

Sessão 3: Teste do Mecanismo de Autenticação

1. Atividade – Tecnologias de autenticação

Esta atividade tem por objetivo abordar tecnologias de autenticação empregadas em aplicações web, dando base para que o aluno realize as atividades de descoberta de vulnerabilidades e exploração. Para iniciá-la, carregue as máquinas virtuais do aluno e do servidor (Fedora) e execute os roteiros na primeira delas.

Avaliação dos aspectos de autenticação

Acesse a página web de onde trabalhe e avalie os seguintes aspectos relacionados ao processo de autenticação de usuários:



Comentário: Essa resposta é pessoal pois irá depender da página que cada aluno irá escolher para avaliar. Via de regra as respostas mais comuns seriam:

- Qual o protocolo utilizado no carregamento da página de autenticação?



Resposta: http ou https

- Que tipo de mecanismo de autenticação é empregado?



Resposta: usuário e senha

- Como é o processo de recuperação de senhas?



Resposta: envia um email para a conta cadastrada com um token

- A aplicação utiliza uma política de senhas fortes? Quais são os critérios adotados?



Resposta: Não. Não tem critérios de senha forte

- Como as senhas são protegidas durante o armazenamento?



Resposta: em um banco de dados relacional utilizando alguma função hash

- As credenciais de acesso são enviadas por canal seguro?



Resposta: vai depender se o servidor for http ou https. Se for http o canal não é seguro.

- Existe uma funcionalidade para lembrar o usuário?



Resposta: normalmente é utilizado cookie para este fim.

2. Atividade – Descoberta de vulnerabilidades e exploração

O propósito desta atividade é introduzir ao aluno os métodos que podem ser utilizados para a descoberta e exploração de vulnerabilidades, em mecanismos de autenticação. Todos os exercícios devem ser realizados na máquina virtual do aluno, e é altamente recomendado que se tente traçar a estratégia de exploração, antes de seguir o roteiro fornecido.

Uso de informações obtidas nas fases de reconhecimento e mapeamento

Neste exercício, informações encontradas nas etapas de reconhecimento e mapeamento são utilizadas para se obter acesso ao sistema.

Parte I – Arquivos contendo credenciais de acesso

1. Abra uma janela de terminal.
2. Visualize o conteúdo do arquivo `robots.txt`, da aplicação Mutillidae:

```
~$ curl http://mutillidae.esr.rnp.br/robots.txt
```

Observe que há um diretório `/passwords` e um arquivo `config.inc`.

3. Encerre a janela de terminal.
4. Inicie o Firefox, presente no menu Usual applications\Internet.
5. Acesse <http://mutillidae.esr.rnp.br/passwords>.
6. Visualize o conteúdo do arquivo `accounts.txt`.
7. Anote as senhas dos usuários `admin` e `john`.
8. Acesse <http://mutillidae.esr.rnp.br>.
9. Clique em `Login`.
10. Tente se conectar como o usuário `admin` e veja a mensagem no topo da tela.
11. Clique novamente em `Login` e repita o passo anterior, para o usuário `john`.

Parte II – Pistas no código

1. Clique no marcador WebGoat e forneça `guest` e `guest`, quando usuário e senha forem solicitados.
2. Clique em Start WebGoat.
3. No menu, ao lado esquerdo, clique em `Code Quality` e, depois, em `Discover Clues in the HTML`. No formulário clique no link `Restart this Lesson`.
4. Visualize o código-fonte da página, pressionando `Ctrl+U`.
5. Pressione `Ctrl+F`, para realizar uma busca, e digite `<!--` no campo `Find`.
6. Clique repetidamente em `Next`, até encontrar alguma informação interessante.
7. Feche a janela de código-fonte.
8. Tente se autenticar com as credenciais encontradas.
9. O exercício é finalizado com sucesso e a mensagem `* Congratulations. You have successfully completed this lesson` é exibida.
10. Encerre o Firefox.

Usuário e senha padronizados

O objetivo deste exercício é utilizar contas pré-definidas para acesso a aplicações e serviços.

1. Inicie o Firefox, presente no menu Usual applications\Internet .
2. Acesse a aplicação Tomcat Web Application Manager em:

`http://exemplo.esr.rnp.br:8080/manager/html`
3. Tente se autenticar com uma das senhas fornecidas na Tabela 5 (figura 3.9 da apostila).
4. Encerre o Firefox.

Enumeração de identificadores de usuários

O foco desta prática são as técnicas de enumeração de identificadores de usuários, os quais são necessários para executar diversos outros ataques. O exercício está dividido em duas partes, ambas baseadas no mesmo formulário de autenticação.

Parte I – Enumeração via navegador

1. Inicie o Firefox, presente no menu Usual applications\Internet .
2. Acesse `http://form-auth.esr.rnp.br/`.
3. Digite seu primeiro nome nos campos Usuário e Senha e clique em Login .
4. Anote a mensagem de erro e clique em Retornar à página de login .
5. Digite `guest` nos campos usuário e senha, e clique em Login .
6. Qual a diferença em relação à mensagem de erro anterior? Como isso pode ser utilizado para enumerar usuários?



Resposta: o mecanismo permite identificar quando um usuário está cadastrado já que a mensagem informa que a senha está errada.

7. Clique em Retornar à página de login .
8. Digite seu primeiro nome no campo Usuário e clique em Esqueci minha senha .
9. Anote a mensagem de erro e clique em Retornar à página de login .
10. Digite `guest` no campo Usuário e clique em Esqueci minha senha .
11. O mecanismo de recuperação de senhas também permite enumerar usuários?



Resposta: o mecanismo permite identificar quando um usuário está cadastrado já que a mensagem informa que a conta não existe.

12. Clique no ícone Retornar do Firefox.
13. Encerre o Firefox.

Parte II – Enumeração via script

1. Abra uma janela de terminal.
2. Digite o comando:

```
~$ cd ~/Arquivos\ do\ Curso/sessao-03/
```

3. Requisite a página de autenticação e identifique os elementos de entrada e o script que realiza a validação das credenciais:

```
~$ curl http://form-auth.esr.rnp.br
```

4. Visualize o script de enumeração e como os elementos identificados no passo anterior são utilizados:

```
~$ cat enumerate
```

5. Visualize a lista de identificadores de usuário:

```
~$ cat ids
```

6. Execute o script de enumeração:

```
~$ ./enumerate
```

Foi possível encontrar credenciais completas no teste?



Resposta: apenas a senha do usuário admin

7. Encerre a janela de terminal.

Mecanismo vulnerável de recuperação de senhas



Neste exercício, o leitor explorará o mecanismo vulnerável de recuperação de senhas, por meio de um ataque de força bruta contra o conjunto de respostas.

1. Inicie o Firefox, presente no menu Usual applications\Internet .
2. Acesse <http://form-auth.esr.rnp.br/>.
3. Digite `usuario` no campo Usuário e clique em `Esqueci minha senha` .
4. Teste todas as cores possíveis, até encontrar a resposta correta.
5. Anote a senha apresentada e clique em `Retornar à página de login` .
6. Tente se autenticar com a senha recuperada.
7. Quais são as vulnerabilidades presentes nesse exemplo?



Resposta: É muito fácil adivinhar a senha já que poucas são as cores que uma pessoa normal tem como favoritas.

Para o usuário `admin` a cor é azul e para o usuário `usuario` a cor é verde

+ . Encerre o Firefox.

Transporte inseguro de credenciais de acesso

O objetivo deste exercício é comparar o transporte de credenciais de acesso por canais seguros contra aqueles sem segurança nenhuma.

1. Inicie o Firefox, presente no menu .
2. Inicie o Wireshark em modo privilegiado. Para isso abra um terminal e execute:

```
~$ sudo wireshark
```

3. Clique no menu **Capture** e em **Options**. Selecione a interface **eth1** e no campo **Capture filter**, digite **tcp port http** e clique em **Start**, para iniciar a captura de pacotes.
4. Acesse com o Firefox a página <http://form-auth.esr.rnp.br/>.
5. Digite **admin** nos campos **Usuário** e **Senha** e clique em **Login**.
6. Pare a captura de pacotes no Wireshark, clicando no quarto botão da barra de ferramentas (**Stop the running live capture**).
7. Procure pela linha contendo **POST /login.php** e a selecione.
8. Na segunda parte da tela, expanda o item **HTML Form URL Encoded** e observe que as credenciais são transmitidas sem nenhuma proteção.
9. Clique no menu **Capture** e em **Options**. Selecione a interface **eth1** e no campo **Capture filter**, digite **tcp port https** e clique em **Start**, para iniciar a captura de pacotes.
10. Clique em **Continue without Saving**.
11. Acesse com o Firefox a página <https://w3s.esr.rnp.br/basic>. Caso apareça a mensagem **Your connection is not secure**, clique no botão **Advanced** → **Add Exception**. Desmarque a opção **Permanently store this exception** e clique em **Confirm Security Exception**.
12. Digite **esruser** tanto para usuário como para senha e clique em **OK**.
13. Pare a captura de pacotes no Wireshark, clicando no quarto botão da barra de ferramentas (**Stop the running live capture**).
14. Procure pela primeira linha contendo **Application data**, **Application data** e a selecione.
15. Na segunda parte da tela, expanda o item **Secure Socket Layer** e observe que todo o conteúdo está cifrado.
16. Encerre o Wireshark e o Firefox.

Falhas na implementação do mecanismo

Conforme visto, erros na lógica do código que trata a autenticação de usuários podem resultar na completa evasão do mecanismo. Nesse contexto, a presente atividade aborda dois cenários, cada um baseado em um problema diferente.

Parte I – Falha de maneira insegura

1. Inicie o WebScarab, presente no menu **03 - Web Application Analysis**.
2. Inicie o Firefox, presente no menu **Usual applications\Internet**.
3. No Firefox, clique no **Multiproxy SwitchOmega**, na barra de estado, e selecione o WebScarab.
4. Acesse o WebGoat, por meio da barra de atalhos.
5. Forneça **guest** para **Usuário** e **Senha**.
6. Clique em **Start WebGoat**.
7. No menu do lado esquerdo, clique em **Improper Error Handling** e, em seguida, em **Fail Open Authentication Scheme**. Nesta página clique no link **Restart this Lesson**.
8. Digite **webgoat** e **password** para os campos **User Name** e **Password**, respectivamente, e clique em **Login**. A autenticação foi bem sucedida?



Resposta: Não

9. No WebScarab, clique na aba Proxy e marque Intercept Requests .
10. Retorne ao Firefox, digite webgoat para os campos User Name e Password e clique em Login .
11. A tela de interceptação do WebScarab aparece. Selecione a linha contendo a variável Password e clique em Delete . O objetivo é induzir um erro na aplicação, devido à falta de um parâmetro esperado.
12. Clique em Accept Changes . O que acontece?



Resposta: agora foi possível pois a aplicação não estava preparada para receber requisições sem campos obrigatórios.

13. No WebScarab, clique na aba Proxy e desmarque Intercept Requests .
14. Clique no Multiproxy Switch, na barra de estado, e selecione Direct .
15. Encerre o WebScarab.

Parte II – Injeção de SQL

1. Acesse o Mutillidae, com o Firefox, por meio da barra de atalhos.
2. No menu do lado esquerdo, clique em Login .
3. Digite uma aspa simples (') no campo Name e qualquer valor no campo Password .
4. Clique em Submit .
5. A mensagem de erro exibida indica que aplicação é vulnerável à injeção de SQL, tema do Capítulo 6.
6. Clique no botão Retornar do Firefox.
7. Digite ' or 1=1# no campo Name e qualquer valor no campo Password .
8. Clique em Submit .
9. Devido ao problema no tratamento das entradas, foi possível se conectar como admin , sem conhecimento de nenhum identificador de usuário.
10. Encerre o Firefox.

Autenticação com múltiplos fatores

O objetivo deste exercício é explorar uma vulnerabilidade no mecanismo de autenticação com múltiplos fatores, para conseguir conectar-se como outro usuário, sem conhecimento da senha e da cartela por ele utilizadas.



Transaction Authorization Number é uma senha descartável, utilizada como segundo fator de autenticação, distribuída, normalmente no formato de uma cartela contendo vários valores distintos. Cada usuário da aplicação recebe um conjunto diferente, gerado de maneira aleatória.

1. Inicie o WebScarab, presente no menu 03 - Web Application Analysis .
2. Inicie o Firefox, presente no menu Usual applications\Internet .
3. No Firefox, clique no Multiproxy SwitchOmega, na barra de estado, e selecione o WebScarab.
4. Acesse o WebGoat, por meio da barra de atalhos.
5. Forneça guest para Usuário e Senha.
6. Clique em Start WebGoat .
7. No menu do lado esquerdo, clique em Authentication Flaws e, em seguida, em Multi Level Login . No formulário clique em Restart this Lesson

8. Digite Jane e tarzan nos campos Name e Password, respectivamente, e clique em Submit.
9. Forneça o TAN solicitado, a partir da lista apresentada na tela, e clique em Submit.
10. No menu do lado esquerdo, clique em Multi Level Login 2.
11. Digite Joe e banana nos campos Name e Password, respectivamente, e clique em Submit.
12. No WebScarab, clique na aba Proxy e marque Intercept Requests.
13. Retorne ao Firefox, digite o TAN solicitado, a partir da lista apresentada na tela, e clique em Submit.
14. Altere o valor do parâmetro hidden_user para Jane e clique em Accept changes.
15. Que vulnerabilidade permitiu a realização do ataque?



Resposta: quem programou o sistema injetou dentro de um campo hidden no formulário a identificação do usuário.

16. No WebScarab, clique na aba Proxy e desmarque Intercept Requests.
17. Clique no Multiproxy Switch, na barra de estado, e selecione Direct.
18. Encerre o WebScarab e o Firefox.

Ataque de força bruta

Um problema existente nos métodos de autenticação Basic e Digest do protocolo HTTP é a falta de um mecanismo de bloqueio de conta, quando múltiplas tentativas de autenticação falham, de maneira consecutiva. Esse problema será explorado neste exercício, para quebrar a senha muito curta de uma conta da aplicação.

Como ataques de força bruta podem demorar várias horas, vamos partir do pressuposto que o sistema possui um usuário chamado esr e que sua senha tem 3 caracteres minúsculos.

1. Primeiro vamos dar uma olhada na ajuda do comando hydra, em especial em como configurar o parâmetro -x que estabelece as regras para um ataque de força bruta.

```
~$ hydra -x -h
```

2. De acordo com nosso pressuposto, iremos implementar nosso ataque utilizando o valor 3:3:a para o parâmetro -x o que indica que nosso usuário alvo tem uma senha de 3 caracteres minúsculos. Como já sabemos a URL `http://exemplo.esr.rnp.br/basic` hospeda uma autenticação HTTP portando já podemos montar o comando de ataque:

```
~$ hydra -l esr -x 3:3:a exemplo.esr.rnp.br http-get /basic/
```

3. Será necessário aguardar alguns minutos para que a ferramenta execute as tentativas de acesso com a conta esr e senha utilizando técnica de força bruta.

Obs.: A senha do usuário esr é juk caso não deseje aguarda a conclusão deste teste.

Ataque de dicionário

Ataques de dicionário são mais eficientes que os de força bruta, pois somente senhas prováveis são testadas. Vejamos algumas ferramentas que podem ser utilizadas com esse propósito.

Parte I – Hydra e Medusa

1. Abra uma janela de terminal.

2. Digite o comando:

```
~$ cd ~/Arquivos\ do\ Curso/sessao-03/
```

3. Visualize o conteúdo do arquivo ids:

```
~$ cat ids
```

4. Visualize o conteúdo do arquivo pwds:

```
~$ cat pwds
```

5. Veja as opções que podem ser utilizadas com o utilitário Medusa:

```
~$ medusa
```

6. Para realizar um ataque de dicionário contra o método de autenticação Basic, com o utilitário Medusa, digite o comando:

```
~$ medusa -h exemplo.esr.rnp.br -U ids -P pwds -e ns -M http -m DIR:basic/ -v 4
```

Identifique o propósito de cada opção utilizada.

7. Veja as opções que podem ser utilizadas com o utilitário Hydra:

```
~$ hydra
```

8. Para realizar um ataque de dicionário contra o método de autenticação Digest, com o utilitário Hydra, digite o comando:

```
~$ hydra -L ids -P pwds -e ns exemplo.esr.rnp.br http-get /digestauth
```

Identifique o propósito de cada opção utilizada.

Parte II – Script

1. Requisite a página de autenticação e identifique os elementos de entrada e o script que realiza a validação das credenciais:

```
~$ curl http://form-auth.esr.rnp.br
```

Esse é o mesmo formulário utilizado no teste de enumeração.

2. Visualize o script de ataque de dicionário e como os elementos identificados no passo anterior são utilizados:

```
~$ cat dict
```

3. Execute o script de enumeração:

```
~$ ./dict
```

4. Encerre a janela de terminal.

Ataque contra senhas armazenadas

Neste exercício, o leitor utilizará o programa John the Ripper, para quebrar o arquivo de senhas utilizado pelo Apache, no modo de autenticação Basic, e, também, realizará um ataque baseado em Rainbow Tables.

Parte I – John the Ripper

1. Abra uma janela de terminal.

2. Digite o comando:

```
~$ cd ~/Arquivos\ do\ Curso/sessao-03/
```

3. Visualize o conteúdo do arquivo passwords :

```
~$ cat passwords
```

4. Veja as opções que podem ser utilizadas com o John the Ripper:

```
~$ /usr/sbin/john
```


5. Digite o comando abaixo para que o utilitário tente efetuar a quebra do arquivo de senhas:

```
~$ sudo /usr/sbin/john passwords
```

Todas as senhas foram quebradas? Qual o tempo gasto na tarefa?

Foram quebradas todas as senhas em 30 segundos.

Tabela 1. Lista de Usuários



Senha	Usuário
esruser	esruser
juk	esr
senhad	hard

6. Encerre a janela de terminal.

Parte II – Rainbow Tables

1. Gere o MD5 de uma senha que contenha exatamente seis letras minúsculas (exemplo: aesrrnp):

```
~$ echo -n aesrrnp | md5sum
```

2. Selecione o MD5 calculado e o copie para a área de transferência, pressionando Ctrl+- Shift+ -C.

3. Digite o comando abaixo para recuperar o texto original, por meio de Rainbow Tables:

```
~$ rcracki_mt -h <MD5_gerado_no_Passo_2> /usr/share/rainbowcrack/*.rt - v
```

Obs.: tente repetir este processo para outras senhas que iniciem com a letra **a** e que tenha de 6 a 8 caracteres minúsculos. Depois tente com uma senha qualquer iniciando com a letra **z**. Foi possível identificar esta última senha? Se não foi, qual seria a razão mais provável?



Não foi possível identificar a senha com a letra **z**. Isso aconteceu porque dentro da pasta `/usr/share/rainbowcrack/` não temos todas as combinações possíveis.

4. Encerre a janela de terminal.

As tabelas utilizadas neste exercício foram geradas com os comandos `rtgen` abaixo e consumiram, ao todo, mais de 10 horas de processamento para uma senha que deve ter de 6 a 8 letras minúsculas e/ou números. Cabe destacar que estas tabelas não possuem todas as senhas possíveis neste espaço de força bruta.

+

```
~$ sudo rtgen md5 loweralpha-numeric 6 8 0 3800 33445532 0
~$ sudo rtgen md5 loweralpha-numeric 6 8 1 3800 33445532 0
~$ sudo rtgen md5 loweralpha-numeric 6 8 2 3800 33445532 0
```



+ O formato deste comando é:

+

```
~$ rtgen hash_algorithm charset plaintext_length_min plaintext_length_max table_index
chain_len chain_num part_index
```

+ Após criar as tabelas execute o comando `rtsort` para reordena-las:

+

```
~$ rtsort /usr/share/rainbowcrack/
```

+ Para mais informações acesse o site: <http://project-rainbowcrack.com/generate.htm>

ENTREGA DA TAREFA

Para que seja considerada entregue você deve anexar a esta atividade no AVA o resultado do comando:

```
~$ hydra -l esr -x 3:3:a exemplo.esr.rnp.br http-get /basic/
```

Obs.: O arquivo resultado pode estar em formato de imagem ou texto

Última atualização 2020-08-27 19:05:06 -0300