Plano de Ação para Correção de Vulnerabilidades Identificadas

---

Objetivo:

Este plano de ação visa corrigir as vulnerabilidades e melhorar a segurança do servidor web identificado durante a análise da máquina com IP 10.0.0.6 na porta 8080. O foco será na mitigação de riscos relacionados a cookies inseguros, exposição de informações sensíveis, e a proteção do servidor contra possíveis ataques.

---

1. Segurança de Cookies

Problema Identificado:

O cookie PHPSESSID não possui a flag HttpOnly configurada, o que possibilita que scripts no navegador acessem o valor do cookie, aumentando o risco de exploração por meio de ataques XSS (Cross-Site Scripting).

Ação Corretiva:

1. **Configuração de HttpOnly**:

   - [x] Corrigido
   - [ ] Não corrigido
   - Atualizar a configuração do servidor web para garantir que o cookie PHPSESSID tenha a flag `HttpOnly` configurada.
   - Se necessário, adicione o seguinte código no arquivo de configuração PHP (`php.ini`):

     ```
     session.cookie_httponly = 1
     ```

2. **Uso de Secure**:

- [x] Corrigido
- [ ] Não corrigido
- Configurar a flag `Secure` para garantir que os cookies sejam transmitidos apenas em conexões HTTPS.
- No arquivo `php.ini`, definir:

  ```
  session.cookie_secure = 1
  ```

Responsável: @Silas Marques

Prazo de Implementação: 3 dias

---

2. Proteção de Diretórios Sensíveis

- [x] Corrigido
- [ ] Não corrigido
- [ ] Corrigido parcialmente

Problema Identificado:

Existem diretórios e arquivos sensíveis expostos, como `/config/`, `/docs/`, e `/external/`, que podem conter informações privadas ou configurações do servidor.

Ação Corretiva:

1. **Restringir o Acesso aos Diretórios Sensíveis**:
   - Criar regras no Apache para restringir o acesso a esses diretórios, utilizando o arquivo `.htaccess`. Exemplo:

```
<Directory "/path/to/config">
    Order Deny,Allow
    Deny from all
</Directory>
```

- Assegurar que os diretórios `/config/`, `/docs/`, e `/external/` sejam protegidos adequadamente para evitar o acesso público.

2. **Configuração do robots.txt**:

- [x] Corrigido
- [ ] Não corrigido
- [ ] Corrigido parcialmente

- Modificar o arquivo `robots.txt` para bloquear os motores de busca de indexar diretórios sensíveis. Exemplo:

```
User-agent: *
Disallow: /config/
Disallow: /docs/
Disallow: /external/
```

Responsável: @Silas Marques

Prazo de Implementação: 2 dias

---

3. Acesso à Página de Login

- [ ] Corrigido
- [ ] Não corrigido
- [x] Corrigido parcialmente

Problema Identificado:

Foi identificado o acesso a uma página de login potencialmente vulnerável à ataques de força bruta e outros métodos de exploração.

Ação Corretiva:

1. **Implementar Limitação de Tentativas de Login**:
   - Implementar mecanismos para limitar o número de tentativas de login por endereço IP, evitando ataques de força bruta. Ferramentas como Fail2Ban podem ser configuradas para bloquear IPs que tentam acessar a página de login várias vezes de maneira maliciosa.
2. **Adicionar CAPTCHA**:
   - Incluir CAPTCHA (exemplo: Google reCAPTCHA) no formulário de login para impedir ataques automatizados.
3. **Autenticação Multifatorial (MFA)**:
   - Implementar autenticação multifatorial (MFA) para melhorar a segurança do processo de login.

Responsável: Equipe de Desenvolvimento / Administrador de Sistemas

Prazo de Implementação: 5 dias

---

4. Revisão e Hardening do Servidor

Problema Identificado:

O servidor Apache não está configurado adequadamente para proteger diretórios e serviços expostos, podendo ser explorado por invasores.O servidor web está vazando inodes através do cabeçalho ETag. Isso pode fornecer informações sobre a estrutura interna de arquivos do servidor, como identificadores exclusivos de arquivos no sistema de arquivos. Essa informação pode ser explorada por atacantes para mapear a estrutura do servidor, potencialmente expondo arquivos sensíveis.

Ação Corretiva:

1. **Revisão das Configurações do Apache**:
   - Reconfigurar o servidor Apache para desabilitar a listagem de diretórios nos locais onde não seja necessária. No arquivo de configuração do Apache (`httpd.conf` ou `apache2.conf`), adicionar a seguinte diretiva:

```
Options -Indexes
```

- Garantir que todas as configurações sensíveis no servidor Apache estejam protegidas e apenas acessíveis por administradores.

2. **Atualização do Servidor Apache**:

- Realizar a atualização para a versão mais recente do Apache, corrigindo possíveis vulnerabilidades de segurança. Utilize o gerenciador de pacotes do Debian para atualizar:

```
sudo apt update
sudo apt upgrade apache2
```

Responsável: @Alisson dos Santos Machado

Prazo de Implementação: 4 dias

3. **Desabilitar a Geração de ETags ou Modificar a Geração de ETags**: O servidor Apache está configurado para gerar cabeçalhos ETag que incluem inodes, o que pode vazar informações sobre a estrutura de arquivos internos. Para corrigir essa vulnerabilidade, é necessário modificar a configuração do Apache para desabilitar ou ajustar a geração de ETags.

Passos para a correção:

1. Abra o arquivo de configuração do Apache (`httpd.conf` ou `apache2.conf`).
2. Localize ou adicione a diretiva `FileETag` e defina-a para uma configuração que não inclua inodes. A configuração recomendada seria:

```
FileETag None
```

- Essa configuração desabilita completamente a geração de ETags no servidor, evitando o vazamento de informações sobre o inode, tamanho ou data de modificação dos arquivos.
- **Alternativa**: Caso você queira continuar gerando ETags mas sem incluir inodes, você pode modificar a configuração da seguinte maneira:

```
FileETag MTime Size
```

Essa configuração garante que o ETag será gerado apenas com base no tempo de modificação (MTime) e no tamanho do arquivo (Size), sem incluir o inode, o que ainda proporciona uma maneira de cache eficiente sem revelar detalhes do sistema de arquivos.

4. **Reiniciar o Servidor Apache**: Após realizar as alterações na configuração, reinicie o servidor Apache para aplicar as modificações:

```
sudo systemctl restart apache2
```

Responsável: @Fabricio Amaral

Prazo de Implementação: 1 dia

6. Correção de Serviços no Servidor

6.1 Serviço: OpenSSH 9.6p1 (Ubuntu Linux; protocolo 2.0)

• **Chaves de host:**

o ECDSA: e5:3b:47:59:b2:fa:9e:f7:43:2e:71:90:f8:d3:3b:8a

o ED25519: 5b:c2:c2:da:a9:6b:b3:74:39:a2:59:17:b0:2a:88:bd

• **Possíveis Riscos:**

o Se senhas fracas forem utilizadas, pode ser vulnerável a ataques de força bruta.

o Dependendo das configurações, pode ser suscetível a exploração de vulnerabilidades conhecidas.

• Recomendação:

o Utilizar autenticação por chave pública.

o Restringir acesso apenas a IPs autorizados via firewall

6.2 Mysql (Porta 3306)

Versão: MySQL 5.5.54-0+deb8u1-log

• Protocolo: 10

• Plugins de Autenticação: mysql_native_password

• Possíveis Riscos:

o MySQL 5.5.54 é uma versão desatualizada e pode conter vulnerabilidades conhecidas.

o Se acesso remoto estiver permitido, pode ser explorado por atacantes.

• Recomendação:

o Atualizar para uma versão suportada do MySQL.

o Restringir conexões remotas e usar autenticação forte.

6.3 Aplicação DVWA na porta 8080:

Enumeração de Diretórios (Gobuster)

Utilizando o Gobuster para escanear diretórios no servidor web, foram encontrados os seguintes recursos:

- Arquivos sensíveis e diretórios expostos:

o /about.php (Status: 200)

o /config (Status: 200)
o /docs (Status: 200)

o /external (Status: 200)
o /index.php (Status: 200)
o /instructions.php (Status: 200)
o /login.php (Status: 200)
o /logout.php (Status: 200)
o /php.ini (Status: 200)
o /phpinfo.php (Status: 200)
o /robots.txt (Status: 200)
o /security.php (Status: 200)
o /setup.php (Status: 200)
•Arquivos de configuração e segurança bloqueados:
o .htaccess, .htpasswd, .hta (Váriosformatos bloqueados com Status: 403)
o /server-status (Status: 403)
•Possíveis riscos:
o A exposição de /php.ini e /phpinfo.php pode revelar configurações sensíveis do servidor.
o A presença de arquivos .htaccess, .htpasswd e suas variantes pode indicar que regras de acesso e autenticação podem ser mal configuradas.
o O arquivo robots.txt pode indicar diretórios ocultos que podem ser explorados.
•Recomendações:
o Remover arquivos desnecessários ou restringir seu acesso.
o Verificar se /phpinfo.php e /php.ini estão disponíveis publicamente e removê-los caso não sejam necessários.
o Configurar permissões corretas para arquivos .htaccess e .ht

---

Conclusão:

A implementação dessas correções e melhorias ajudará a reduzir significativamente os riscos associados à segurança da máquina alvo. Além disso, garantirá que o servidor esteja mais protegido contra acessos não autorizados, ataques de força bruta, e outras ameaças. A equipe deve monitorar continuamente o servidor e aplicar as melhores práticas de segurança.Com essas alterações, a geração de ETags será corrigida, evitando o vazamento de informações sensíveis, como inodes, que poderiam ser exploradas por atacantes. A desativação ou modificação das ETags ajuda a proteger melhor a estrutura interna do sistema de arquivos do servidor, minimizando o risco de exposição acidental de informações importantes.