

**FACULDADE DE TECNOLOGIA SENAC GOIÁS**  
**Segurança da Informação**



Aldo Brito  
Leniker Lettierre  
Matheus Mello  
Rony Carneiro

**ANÁLISE DE VULNERABILIDADE**

Francisco Calça

GOIÂNIA,  
2017

Aldo Brito  
Leniker Littierre  
Matheus Mello  
Rony Carneiro

# **ANÁLISE DE VULNERABILIDADE**

Relatório apresentado como requisito parcial para  
obtenção de aprovação no Projeto Integrador, no  
Curso de Segurança da Informação, na Faculdade de  
Tecnologia Senac Goiás.

Francisco Calaça

GOIÂNIA, 2017

# SUMÁRIO

1	VUNERABILIDADES	4
1.1	INJECTION	4
1.1.1	EXPLORAÇÃO DA VULNERABILIDADE	4
1.1.2	COMO MITIGAR ESTA VULNERABILIDADE	5
1.2	BROKEN AUTHENTICATION AND SESSION MANAGEMENT	6
1.2.1	EXPLORAÇÃO DA VULNERABILIDADE	6
1.2.2	COMO MITIGAR ESTA VULNERABILIDADE	8
1.3	CROSS-SITE SCRIPTING (XSS)	8
1.3.1	EXPLORAÇÃO DA VULNERABILIDADE	8
1.3.2	COMO MITIGAR ESTA VULNERABILIDADE	9
1.4	BROKEN ACCESS CONTROL	10
1.4.1	EXPLORAÇÃO DA VULNERABILIDADE	10
1.4.2	COMO MITIGAR ESTA VULNERABILIDADE	11
1.5	SECURITY MISCONFIGURATION	11
1.5.1	EXPLORAÇÃO DA VULNERABILIDADE	12
1.5.2	COMO MITIGAR ESTA VULNERABILIDADE	13
1.6	BRUTE FORCE	13
1.6.1	COMO FOI EXPLORADO A VULNERABILIDADE	13
1.6.2	COMO MITIGAR ESTA VULNERABILIDADE	14
2	CONCLUSÃO	14
3	REFERÊNCIAS BIBLIOGRÁFICAS	14

# **1 VUNERABILIDADES**

Neste relatório iremos identificar algumas vulnerabilidades existentes na aplicação web no site: <https://vdmcorp.com.br/> que se encontra com o WAF ativo, e levantar evidencias destas vulnerabilidades e apresentação da proposta de mitigação das vulnerabilidades.

## **1.1 INJECTION**

Um ataque de injeção SQL consiste em inserção ou "injeção" de uma consulta SQL através dos dados de entrada do cliente para a aplicação. Uma exploração de injeção SQL bem-sucedida pode ler dados confidenciais do banco de dados, modificar dados do banco de dados (Inserir / Atualizar / Excluir), executar operações de administração no banco de dados (como encerrar o SGBD), recuperar o conteúdo de um determinado arquivo presente no arquivo DBMS Sistema e em alguns casos emite comandos para o sistema operacional. Os ataques de injeção SQL são um tipo de ataque de injeção, no qual os comandos SQL são injetados na entrada do plano de dados para efetuar a execução de comandos SQL predefinidos.

SQL Injection é muito comum com PHP e aplicativos ASP devido à prevalência de interfaces funcionais antigas. Devido à natureza das interfaces programáticas disponíveis, as aplicações J2EE e ASP.NET são menos propensas a implantar injeções de SQL facilmente exploradas.

A gravidade dos ataques de Injeção de SQL é limitada pela habilidade e imaginação do atacante, e em menor medida, contramedidas de defesa em profundidade, como conexões de baixo privilégio para o servidor de banco de dados e assim por diante. Em geral, considere a Injeção SQL como uma gravidade de alto impacto.

### **1.1.1 EXPLORAÇÃO DA VULNERABILIDADE**

Os ataques com a injeção de código sql no servidor <https://vdmcorp.com.br/> foi explorada a partir da tela de login:




## VDM - Sistema de Plano de Ensino

Login:

Senha:

Entrar

Verificado que o acesso é bloqueado:

 <https://vdmcorp.com.br/BackEnd/login.php>

# Forbidden

You don't have permission to access /BackEnd/login.php on this server.

*Apache/2.4.7 (Ubuntu) Server at vdmcorp.com.br Port 443*

Nesta exploração não foi possível obter o acesso ao sistema, por meio de injeção de códigos SQL na tela de Login.

### 1.1.2 COMO MITIGAR ESTA VULNERABILIDADE

Algumas providências devem ser tomadas para amenizar a utilização da SQL Injection, algumas das ações devem ser realizadas no servidor de banco de dados, outras devem ser garantidas pelo código fonte.

Deve-se tomar cuidado com a configuração do usuário que estabelece a conexão com o banco de dados. O ideal é que as permissões de acesso deste usuário estejam estritamente limitadas às funções que irá realizar, ou seja, para a exibição de um relatório, a conexão com o banco de dados deve ser realizada por um usuário com permissões de leitura e acesso somente às tabelas necessárias para sua operação.

Todos os valores originados da coleta de dados externos, devem ser validados e tratados a fim de impedir a execução de eventuais instruções destrutivas ou operações que não sejam as esperadas.

## **1.2 BROKEN AUTHENTICATION AND SESSION MANAGEMENT**

Autenticação e gerenciamento de sessão inclui todos os aspectos de lidar com a autenticação do usuário e gerenciamento de sessões ativas. A autenticação é um aspecto crítico deste processo, mas mecanismos de autenticação, mesmo sólidos pode ser prejudicada por funções de gerenciamento de credenciais falhos, incluindo alteração de senha, esqueci minha senha, lembro da minha senha, atualizações da conta, e outras funções relacionadas. Porque “caminhada por” ataques são prováveis para muitas aplicações web, todas as funções de gerenciamento de contas devem exigir uma nova autenticação, mesmo se o usuário tem um ID de sessão válido.

A autenticação do usuário na web geralmente envolve o uso de um ID de usuário e senha. Métodos de autenticação mais fortes encontram-se comercialmente disponíveis, tais como software e de hardware com base tokens criptográficos ou biometria, mas esses mecanismos são custos proibitivos para a maioria das aplicações de web. Uma grande variedade de contas e gerenciamento de sessão falhas podem resultar no comprometimento de contas de usuário ou de administração do sistema. As equipes de desenvolvimento frequentemente subestimam a complexidade de projetar um esquema de gerenciamento de autenticação e sessão que protege adequadamente credenciais em todos os aspectos do site. Aplicações Web devem estabelecer sessões para manter o controle do fluxo de pedidos de cada usuário. HTTP não fornece essa capacidade. Frequentemente aplicações de desenvolvimento provêm mecanismos para realizar controle de sessão, mas muitos desenvolvedores preferem criar seus próprios tokens de sessão. Em ambos os casos, se os tokens de sessão não são devidamente protegidos, um atacante pode sequestrar uma sessão ativa e assumir a identidade de um usuário. Criando um esquema para criar tokens de sessão fortes e protegê-los durante todo seu ciclo de vida provou indescritível para muitos desenvolvedores. A menos que todas as credenciais de autenticação e os identificadores de sessão são protegidos com SSL em todos os momentos e protegido contra a divulgação de outras falhas, como cross site scripting, um atacante pode sequestrar uma sessão do usuário e assumir a sua identidade.

Todos os servidores conhecidos da web, servidores de aplicação e ambientes de aplicações web são suscetíveis à quebras de autenticação e gerenciamento de sessão.

### **1.2.1 EXPLORAÇÃO DA VULNERABILIDADE**

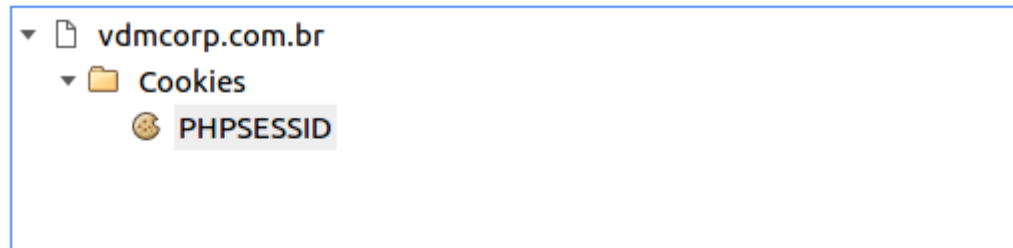
Foi identificado um cookie no navegador:

## Cookies instalados por esta página



**Permitido** Bloqueados

Os seguintes cookies foram definidos quando você visualizou esta página:



Bloquear

Remover

Nome: PHPSESSID

Conteúdo: di5q549o2rqoidtfn91dts6005

Domínio: vdmcorp.com.br

Caminho: /

Enviar para: Qualquer tipo de conexão

Realizado aquisição de um cookies de sessão com a permissão de acesso, utilizado a ferramenta Burp suite para realizar a toca dos cookies:

```
GET /fonts/Exo2/Exo2-Regular.ttf HTTP/1.1
Host: vdmcorp.com.br
User-Agent: Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: application/font-woff2;q=1.0,application/font-woff;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Referer: https://vdmcorp.com.br/_css/estilo.css
Cookie: PHPSESSID=49nn87dg5tf3g4jbq8a2lg0iq2
Connection: close

GET /fonts/Exo2/Exo2-Regular.ttf HTTP/1.1
Host: vdmcorp.com.br
User-Agent: Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: application/font-woff2;q=1.0,application/font-woff;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Referer: https://vdmcorp.com.br/_css/estilo.css
Cookie: PHPSESSID=di5q549o2rqoidtfn91dts6005
Connection: close
```

Verificado que o acesso foi bloqueado:



# Forbidden

You don't have permission to access / on this server.

---

*Apache/2.4.7 (Ubuntu) Server at vdmcorp.com.br Port 443*

## 1.2.2 COMO MITIGAR ESTA VULNERABILIDADE

O uso cuidadoso e adequado de mecanismos de gerenciamento de autenticação e de sessão personalizados ou fora da prateleira deve reduzir significativamente a probabilidade de um problema nessa área. Definir e documentar a política do seu site no que diz respeito à gestão segura credenciais de usuários é um bom primeiro passo. Garantir que a sua implementação reforça consistentemente esta política é a chave para ter um mecanismo de autenticação e gerenciamento de sessão seguro e robusto.

## 1.3 CROSS-SITE SCRIPTING (XSS)

Cross-Site Scripting (XSS) são um tipo de injeção. Ataques XSS ocorrem quando um invasor usa uma aplicação web para enviar código malicioso, geralmente na forma de um script do lado do navegador, para um usuário final diferente. As falhas que permitem que esses ataques sejam bem-sucedidas são bastante difundidas e ocorrem em qualquer lugar onde uma aplicação web usa a entrada de um usuário como saída sem validá-la ou codificar.

Um invasor pode usar XSS para enviar um script malicioso para um usuário desavisado. O navegador do usuário final não tem nenhuma maneira de saber que o script não deve ser confiável, e irá executar o script. Porque ele acha que o roteiro veio de uma fonte confiável, o script malicioso pode acessar os cookies, tokens de sessão, ou outras informações confidenciais retidas pelo navegador e usados com esse site. Esses scripts podem até reescrever o conteúdo da página HTML.

### 1.3.1 EXPLORAÇÃO DA VULNERABILIDADE

A vulnerabilidade foi explorada a partir que o do campo de cadastro de professor:



https://vdmcorp.com.br/insProfessor.php

VDM CORP

Usuário: Fabricio

### Novo Professor

Matrícula: 33443

Professor: alert("Hello! Eu sou o Curinga!");

Usuário: chiccox

### Endereço

CEP: 74085-010

Cidade: Goiânia

UF: GO

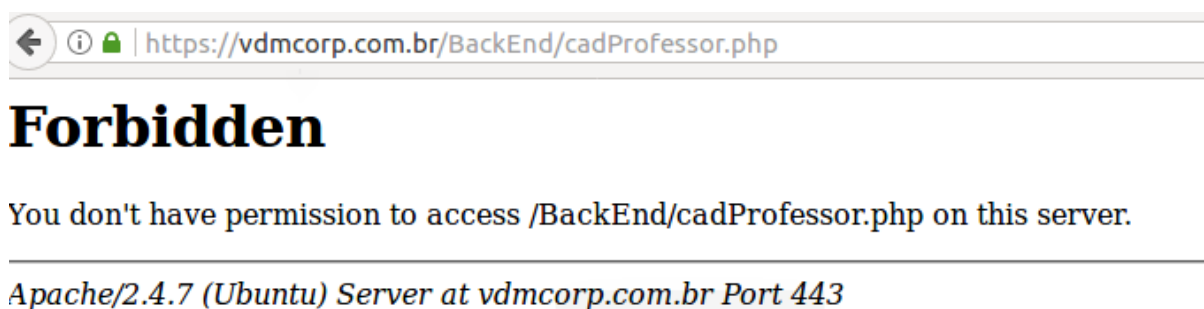
Logradouro: Rua 88

Número: 22

Complemento: asfdsf

Bairro: Setor Sul

Verificado que ao tentar inserir o script não foi possível salvar o cadastro:



### 1.3.2 COMO MITIGAR ESTA VULNERABILIDADE

É fundamental que você desativar o suporte TRACE HTTP em todos os servidores web. Um atacante pode roubar dados de cookies via Javascript mesmo quando document.cookie está desativado ou não é suportado no cliente. Este ataque é montado quando um usuário insere um script malicioso em entrada de dados, então, quando outro usuário acessa a página que contém este script, uma chamada de rastreamento HTTP assíncrona é acionada, que coleta as informações do cookie do usuário do servidor e depois envia para outro servidor malicioso que coleta a informação do cookie para que o invasor possa montar um ataque de sequestro de sessão. Isso é facilmente mitigado através da remoção de suporte para TRACE HTTP em todos os servidores web.

## **1.4 BROKEN ACCESS CONTROL**

Access control, às vezes chamado de “authorization”, é quando uma aplicação web acesso ao conteúdo e funções para alguns usuários e outros não. Essas verificações são executadas após a autenticação, e governar o que os usuários estão autorizados a fazer. Controle de acesso soa como um problema simples, mas é insidiosamente difícil de implementar corretamente. Modelo de controle de acesso de uma aplicação web está intimamente ligada ao conteúdo e funções que o site oferece. Além disso, os usuários podem cair em um número de grupos ou funções com diferentes habilidades ou privilégios.

Desenvolvedores frequentemente subestimam a dificuldade de implementar um mecanismo de controle de acesso confiável. Muitos destes sistemas não foram deliberadamente concebidos, mas tem simplesmente evoluído juntamente com a web site. Nestes casos, são inseridas as regras de controle de acesso em vários locais em todo o código. À medida que o site se aproxima da implantação, a coleção de regras torna-se tão difícil de controlar que é quase impossível entender.

Muitos destes sistemas de controle de acesso são falhos não são difíceis de explorar. Tudo o que é necessário é criar um pedido de funções ou conteúdo que não deve ser concedido. Uma vez que uma falha for descoberta, as consequências de um esquema de controle de acesso falho podem ser devastadoras. Além de exibir conteúdo não autorizado, um invasor pode ser capaz de alterar ou apagar o conteúdo, execute funções não autorizadas, ou mesmo assumir a administração do site.

Um tipo específico de problema de controle de acesso é interfaces administrativas que permitem que os administradores do site para gerenciar com eficiência os usuários, dados e conteúdo em seu site. Em muitos casos, os sites têm suporte a uma variedade de funções administrativas para permitir a granularidade de administração do site. Devido ao seu poder, essas interfaces são frequentemente alvos de ataques de ambos os outsiders e insiders.

### **1.4.1 EXPLORAÇÃO DA VULNERABILIDADE**

Verificado que é possível acessar um plano de ensino de outro professor a partir da url, alterando o código:



**Faculdade Senac Goiás**  
**Curso Superior de Tecnologia em Segurança da Informação**  
**Plano de Ensino**

<b>COMPONENTE CURRICULAR:</b>	Princípios de Segurança da Informação	
<b>PROFESSOR:</b>	Elias Batista Ferreira	
<b>MÓDULO: I</b>	<b>ANO: 2017 / SEMESTRE: 2</b>	<b>CARGA HORÁRIA: 58h</b>



**Faculdade Senac Goiás**  
**Curso Superior de Tecnologia em Gestão da Tecnologia da Informação**  
**Plano de Ensino**

<b>COMPONENTE CURRICULAR:</b>	Gerência de Redes de Computadores	
<b>PROFESSOR:</b>	Marissol Martins Barros	
<b>MÓDULO: IV</b>	<b>ANO: 2017 / SEMESTRE: 2</b>	<b>CARGA HORÁRIA: 36h</b>

## 1.4.2 COMO MITIGAR ESTA VULNERABILIDADE

O passo mais importante é pensar nos requisitos de controle de acesso de uma aplicação e capturá-la em uma política de segurança de aplicação da Web. Recomendado o uso de uma matriz de controle de acesso para definir as regras de controle de acesso. Sem documentar a política de segurança, não existe uma definição do que significa ser seguro para esse site. A política deve documentar quais tipos de usuários podem acessar o sistema e quais funções e conteúdo de cada um desses tipos de usuários devem ter permissão para acessar. O mecanismo de controle de acesso deve ser amplamente testado para garantir que não há como ignorá-lo. Este teste requer uma variedade de contas e extensas tentativas de acessar conteúdo ou funções não autorizadas.

## 1.5 SECURITY MISCONFIGURATION

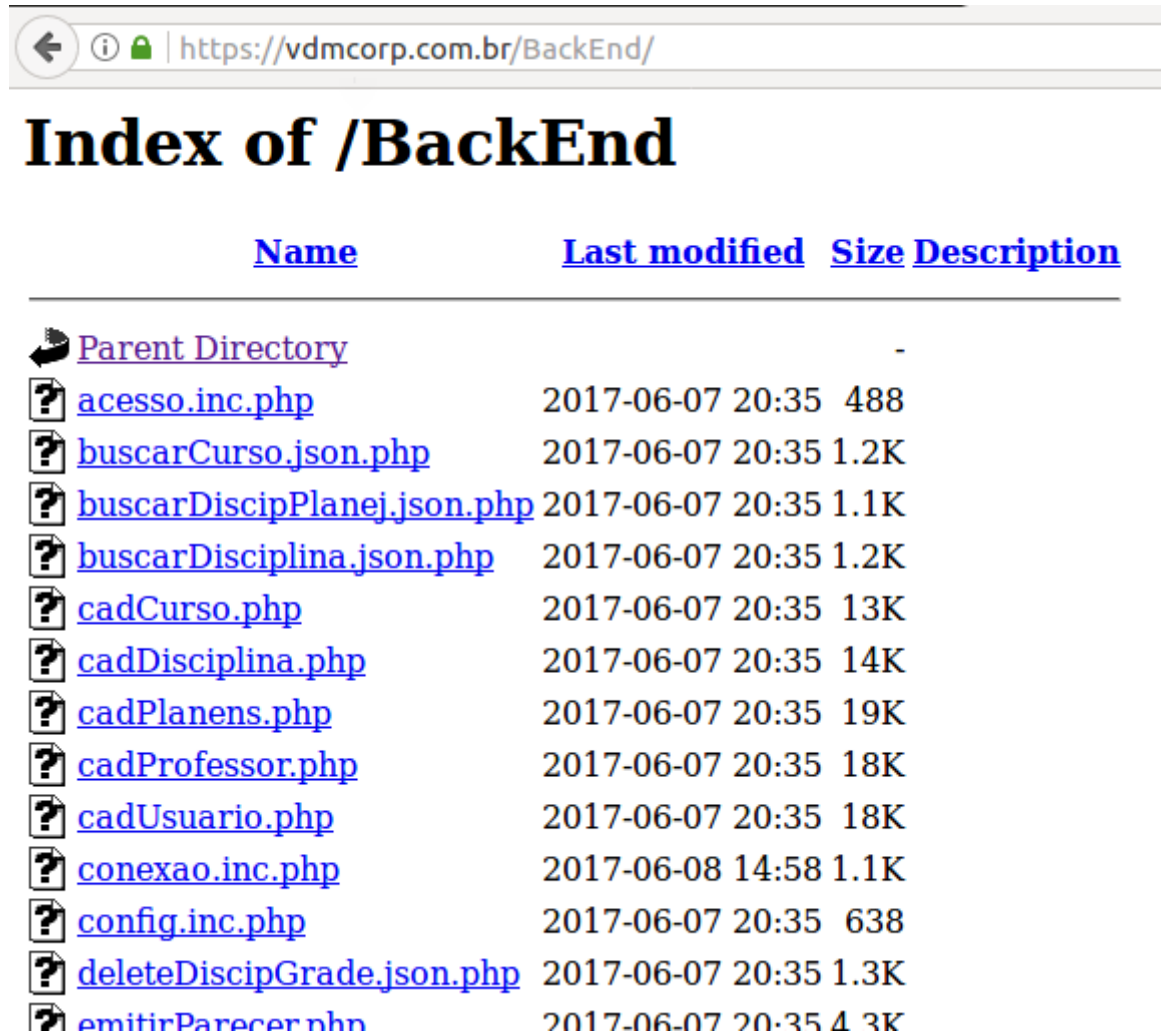
A má configuração de segurança é simplesmente essa - montagem incorreta das salvaguardas para uma aplicação web. Essas configurações erradas geralmente ocorrem















quando os furos são deixados na estrutura de segurança de uma aplicação por administradores de sistemas, DBAs e desenvolvedores. Eles podem ocorrer em qualquer nível da pilha da aplicação, incluindo a plataforma, servidor web, servidor de aplicação, banco de dados, estrutura e código personalizado. Essas configurações erradas de segurança podem levar um invasor diretamente ao sistema e resultar em um sistema parcial ou mesmo totalmente comprometido.

Os atacantes encontram essas configurações erradas através de acesso não autorizado a contas padrão, páginas da web não utilizadas, falhas não corrigidas, arquivos e diretórios desprotegidos e muito mais. Se um sistema for comprometido através de configurações de segurança defeituosas, os dados podem ser roubados ou modificados lentamente ao longo do tempo e podem ser demorados e dispendiosos para serem recuperados.

### 1.5.1 EXPLORAÇÃO DA VULNERABILIDADE

Verificado que na aplicação é possível verificar quais são os arquivos contidos nas pastas:



<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">acesso.inc.php</a>	2017-06-07 20:35	488	
 <a href="#">buscarCurso.json.php</a>	2017-06-07 20:35	1.2K	
 <a href="#">buscarDiscipPlanej.json.php</a>	2017-06-07 20:35	1.1K	
 <a href="#">buscarDisciplina.json.php</a>	2017-06-07 20:35	1.2K	
 <a href="#">cadCurso.php</a>	2017-06-07 20:35	13K	
 <a href="#">cadDisciplina.php</a>	2017-06-07 20:35	14K	
 <a href="#">cadPlanens.php</a>	2017-06-07 20:35	19K	
 <a href="#">cadProfessor.php</a>	2017-06-07 20:35	18K	
 <a href="#">cadUsuario.php</a>	2017-06-07 20:35	18K	
 <a href="#">conexao.inc.php</a>	2017-06-08 14:58	1.1K	
 <a href="#">config.inc.php</a>	2017-06-07 20:35	638	
 <a href="#">deleteDiscipGrade.json.php</a>	2017-06-07 20:35	1.3K	
 <a href="#">emitirParecer.php</a>	2017-06-07 20:35	4.3K	

## **1.5.2 COMO MITIGAR ESTA VULNERABILIDADE**

As configurações errôneas de segurança são fáceis de explorar, mas existem várias maneiras criativas de preveni-las, incluindo as seguintes recomendações de especialistas da indústria:

- Desenvolva um processo repetitivo para reduzir a superfície da vulnerabilidade
- Desativar contas padrão e alterar senhas
- Mantenha o software atualizado
- Desenvolva uma forte arquitetura de aplicativos que efetivamente isole componentes e criptografe dados que seja especialmente importante com dados confidenciais.
- Desabilite arquivos ou recursos desnecessários
- Não apresentar traçadores de pilha para usuários
- Certificar-se de que as configurações de segurança nas estruturas de desenvolvimento e as bibliotecas sejam definidas para garantir valores
- Executar ferramentas (por exemplo, scanners automatizados) e realize auditorias regulares para identificar furos na configuração de segurança.

## **1.6 BRUTE FORCE**

Um ataque de força bruta pode se manifestar de muitas maneiras diferentes, mas consiste principalmente em um invasor configurando valores predeterminados, fazendo solicitações para um servidor usando esses valores e depois analisando a resposta. Por uma questão de eficiência, um invasor pode usar um ataque de dicionário (com ou sem mutações) ou um ataque de força bruta tradicional (com determinadas classes de caracteres, por exemplo: alfanumérico, especial, caso (em) sensível). Considerando um determinado método, o número de tentativas, a eficiência do sistema que conduz o ataque e a eficiência estimada do sistema da vítima, o invasor seja capaz de calcular aproximadamente quanto tempo levará para enviar todos os valores predeterminados escolhidos.

### **1.6.1 COMO FOI EXPLORADO A VULNERABILIDADE**

A vulnerabilidade foi explorada utilizando a ferramenta Burp Suite:

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
84	fabricao	complexpassword	302	<input type="checkbox"/>	<input type="checkbox"/>	2752	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
1	fabricao	Spring2017	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
2	fabricao	Spring2016	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
3	fabricao	Spring2015	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
4	fabricao	Spring2014	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
5	fabricao	Spring2013	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	
6	fabricao	spring2017	200	<input type="checkbox"/>	<input type="checkbox"/>	2832	

Request Response

Raw Params Headers Hex

```

POST /BackEnd/login.php HTTP/1.1
Host: vdmcorp.com.br
User-Agent: Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Content-Type: application/x-www-form-urlencoded
Content-Length: 114
Referer: https://vdmcorp.com.br/
Cookie: PHPSESSID=3ar5hbjnd3jebqao8epu3ni965
Connection: close
Upgrade-Insecure-Requests: 1

:xtLogin=fabricao&passwdSenha=complexpassword&Cadastrar=Entrar&CSRF_TOKEN=dbfe60298d697cf174d325c03ce8f6b925cfd329

```

? < + > Type a search term 0 matches

## 1.6.2 COMO MITIGAR ESTA VULNERABILIDADE

Realizar configuração de limite de tempo de login, controle de tentativa máxima de login com o usuário.

## 2 CONCLUSÃO

Neste trabalho apresentamos as algumas das principais vulnerabilidades WEB, durante as explorações destas vulnerabilidades foi verificado que com pouco conhecimento técnico é possível explorar as vulnerabilidades com certa facilidade, demonstrando assim o grau de dificuldade das vulnerabilidades Web conhecidas e abordadas na OWASP Top 10.

## 3 REFERÊNCIAS BIBLIOGRÁFICAS

[https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project#tab=Main](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=Main)

[https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project#tab=OWASP\\_Top\\_10\\_for\\_2017\\_Release\\_Candidate](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=OWASP_Top_10_for_2017_Release_Candidate)

[https://www.owasp.org/index.php/SQL\\_Injection](https://www.owasp.org/index.php/SQL_Injection)

[https://www.owasp.org/index.php/Broken\\_Authentication\\_and\\_Session\\_Management](https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management)

<https://www.veracode.com/security/insufficient-transport-layer-protection>

[https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

<https://codedx.com/security-misconfiguration/>

[https://www.owasp.org/index.php/Broken\\_Access\\_Control](https://www.owasp.org/index.php/Broken_Access_Control)

[https://www.owasp.org/index.php/Brute\\_force\\_attack](https://www.owasp.org/index.php/Brute_force_attack)

<https://bounty.github.com/classifications/sensitive-data-exposure.html>