

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

1 VUNERABILIDADES

Neste relatório iremos identificar algumas vulnerabilidades existentes na aplicação web no site: <https://farmmachine.ipcalling.com.br>, e levantar evidencias destas vulnerabilidades e apresentação da proposta de mitigação das vulnerabilidades.

1.1 INJECTION

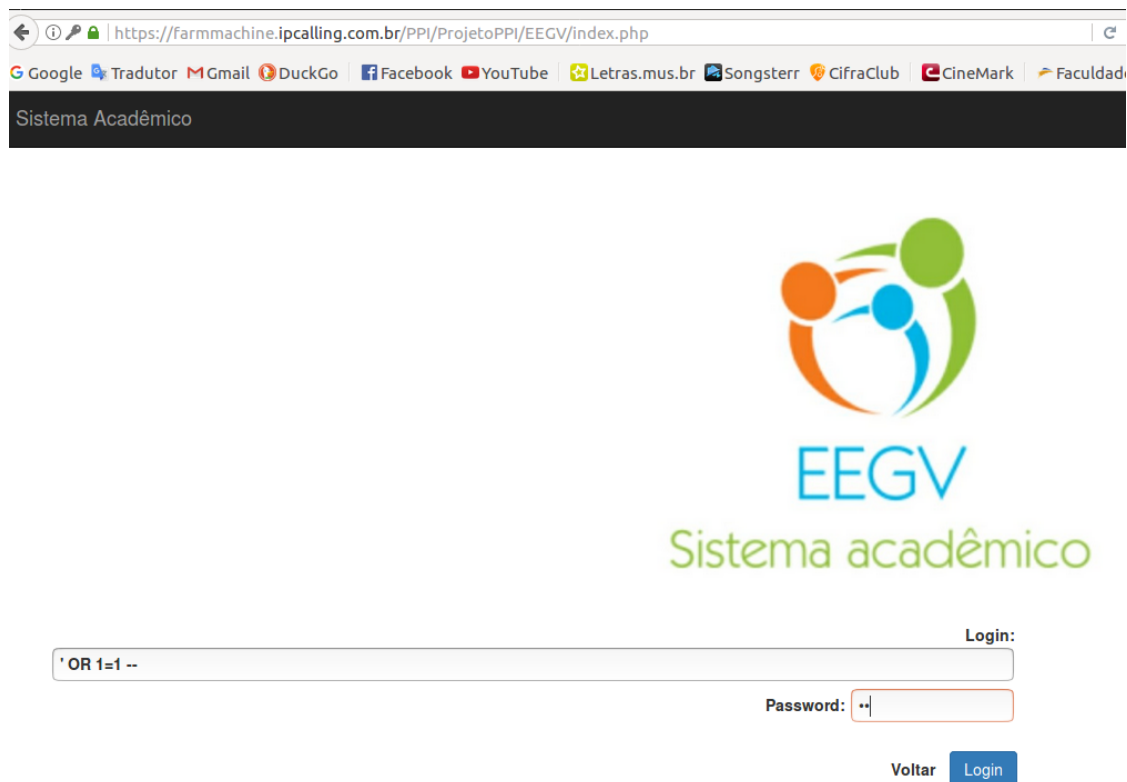
Um ataque de injeção SQL consiste em inserção ou "injeção" de uma consulta SQL através dos dados de entrada do cliente para a aplicação. Uma exploração de injeção SQL bem-sucedida pode ler dados confidenciais do banco de dados, modificar dados do banco de dados (Inserir / Atualizar / Excluir), executar 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://farmmachine.ipcalling.com.br> foi explorada a partir do da área de login do usuário:



Realizado o acesso a aplicação com um usuário aleatório.



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 validadas e tratadas a fim de impedir a execução de eventuais instruções destrutivas ou operações que não sejam as esperadas.

1.2 BROKEN AUTHENTICATION AND SESSION MANAGEMENT

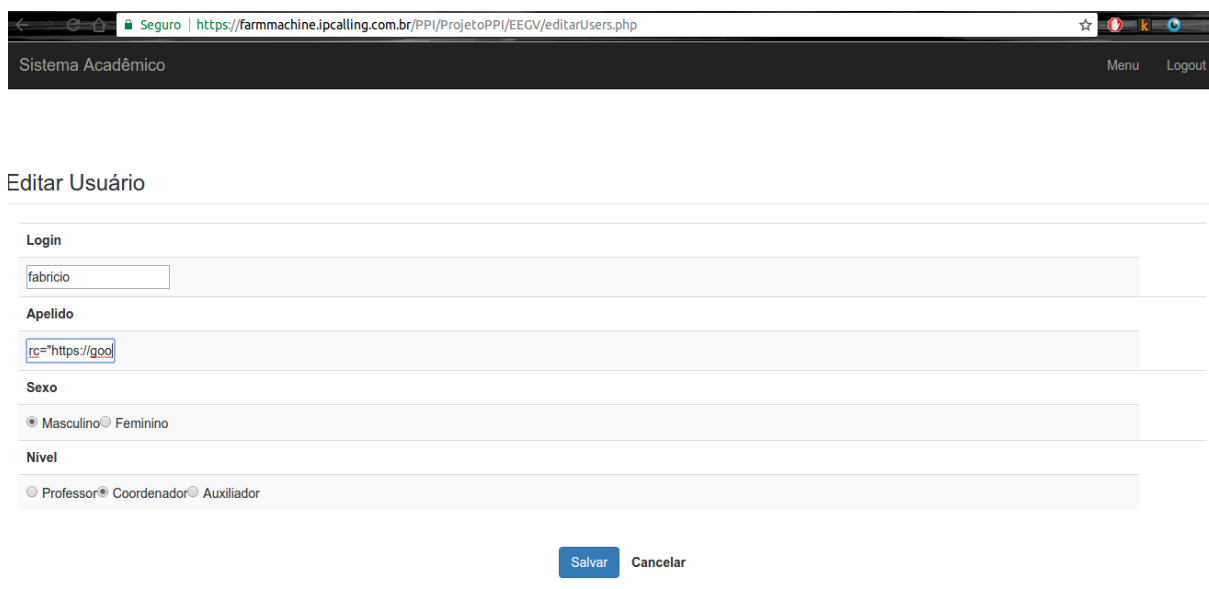
Autenticação e gerenciamento de sessão inclui todos os aspectos de lidar com a autenticação do usuário e gerenciamento de sessões ativas. A autenticação é um aspecto crítico deste processo, mas mecanismos de autenticação, mesmo sólidos pode ser prejudicada por funções de gerenciamento de credenciais 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

Para realizar esta exploração foi elaborado um script que gera um relatório com o IP, o cookie, código de sessão e horário do acesso a aplicação, que foi inserido no campo do usuário do usuário do usuário:



The screenshot shows a web browser window with the address bar displaying "Seguro | https://farmmachine.ipcalling.com.br/PPI/ProjetoPPI/EEGV/editarUsers.php". The page title is "Sistema Acadêmico" and there are "Menu" and "Logout" links in the top right. The main content area is titled "Editar Usuário" and contains a form with the following fields:

- Login:** A text input field containing the value "fabricao".
- Apelido:** A text input field containing the value "rc=https://goo".
- Sexo:** A radio button group with "Masculino" selected and "Feminino" unselected.
- Nível:** A radio button group with "Professor" selected, "Coordenador" unselected, and "Auxiliador" unselected.

At the bottom of the form are two buttons: "Salvar" (Save) and "Cancelar" (Cancel).

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:



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 SECURITY MISCONFIGURATION

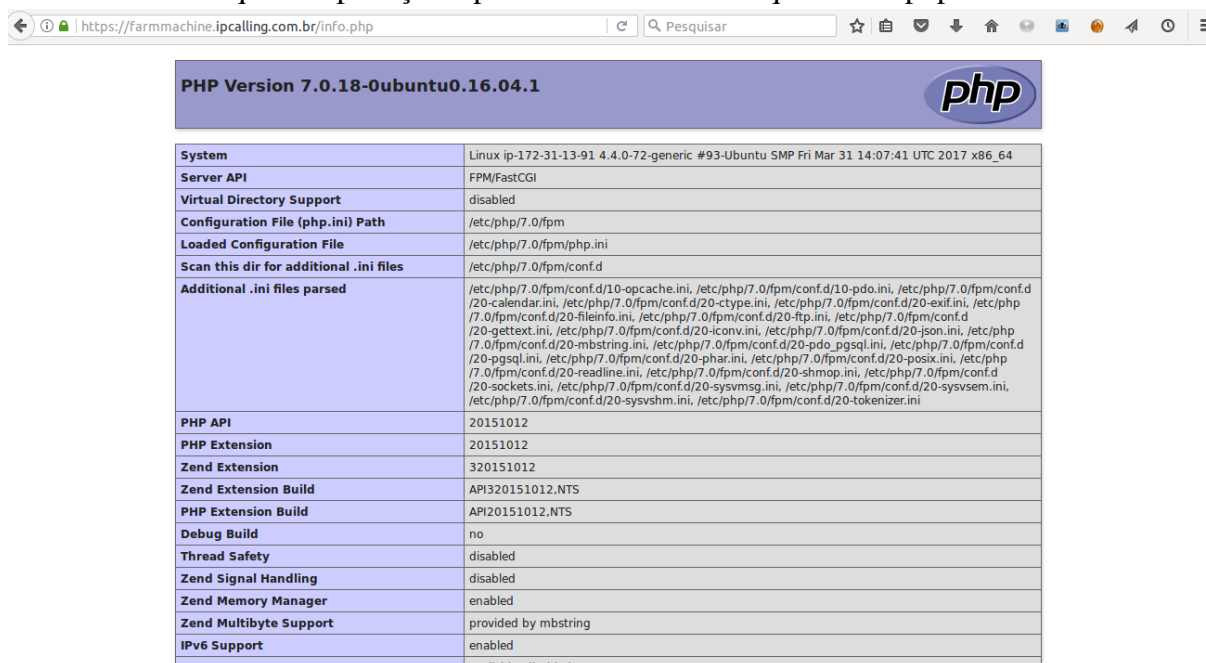
A má configuração de segurança é simplesmente essa - montagem incorreta das salvaguardas para uma aplicação web. Essas configurações erradas geralmente ocorrem quando os furos são deixados na estrutura de segurança de uma aplicação por administradores de sistemas, DBAs e desenvolvedores. Eles podem ocorrer 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 encontrar o arquivo info.php:



PHP Version 7.0.18-0ubuntu0.16.04.1	
System	Linux ip-172-31-13-91 4.4.0-72-generic #93-Ubuntu SMP Fri Mar 31 14:07:41 UTC 2017 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-ctype.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gd.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-imagick.ini, /etc/php/7.0/fpm/conf.d/20-imagick.ini, /etc/php/7.0/fpm/conf.d/20-ldap.ini, /etc/php/7.0/fpm/conf.d/20-mbstring.ini, /etc/php/7.0/fpm/conf.d/20-mcrypt.ini, /etc/php/7.0/fpm/conf.d/20-mysql.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-pdo_pgsql.ini, /etc/php/7.0/fpm/conf.d/20-pgsql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.0/fpm/conf.d/20-sysvsem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
NTS Support	available, disabled

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 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.

1.5.1 COMO FOI EXPLORADO A VULNERABILIDADE

A vulnerabilidade foi explorada com o auxílio da ferramenta Burp Suite:



1.5.2 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>