

Aula TP - 06/Mai/2019

Cada grupo deve colocar a resposta às perguntas dos seguintes exercícios na área do seu grupo no Github até ao final do dia 20/Mai/2019. Por cada dia de atraso será descontado 0,15 valores à nota desse trabalho.

Note que estes exercícios devem ser feitos na máquina virtual disponibilizada.

Instruções de instalação do *WebGoat*, para quem não utiliza a máquina virtual disponibilizada:

Nesta aula vamos utilizar o WebGoat a partir de uma imagem Docker. O WebGoat é uma aplicação Web deliberadamente insegura mantida pelo OWASP, cujo objetivo é ser um complemento às aulas de segurança de aplicações Web, pelo que contém falhas aplicacionais comuns.

Pode instalar o Docker diretamente no sistema operativo do seu computador.

Se quiser instalar o docker no seu computador siga as instruções em <https://store.docker.com/search?type=edition&offering=community> relativas ao seu sistema operativo.

Caso o sistema operativo do seu computador seja o mesmo da máquina virtual, siga as seguintes instruções para instalar o docker:

1. `sudo apt-get update`
2. `sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common`
3. `curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -`
4. `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"`
5. `sudo apt-get update`
6. `sudo apt-get install docker-ce`

Nota: Sempre que nos exercícios abaixo lhe for pedido para utilizar o WebGoat, deve executar os seguintes passos:

1. Garantir que já instalou o docker no seu computador ou na máquina virtual.
2. Obter a versão 7.1 do WebGoat, através do comando `sudo docker pull webgoat/webgoat-7.1`
3. Arrancar o WebGoat com o comando `sudo docker run -p 8080:8080 -t webgoat/webgoat-7.1`.

Note que o WebGoat demora 30 - 50 segundos a arrancar, tendo arrancado quando lhe aparecer no ecrã uma linha que contenha a seguinte informação `INFO: Starting ProtocolHandler ["http-bio-8080"]`

4. No Firefox da máquina virtual aceder a `http://localhost:8080/WebGoat`.

No final, para parar o docker e WebGoat, deverá efectuar os seguintes comandos:

- `sudo docker ps` para obter o Container ID
- `sudo docker kill <Container ID>` em que `<Container ID>` é o valor indicado no comando anterior para

esse campo.

Exercícios

1. *Injection*

Experiência 1.1

Aceda a <https://free.codebashing.com/courses/java> e siga o exemplo de *SQL Injection*.

Experiência 1.2

Aceda a <https://free.codebashing.com/courses/dotnet> e siga o exemplo de *XXE Injection*.

Pergunta 1.1 - *String SQL Injection*

No WebGoat tente resolver o exercício *Injection Flaws -> String SQL Injection*:

1. O objetivo é utilizar *SQL Injection* que resulte em todos os cartões de crédito serem apresentados. Experimente com alguns nomes e veja como se comporta.
2. Execute o ataque de *SQL Injection* - experimente utilizar uma tautologia.

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.

Pergunta 1.2 - *Numeric SQL Injection*

No WebGoat tente resolver o exercício *Injection Flaws -> Numeric SQL Injection*:

1. O objetivo é utilizar *SQL Injection* que resulte em todos os dados meteorológicos serem apresentados. Experimente com alguns locais e veja como se comporta.
2. Execute o ataque de *SQL Injection* - experimente utilizar uma tautologia. O problema é que aparentemente não temos maneira de alterar a query ...
3. Talvez se possa utilizar as ferramentas de desenvolvimento do Firefox (canto superior direito do browser e selecione o botão "Developer", seguido da ferramenta "Inspector"). Mas como é que modificamos a informação que é enviada quando clicamos no botão "Go!"?
4. Talvez possa começar por procurar "Columbia" no código. A seguir altere o HTML para executar a tautologia.

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.

Pergunta 1.3 - *Database Backdoors*

No WebGoat tente resolver o exercício *Injection Flaws -> Database Backdoors*, um passo de cada vez.

- Passo 1

1. O objetivo é utilizar *SQL Injection* para executar mais do que um comando SQL, para uma conta com o ID 101. Experimente como é que a aplicação se comporta com este ID.
2. Tente explorar a vulnerabilidade para alterar o salário para um valor mais elevado.

Relembre-se que o SQL UPDATE tem o seguinte formato:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

- Passo 2

3. O objetivo é relativamente simples. Efetue o ataque com o comando SQL indicado para alterar o salário para um valor mais elevado.

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.

2. XSS

Experiência 2.1

Aceda a <https://free.codebashing.com/courses/nodejs> e siga o exemplo de *Persistent (Stored) XSS*.

Pergunta 2.1 - *Reflected XSS*

No WebGoat tente resolver o exercício *Cross-Site Scripting (XSS) -> Reflected XSS Attacks*:

1. O objetivo é utilizar o formulário de compras para refletir o input do utilizador. Como é que se pode fazer?
2. Este formulário é muito similar ao que pode encontrar em diversos sites. Tem uma lista de produtos, preços e quantidades. É possível mudar a quantidade e ver o preço alterado. No final, o utilizador tem que fornecer o cartão de crédito e o CCV para pagar os bens selecionados.
3. Por tentativa e erro, experimente combinação de letras e dígitos nos vários campos de input para ver que tipo de validações são feitas pelo servidor. Se encontrar algum campo com potencial para Reflected XSS, tente a seguinte script: `<script> alert("SSA!!!")</script>`

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.

Experiência 2.2 - *Stored XSS*

No WebGoat tente resolver o exercício *Cross-Site Scripting (XSS) -> Stored XSS Attacks*:

1. O objetivo é deixar uma mensagem no seu site favorito que desencadeie uma ação quando outro utilizador for ler a sua mensagem. Como é que se pode fazer?
2. Faça alguns testes no formulário de mensagens para verificar o tipo de validação e filtragem do input que é efetuado.
3. Se achar que tem potencial para Stored XSS, tente a seguinte script: `<script language="javascript" type="text/javascript">alert(document.cookie);</script>` que vai

apresentar os cookies num Popup.

4. Verifique se funcionou.
5. Como é que poderia um atacante beneficiar desta vulnerabilidade?

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.

3. Quebra na Autenticação

Pergunta 3.1 - *Forgot Password*

No WebGoat tente resolver o exercício *Authentication flaws -> Forgot Password*.

Os mecanismos de autenticação têm usualmente a funcionalidade de permitirem recuperar a password através da resposta a uma questão pessoal. Contudo, muitas vezes é simples de adivinhar a resposta à questão!

1. A aplicação WebGoat permite que os utilizadores recuperem a password se conseguirem responder a uma questão sobre a sua cor favorita. Tente o utilizador "webgoat" e cor "red" e veja o que acontece.
2. Agora tente atacar outro utilizador. Que utilizadores existem normalmente num sistema? De todos esses utilizadores, qual era a conta que gostaria de comprometer? Tente essa.
3. Como é que pode obter a password dessa conta? Talvez possa tentar acertar na cor por força bruta ...

Em todos os exercícios tente primeiro resolver sem ajuda. Se chegar a um ponto em que não consegue avançar, utilize "Show Hints" para ajuda. Apenas deve utilizar "Show Solution" em situações extremas.