, explore as diferentes lições e tópicos disponíveis. Cada lição aborda uma vulnerabilidade específica e fornece informações detalhadas sobre ela. Selecione uma lição para começar. Leia as instruções fornecidas e siga as orientações passo a passo para entender e explorar a vulnerabilidade selecionada.

3. Práticas comuns de segurança em aplicações Web

3.1. Conceitos básicos de segurança em aplicações web

Os conceitos básicos de segurança em aplicações web são fundamentais para proteger os sistemas contra ameaças e ataques. Esses conceitos incluem autenticação, autorização, criptografia, gerenciamento de sessões, validação de entrada, proteção contra ataques conhecidos e atualizações de segurança. É importante implementar medidas de segurança em várias camadas, manter os sistemas atualizados e realizar auditorias regulares para garantir a proteção contínua da aplicação contra ameaças.

3.2. Identificação de vulnerabilidades comuns em aplicações web

A identificação de vulnerabilidades comuns em aplicações web é essencial para garantir a segurança dos sistemas. Algumas vulnerabilidades comuns incluem injeção de SQL, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), autenticação e gerenciamento de sessões deficientes, exposição de informações sensíveis, configuração incorreta do servidor, controle de acesso inadequado, validação de entrada insuficiente, exposição de diretórios e falhas de controle de erros. É importante realizar testes de segurança, revisões de código e implementar práticas de desenvolvimento seguro para identificar e mitigar essas vulnerabilidades.

3.3. Boas práticas para mitigação de vulnerabilidades em aplicações web

Para mitigar vulnerabilidades em aplicações web, é importante seguir boas práticas de segurança. Algumas dessas práticas incluem manter o sistema atualizado, aplicar o princípio do menor privilégio, utilizar autenticação e autorização seguras, validar rigorosamente a entrada de dados, criptografar informações sensíveis, proteger contra ataques de força bruta, realizar testes de segurança regulares, implementar proteção contra CSRF, gerenciar adequadamente erros, e promover a educação e conscientização sobre segurança. Seguir essas práticas ajuda a fortalecer a segurança das aplicações web e reduzir o risco de exploração de vulnerabilidades.

3.4. SQL Injection

A injeção de SQL (SQL Injection) é uma vulnerabilidade comum encontrada em aplicações web. Ela permite que um invasor insira comandos SQL maliciosos em uma consulta SQL, explorando a falta de validação ou sanitização adequada dos dados fornecidos pelos usuários. Isso ocorre quando os dados do usuário são diretamente concatenados em uma consulta SQL, sem passar por etapas de verificação.

Essa vulnerabilidade pode levar a diversas ações maliciosas, como manipulação ou exclusão de dados do banco de dados, acesso não autorizado a informações sensíveis e até mesmo o controle total do sistema. Para evitar ataques de injeção de SQL, é necessário adotar boas práticas de programação, como o uso de consultas parametrizadas ou prepared statements, que separam os comandos SQL dos dados do usuário.

Além disso, é importante realizar a validação e sanitização adequada dos dados de entrada, limitar os privilégios de acesso ao banco de dados e manter todas as camadas do sistema atualizadas para evitar vulnerabilidades conhecidas. Ao implementar essas medidas, é possível mitigar significativamente o risco de ataques de injeção de SQL em aplicações web.

4. Próximos passos

A lição (A2) Cryptographic Failures explica diferentes tipos de técnicas de criptografia comumente usadas em aplicativos da web.

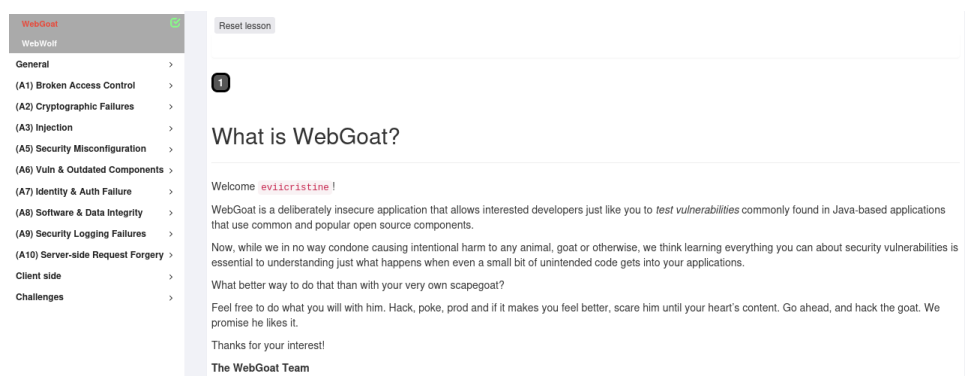
Tendo como objetivo familiarizar-se com as seguintes técnicas, sendo elas as próximas etapas:

- Encoding: Não é realmente criptografia, mas é muito usada em todos os tipos de padrões em torno de funções criptográficas. Especialmente a codificação Base64 (Base64 é uma técnica usada para transformar todos os tipos de bytes em um intervalo específico de bytes).
- Hashing: É um tipo de criptografia usado principalmente para detectar se os dados originais foram alterados. Um hash é gerado a partir dos dados originais. Baseia-se em técnicas criptográficas irreversíveis.
- Encryption: É baseada em um segredo compartilhado que é usado tanto para criptografia quanto para descryptografia. Portanto, ambas as partes (que estão envolvidas na troca de segredos) compartilham a mesma chave.
- Signing: Usada quando a integridade é importante. É uma garantia de que os dados enviados da Parte-A para a Parte-B não foram alterados. Portanto, a Parte-A assina os dados calculando o hash dos dados e criptografando esse hash usando uma chave privada assimétrica. A Parte-B pode verificar os dados calculando o hash dos dados e descryptografando a assinatura para comparar se os dois hashes são iguais.
- Keystores: É um lugar onde você armazena as chaves. Além do keystore, o termo truststore também é usado com frequência. Um truststore é o mesmo que um keystore. Só que geralmente contém apenas os certificados (basicamente apenas chaves públicas e informações do emissor) de certificados confiáveis ou autoridades de certificação.
- Security defaults: Um grande problema em todos os tipos de sistemas é o uso de configurações padrão. Por exemplo. nome de usuário/senhas padrão em roteadores, senhas padrão para keystores, modo não criptografado padrão, etc... (deve-se proteger sua chave privada id_rsa e nome de usuário/senha SSH para seu servidor).
- Post quantum crypto: Os computadores quânticos estão aqui e obtendo mais potência em qubits disponíveis a cada ano. Os computadores quânticos são e serão capazes de descryptografar informações criptografadas com algoritmos considerados seguros. Há alguns anos, muita comunicação criptografada usando criptografia quântica vulnerável

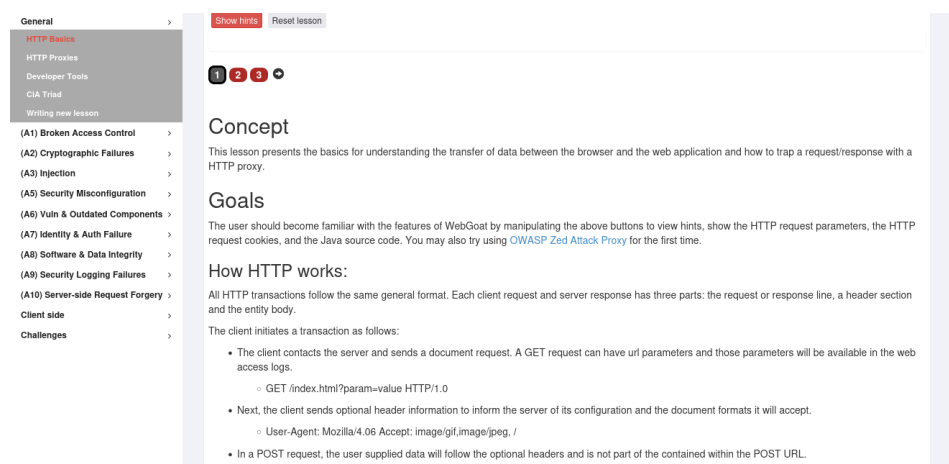
está sendo registrada. Essas informações serão descritografadas quando os computadores quânticos forem poderosos o suficiente. Mesmo que as informações possam ser antigas, elas ainda podem conter informações valiosas que podem ser mal utilizadas. Além do fato de que algumas informações privadas serão conhecidas por terceiros aos quais não foram destinadas.

5. Lições de introdução e telas

Introdução com explicação do WebGoat:



Introdução aos testes de HTTP Basic:



Fundamentos do tratamento de uma solicitação HTTP:

WEBGOAT

Introduction >

General >

HTTP Basics

HTTP Proxies

Developer Tools

CIA Triad

Writing new lesson

(A1) Broken Access Control >

(A2) Cryptographic Failures >

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >

(A9) Security Logging Failures >

HTTP Basics

Search lesson

Show hints Reset lesson

1 2 3

Enter your name in the input field below and press "Go!" to submit. The server will accept the request, reverse the input and display it back to the user, illustrating the basics of handling an HTTP request.

Try It!

Enter your name in the input field below and press "Go!" to submit. The server will accept the request, reverse the input and display it back to the user, illustrating the basics of handling an HTTP request.

Enter Your Name: Evillin Go!

Quiz:

WEBGOAT

Introduction >

General >

HTTP Basics

HTTP Proxies

Developer Tools

CIA Triad

Writing new lesson

(A1) Broken Access Control >

(A2) Cryptographic Failures >

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >

HTTP Basics

Search lesson

Show hints Reset lesson

1 2 3

The Quiz

What type of HTTP command did WebGoat use for this lesson. A POST or a GET.

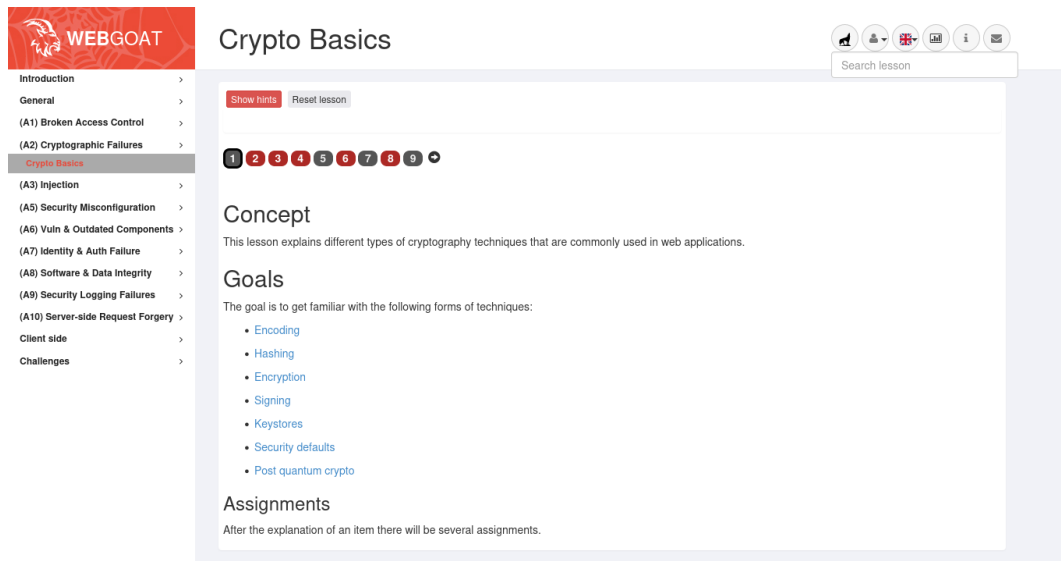
Was the HTTP command a POST or a GET: POST

What is the magic number: 5 Go!

Congratulations. You have successfully completed the assignment.

6. Relatório


ETAPA 1: Parte introdutória, onde fala dos conceitos a serem abordados. Existem diversos tipos de codificação, mas nessa Tarefa escolhida irá abordar apenas base64, XOR e URL



The screenshot displays the WEBGOAT web application interface. On the left is a sidebar menu with the following items: Introduction, General, (A1) Broken Access Control, (A2) Cryptographic Failures, **Crypto Basics** (highlighted), (A3) Injection, (A5) Security Misconfiguration, (A6) Vuln & Outdated Components, (A7) Identity & Auth Failure, (A8) Software & Data Integrity, (A9) Security Logging Failures, (A10) Server-side Request Forgery, Client side, and Challenges. The main content area is titled 'Crypto Basics' and features a progress bar with 9 steps, where step 2 is currently active. The lesson content includes a 'Concept' section explaining cryptography techniques, a 'Goals' section listing techniques like Encoding, Hashing, Encryption, Signing, Keystores, Security defaults, and Post quantum crypto, and an 'Assignments' section. A search bar is located in the top right corner.

ETAPA 2: Codificação não é muito usada na criptografia, porém muito utilizada em padrões entorno da função criptográficas, especialmente na Base64.

A codificação Base64 utilizada para transformar todos os tipos de bytes em um intervalo específico de bytes. Esse intervalo específico são os bytes legíveis ASCII. Assim, você pode transferir dados binários, como chaves secretas ou privadas com mais facilidade. Nessa etapa foi passado um cabeçalho para decodificação, assim depois de muito pesquisar como realizar a decodificação, utilizei um site que retornava meu nome de usuário e uma senha de demonstração, notei que a sequência de letras do código forma uma letra, por exemplo “ZXZ” forma a letra E, tendo assim um padrão para codificação.



Introduction >

General >

(A1) Broken Access Control >

(A2) Cryptographic Failures >

Crypto Basics

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >


(A9) Security Logging Failures >

(A10) Server-side Request Forgery >

Client side >

Challenges >

Crypto Basics



1
2
3
4
5
6
7
8
9

Base64 Encoding

Encoding is not really cryptography, but it is used a lot in all kinds of standards around cryptographic functions. Especially Base64 encoding.

Base64 encoding is a technique used to transform all kinds of bytes to a specific range of bytes. This specific range is the ASCII readable bytes. This way you can transfer binary data such as secret or private keys more easily. You could even print these out or write them down. Encoding is also reversible. So if you have the encoded version, you can create the original version.

On wikipedia you can find more details. Basically it goes through all the bytes and transforms each set of 6 bits into a readable byte (8 bits). The result is that the size of the encoded bytes is increased with about 33%.

```
Hello ==> SGVsbG8=
0x4d 0x61 ==> TW8=
```

Basic Authentication

Basic authentication is sometimes used by web applications. This uses base64 encoding. Therefore, it is important to at least use Transport Layer Security (TLS) or more commonly known as https) to protect others from reading the username password that is sent to the server.

```
$echo -n "myuser:mypassword" | base64
bX11c2VyOm15c6Gzc3dvcnQ=
```


The HTTP header will look like:

```
Authorization: Basic bX11c2VyOm15c6Gzc3dvcnQ=
```

Now suppose you have intercepted the following header:
Authorization: Basic ZXZpaWNyaXNoaW50EjYmZQ1Ng==

Then what was the username and what was the password: post the answer

ETAPA 3: Inicia comentando sobre outros tipos de codificação, sendo eles, URL, HTML, UUE, e XOR encoding. Nessa etapa, foi passado uma senha de um banco de dados codificado, para descobrir, utilizei o WebSphere {xor} password decoder and encoder, tendo como resultado da senha “databasepassword”, notei mais uma vez no padrão de letas, onde o “OZ” retorna a letra D.



Introduction >

General >

(A1) Broken Access Control >

(A2) Cryptographic Failures >

Crypto Basics

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >



(A9) Security Logging Failures >

(A10) Server-side Request Forgery >

Client side >

Challenges >

Crypto Basics

1
2
3
4
5
6
7
8
9

Other Encoding

Also other encodings are used.

URL encoding

URL encoding is used a lot when sending form data and request parameters to the server. Since spaces are not allowed in a URL, this is then replaced by %20. Similar replacements are made for other characters.

HTML encoding

HTML encoding ensures that text is displayed as-is in the browser and not interpreted by the browser as HTML.

UUEncode

The Unix-2-Unix encoding has been used to send email attachments.

XOR encoding

Sometimes encoding is used as a first and simple obfuscation technique for storing passwords. IBM WebSphere Application Server e.g. uses a specific implementation of XOR encoding to store passwords in configuration files. IBM recommends to protect access to these files and to replace the default XOR encoding by your own custom encryption. However when these recommendations are not followed, these defaults can become a vulnerability.

Assignment

Now let's see if you are able to find out the original password from this default XOR encoded string.

Suppose you found the database password encoded as (xor)Oz4rPj0+L.DovPiwsKDA0w==

What would be the actual password

post the answer

Congratulations.

ETAPA 4: Inicia comentando sobre Hash, onde o hash simples é muito utilizado para identificar se dados originais foram alterados e o salted hash para armazenamento em dB. Nessa etapa foram passados dois hash's, porém não especificaram de qual tipo, usando hash type descobri que o primeiro é do tipo MD5 e quando o hash é jogado na barra de pesquisa do google, sites mostram a palavra “passw0rd”. O segundo hash

é do tipo SHA256, fazendo o mesmo passo a passo do anterior, ele retorna a mesma palavra “passw0rd”.

WEBGOAT

Introduction >

General >

(A1) Broken Access Control >

(A2) Cryptographic Failures >

Crypto Basics

(A3) Injection >

(A5) Security Misconfiguration >

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >

(A9) Security Logging Failures >

(A10) Server-side Request Forgery >

Client side >

Challenges >

Crypto Basics

Show hints

Reset lesson

123456789

Plain Hashing

Hashing is a type of cryptography which is mostly used to detect if the original data has been changed. A hash is generated from the original data. It is based on irreversible cryptographic techniques. If the original data is changed by even one byte, the resulting hash is also different.

So in a way it looks like a secure technique. However, it is NOT and even NEVER a good solution when using it for passwords. The problem here is that you can generate passwords from dictionaries and calculate all kinds of variants from these passwords. For each password you can calculate a hash. This can all be stored in large databases. So whenever you find a hash that could be a password, you just look up the hash in the database and find out the password.

Some hashing algorithms should no longer be used: MD5, SHA-1 For these hashes it is possible to change the payload in such a way that it still results in the same hash. This takes a lot of computing power, but is still a feasible option.

Salted Hashes

Plain passwords should obviously not be stored in a database. And the same goes for plain hashes. The [OWASP Password Storage Cheat Sheet](#) explains what should be used when password related information needs to be stored securely.

Assignment

Now let's see if you can find what passwords matches which plain (unsalted) hashes.

✓

Which password belongs to this hash:
BED128365216C019988915ED3ADD75FB

Which password belongs to this hash:
8F0E2F76E22B43E2855189877E7DC1E1E7D98C226C95DB247CD1D547928334A9

Congratulations. You found it!

CONCLUSÃO

Após o processo de configuração e instalação das aplicações já citadas, com o foco em explorar e adquirir conhecimentos em vulnerabilidades em aplicações Web, realizei injeções dentro do ambiente WebGoat, também pude ter a experiência de realizar alguns testes que provavelmente no dia a dia eu não iria fazer e ter a primeira experiência utilizando a VirtualBox.

Dificuldades que encontrei durante o processo, foi em questão de versão de aplicativos, links que haviam expirados e na procura da senha do Linux Kali, pois no documento de apoio não estava informando a senha nem que o sistema operacional possuía alguma senha.

CURITIBA

2023

REFERÊNCIAS

1. Conheça as 10 principais vulnerabilidades web de 2021. Disponível em: <https://blog.4linux.com.br/conheca-as-10-principais-vulnerabilidades-web-de-2021/>
2. Principais tipos de ataques a aplicações web. Disponível em: <https://www.treinaweb.com.br/blog/principais-tipos-de-ataques-a-aplicacoes-web>
3. OWASP WebGoat. Disponível em: <https://owasp.org/www-project-webgoat/>
4. Segurança em aplicação web. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First_steps/Website_security
5. O que é segurança de aplicativos web? Disponível em: <https://www.cloudflare.com/pt-br/learning/security/what-is-web-application-security/>
6. What is SQL injection (SQLi)? Disponível em: [https://portswigger.net/web-security/sql-injection#:~:text=SQL%20injection%20\(SQLi\)%20is%20a,not%20normally%20able%20to%20retrieve.](https://portswigger.net/web-security/sql-injection#:~:text=SQL%20injection%20(SQLi)%20is%20a,not%20normally%20able%20to%20retrieve.)
7. Boas práticas em Cyber Security para aplicações Web. Disponível em: <https://www.eldorado.org.br/blog/boas-praticas-em-cyber-security-para-aplicacoes-web/>
8. Hash type, disponível em: https://hashes.com/en/tools/hash_identifier
9. WebSphere{xor}, disponível em: <https://strelitzia.net/wasXORdecoder/wasXORdecoder.html>
10. Code base64, disponível em: <https://www.base64decode.org/pt/>