

Segurança Web

Fernanda e Katharine

CERT.Bahia

Agenda

1. Segurança da Informação

2. Aplicações:

2.1. Desenvolvimento Seguro de Software

2.2. Vulnerabilidades:

- Broken Authentication
- Open Redirect
- SSRF

2.3. Análise Estática de Código

3. Servidores:

3.1. Princípios

3.2. Hardening de Servidores Web

4. h2t

Segurança da Informação

Vertentes

- Ofensiva
- Defensiva

Segurança da Informação

Vertentes

- Ofensiva
- Defensiva → **Hardening**:
 - Proteção do sistema por meio da redução de suas possíveis vulnerabilidades
 - Aplica-se:
 - Sistema Operacional
 - Serviços de Rede
 - Servidor Web
 - Aplicação
 - (...)

Segurança da Informação

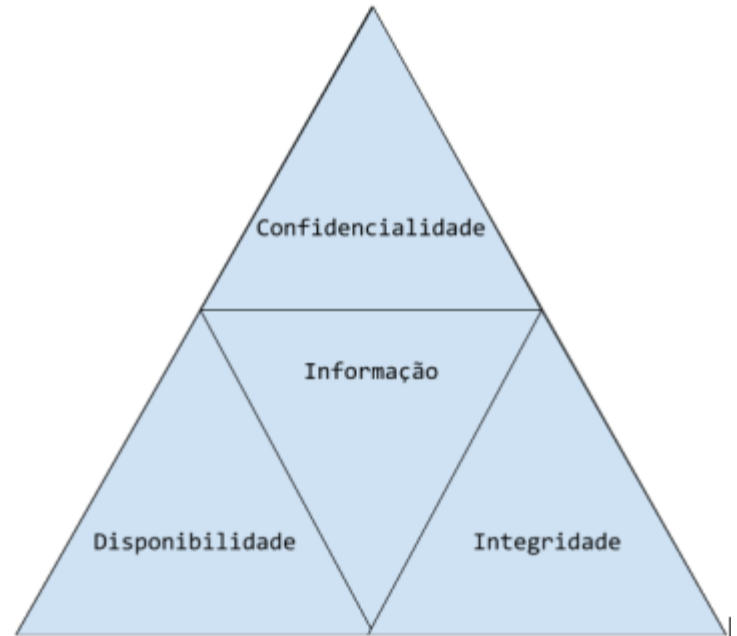
Vertentes

- Ofensiva
- Defensiva → **Hardening**:
 - Proteção do sistema por meio da redução de suas possíveis vulnerabilidades
 - Aplica-se:
 - Sistema Operacional
 - Serviços de Rede
 - **Servidor Web**
 - **Aplicação**
 - (...)

Segurança da Informação

Princípios

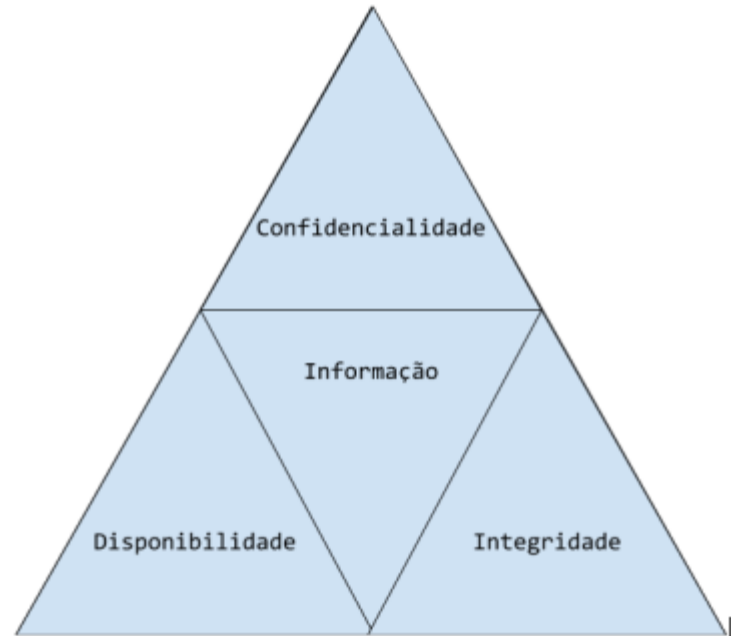
- Segurança **não é** "plug-and-play"
- Pilares da segurança da informação
- Desenvolvimento seguro de software



Segurança da Informação

Princípios

- Segurança **não é** "plug-and-play"
- Pilares da segurança da informação
- **Desenvolvimento seguro de software**



Desenvolvimento Seguro de Software

Ciclo de Desenvolvimento

1. Fase de Requisitos
 - Requisitos Relacionados a Segurança
2. Fase de Projeto
 - Modelagem de Ameaças
3. Fase de Codificação
 - Análise Estática de Código
 - Hardening
4. Fase de Testes
 - Pentest
5. Fase de Implementação
 - Hardening
 - Monitoramento de Incidente

Desenvolvimento Seguro de Software

Ciclo de Desenvolvimento

1. Fase de Requisitos
 - Requisitos Relacionados a Segurança
2. Fase de Projeto
 - Modelagem de Ameaças
3. Fase de Codificação
 - **Análise Estática de Código**
 - **Hardening**
4. Fase de Testes
 - Pentest
5. Fase de Implementação
 - **Hardening**
 - Monitoramento de Incidente

Aplicações

Vulnerabilidades

- Autenticação
- Gerenciamento de sessão
- Interação com o banco de dados
- Validação de entrada genérica
- Lógica do aplicativo

+-----+	
	OWASP Top 10 - 2017
+-----+	
	A1: Injection
+-----+	
	A2: Broken Authentication
+-----+	
	A3: Sensitive Data Exposure
+-----+	
	A4: XML External Entities (XXE)
+-----+	
	A5: Broken Access Control
+-----+	
	A6: Security Misconfiguration
+-----+	
	A7: Cross-Site Scripting (XSS)
+-----+	
	A8: Insecure Deserialization
+-----+	
	A9: Using Components With Known Vulnerabilities
+-----+	
	A10: Insufficient Logging & Monitoring
+-----+	

Vulnerabilidades

- Autenticação
- Gerenciamento de sessão
- Interação com o banco de dados
- Validação de entrada genérica
- Lógica do aplicativo

+-----+	
	OWASP Top 10 - 2017
+-----+	
	A1: Injection
+-----+	
	A2: Broken Authentication
+-----+	
	A3: Sensitive Data Exposure
+-----+	
	A4: XML External Entities (XXE)
+-----+	
	A5: Broken Access Control
+-----+	
	A6: Security Misconfiguration
+-----+	
	A7: Cross-Site Scripting (XSS)
+-----+	
	A8: Insecure Deserialization
+-----+	
	A9: Using Components With Known Vulnerabilities
+-----+	
	A10: Insufficient Logging & Monitoring
+-----+	

Broken Authentication

... and Session Management

- Permite que o atacante possa capturar ou bypassar métodos de autenticação utilizados por uma aplicação web vulnerável
- Aplicação **vulnerável**: implementação precária de funções relacionadas com a autenticação e administração da sessão
- Inclui **todos** os aspetos de lidar com a autenticação:
 - Credenciais não protegidas no armazenamento
 - Credenciais de login previsíveis
 - IDs de Sessão expostos na URL
 - Senhas, IDs de sessão enviados em conexões não criptografadas
 - (...)

Broken Authentication

Impacto

- Atacantes podem fazer um ataque de força bruta
 - Baseados em dicionário
 - Credential stuffing
- Atacantes podem gerar sessions IDs se elas não forem geradas randômicamente
- Um ataque bem sucedido pode comprometer todo o sistema

Broken Authentication

Mitigando

Não expor ID de Sessão na URL

- URL rewriting: **X**

```
http://example.com/sale/saleitems;jsessionid=2P00C2JSNDLPSKHCJUN2JV?dest=Hawaii
```

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização

- Password Strenght ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização

- Password Strenght ✓
- Password Use ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização

- Password Strenght ✓
- Password Use ✓
- Password Change Control ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT

- Criptografia Caseira ✗

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT

- Criptografia Caseira ✗
- MD5 ✗

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT

- Criptografia Caseira ✗
- MD5 ✗
- SHA-1 ✗

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT

- Criptografia Caseira ✗
- MD5 ✗
- SHA-1 ✗
- SHA-2 ✓ (SHA-256,SHA-384,SHA-512...)

Broken Authentication

Mitigando

- Não expor ID de Sessão na URL ✓
- Definir política de senhas da organização ✓
- Aplicar hash de encriptação com SALT ✓

Broken Authentication

Mitigando

- Não expor ID de Sessão na URL ✓
- Definir política de senhas da organização ✓
- Aplicar hash de encriptação com SALT ✓
- Colocar um timeout na sessão ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT ✓

Colocar um timeout na sessão ✓

Canal criptografado para transações - SSL/TLS ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT ✓

Colocar um timeout na sessão ✓

Canal criptografado para transações - SSL/TLS ✓

Implementar autenticação multi-fator ✓

Broken Authentication

Mitigando

Não expor ID de Sessão na URL ✓

Definir política de senhas da organização ✓

Aplicar hash de encriptação com SALT ✓

Colocar um timeout na sessão ✓

Canal criptografado para transações - SSL/TLS ✓

Implementar autenticação multi-fator ✓

Proteção contra ataques força-bruta -ex: Fail2Ban ✓

Unvalidated Redirects and Forwards

Open Redirect

- Parâmetros em uma requisição GET não são validados e permitem inserção de informações que redirecionam o usuário para um novo website
- Baseados em Header

```
https://www.example.com/login.html?RelayState=http%3A%2F%2FEvilWebsite.com
```

- Baseados em Javascript
 - Talvez não funcione para funções do lado do servidor, mas funcione no browser do usuário
 - Vetor de XSS

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing



Acesse: <https://catalogodefraudes.rnp.br/> e saiba mais!

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing
- Oauth

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing
- Oauth

```
facebook.com/oauth.php?clientid=123&state=abc&redirect_url=https://yourdomain.com/  
red.php?url%3dhttps://attacker.com/
```

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing
- Oauth
- SSRF

Unvalidated Redirects and Forwards

Open Redirect

- Muito utilizado para ataques de phishing
- Oauth
- **SSRF**

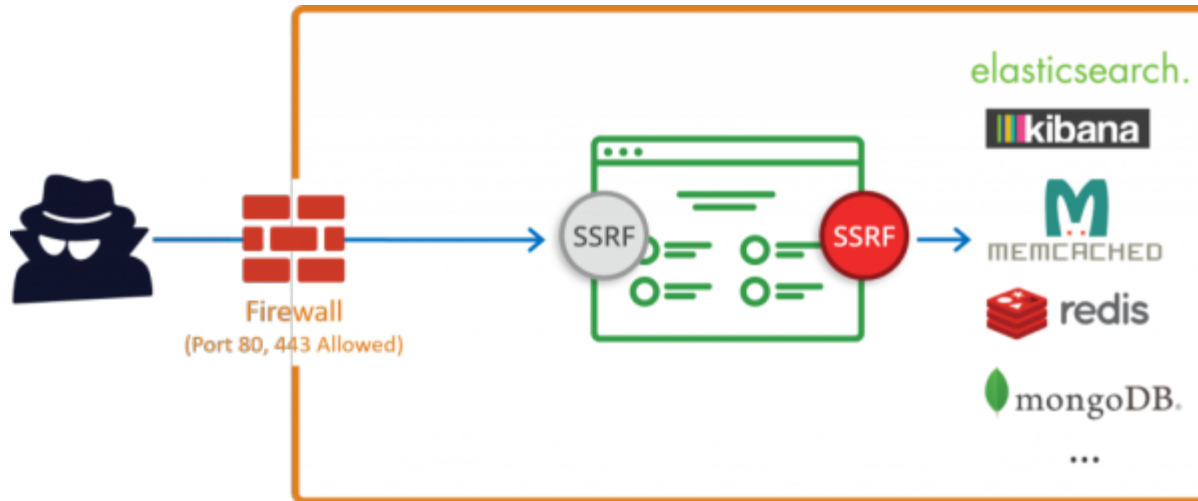
SSRF

Server Side Request Forgery

- Atacante pede ao servidor web vulnerável para buscar uma URL por ele
- Servidor **vulnerável**: o atacante tem controle parcial ou total das requisições enviadas pela aplicação web
- Tipos
 - Blind
 - Basic

SSRF

Server Side Request Forgery



Exemplo de aplicação vulnerável:

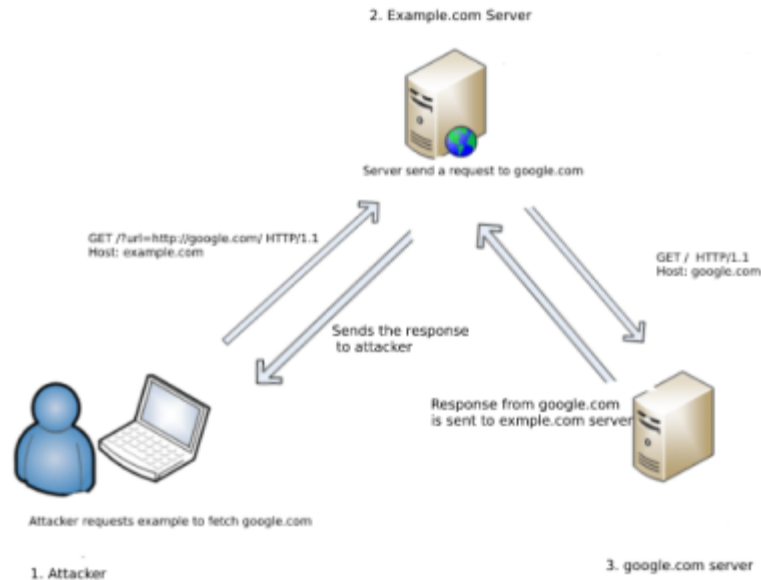
```
if (isset($_GET['url'])) {  
    $url = $_GET['url'];  
}
```

SSRF

Impacto

Atacante pode fazer requests arbitrários para qualquer website na Internet

```
GET /?url=http://google.com/ HTTP/1.1  
Host: example.com
```



SSRF

Impacto

Atacante consegue acessar serviços e arquivos internos do web server que não estão expostos para a internet

```
GET /?url=file:///etc/passwd HTTP/1.1  
Host: example.com
```

E mais...

- Conseguir metadata relacionado a Cloud Instances (ex: keys)
- Scans de redes e portas nos IPs internos
- Remote Code Execution (RCE)
- (...)

SSRF

Blind

Servidor manda somente a resposta, sem o conteúdo

URL parameter	Response HTTP status	RTT	Conclusion
http://127.0.0.1:22	200	10ms	Port is open
http://127.0.0.1:23	500	10ms	Port is closed
http://10.0.0.1/	500	30010ms	Firewalled or unable to route traffic to server
http://10.0.0.1:8080/	500	10ms	Port is closed and traffic is routed to server

SSRF

Mitigando

Badlists e expressões regulares

SSRF

Mitigando

Badlists e expressões regulares 

- HTTP Redirect
- Wildcard DNS
- Alternate IP encoding

SSRF

Mitigando

Badlists e expressões regulares 

Goodlist e resolução DNS 

SSRF

Mitigando

Badlists e expressões regulares ❌

Goodlist e resolução DNS ✔️

Atenção com as respostas do servidor ✔️

SSRF

Mitigando

Badlists e expressões regulares ❌

Goodlist e resolução DNS ✔️

Atenção com as respostas do servidor ✔️

Desabilite schemas URLs não utilizados ✔️

SSRF

Mitigando

Badlists e expressões regulares ❌

Goodlist e resolução DNS ✔️

Atenção com as respostas do servidor ✔️

Desabilite schemas URLs não utilizados ✔️

Autentique nos serviços internos ✔️

Análise Estática de Código

- Verifica a qualidade do código sem executá-lo
- Ferramentas:
 - Bandit (Python)
 - Coverity (C, C++, C#, Objective-C, Java...)
 - Brakeman (Ruby on Rails)

Análise Estática de Código

- Verifica a qualidade do código sem executá-lo
- Ferramentas:
 - Bandit (Python)
 - Coverity (C, C++, C#, Objective-C, Java...)
 - **Brakeman (Ruby on Rails)** → Railsgoat

```
== Brakeman Report ==  
  
Application Path: /media/nanda22k/DATA/STI/Rails_Girls/railsgoat  
Rails Version: 5.1.7  
Brakeman Version: 4.6.1  
Scan Date: 2019-09-16 22:21:13 -0300  
Duration: 0.573743686 seconds
```

Análise Estática de Código

- Verifica a qualidade do código sem executá-lo
- Ferramentas:
 - Bandit (Python)
 - Coverity (C, C++, C#, Objective-C, Java...)
 - **Brakeman (Ruby on Rails)** → Railsgoat

```
== Overview ==  
Controllers: 17  
Models: 12  
Templates: 27  
Errors: 0  
Security Warnings: 17
```

```
== Warning Types ==  
Command Injection: 1  
Cross-Site Request Forgery: 1  
Cross-Site Scripting: 1  
Dangerous Send: 1  
File Access: 1  
Format Validation: 1  
Mass Assignment: 2  
Remote Code Execution: 4  
SQL Injection: 2  
Session Setting: 3
```

Análise Estática de Código

- Verifica a qualidade do código sem executá-lo
- Ferramentas:
 - Bandit (Python)
 - Coverity (C, C++, C#, Objective-C, Java...)
 - **Brakeman (Ruby on Rails)** → Railsgoat

```
== Warnings ==  
  
Confidence: High  
Category: Cross-Site Request Forgery  
Check: ForgerySetting  
Message: `protect_from_forgery` should be called in `ApplicationController`  
File: app/controllers/application_controller.rb  
Line: 2  
  
Confidence: High  
Category: Cross-Site Scripting  
Check: CrossSiteScripting  
Message: Unescaped cookie value  
Code: cookies[:font]  
File: app/views/layouts/application.html.erb  
Line: 12
```

Servidores

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Defesa em profundidade ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Defesa em profundidade ✓

Desinstale os serviços desnecessários ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Defesa em profundidade ✓

Desinstale os serviços desnecessários ✓

Princípio do menor privilégio → trust policy ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Defesa em profundidade ✓

Desinstale os serviços desnecessários ✓

Princípio do menor privilégio → trust policy ✓

A configuração padrão **nunca** é a mais segura ✓

Princípios de Segurança de Servidores

Conheça seu sistema ✓

Conheça seu inimigo ✓

Defesa em profundidade ✓

Desinstale os serviços desnecessários ✓

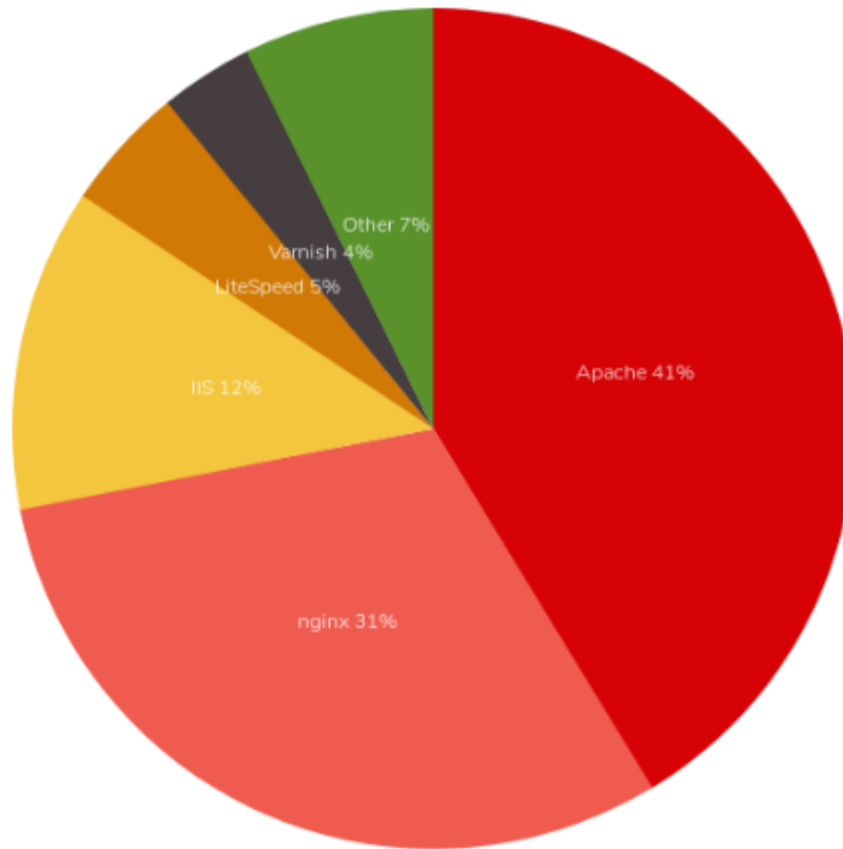
Princípio do menor privilégio → trust policy ✓

A configuração padrão **nunca** é a mais segura ✓

Proteção é importante, mas detecção é essencial ✓

Servidores Web

Distribution for websites using Web Server technologies



Fonte: <https://trends.builtwith.com/web-server>

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

▼ Response Headers [view source](#)

Accept-Ranges: bytes
Connection: Keep-Alive
Content-Length: 4897
Content-Type: text/html; charset=UTF-8
Date: Sun, 18 Feb 2018 07:01:37 GMT
ETag: "1321-5058a1e728280"
Keep-Alive: timeout=5, max=95
Last-Modified: Thu, 16 Oct 2014 13:20:58 GMT
Server: Apache/2.4.6 (CentOS)

```
# vim $Web_Server/httpd.conf
```

```
ServerTokens Prod  
ServerSignature Off
```

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory	13-Oct-2006 22:24	-	
 activity-icon.gif	01-Aug-2002 13:52	1k	
 adesso-keyboard.jpg	19-Jan-2006 09:40	10k	
 amazon-uk-dvd-rental..>	21-Aug-2006 00:39	27k	
 american_red_cross.gif	31-Aug-2005 17:15	5k	

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

```
# vim $Web_Server/httpd.conf
```

```
<Directory /opt/apache/htdocs>  
Options None  
</Directory>
```

Forbidden

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

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Etag ✓

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Etag ✓

```
# vim $Web_Server/httpd.conf
```

```
FileETag None
```

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Etag ✓

Execute o Apache de uma conta não privilegiada ✓

Hardening de Servidores Web

Remova a versão do servidor do Banner ✓

Desative a listagem dos diretórios em browser ✓

Etag ✓

Execute o Apache de uma conta não privilegiada ✓

```
# groupadd apache  
# useradd -G apache apache  
# chown -R apache:apache /opt/apache
```

```
# vim $Web_Server/httpd.conf
```

```
User apache  
Group apache
```

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

```
# cd $Web_Server/  
# chmod -R 750 bin conf
```

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

Proteja as configurações do sistema ✓

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

Proteja as configurações do sistema ✓

```
# vim $Web_Server/httpd.conf
```

```
<Directory />  
    AllowOverride None  
</Directory>
```

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

Proteja as configurações do sistema ✓

Mantenha apenas os métodos HTTP que vá realmente usar ✓

Hardening de Servidores Web

Mude a permissão do binário e do diretório de configuração ✓

Proteja as configurações do sistema ✓

Mantenha apenas os métodos HTTP que vá realmente usar ✓

```
# vim $Web_Server/httpd.conf
```

```
<LimitExcept GET POST HEAD>  
    deny from all  
</LimitExcept>
```

Hardening de Servidores Web

MITM → Use SSL ✓

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

```
# openssl req -out name.csr -newkey rsa:2048 -nodes -keyout name.key
```

```
SSLCipherSuite HIGH:!MEDIUM:!aNULL:!MD5:!RC4
```

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

DoS → Configure o valor de Timeout ✓

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

DoS → Configure o valor de Timeout ✓

Timeout 60

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

DoS → Configure o valor de Timeout ✓

XSS → Configure cookie com as flags HttpOnly e Secure ✓

Hardening de Servidores Web

MITM → Use SSL ✓

Break SSL Key → Mude para 2048 bit key e habilite cifras SSL fortes ✓

DoS → Configure o valor de Timeout ✓

XSS → Configure cookie com as flags HttpOnly e Secure ✓

```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
```



HTTP Hardening Tool

Verifica os cabeçalhos HTTP da aplicação web e sugere melhorias de configuração - adicionar ou retirar cabeçalhos

```
(venv) [gildasio@qwerty h2t]$ ./h2t.py scan hack.me
|      /\      |
| /\      ./~\  |
|      |/_     |

https://github.com/gildasio/h2t

Output explanation:
[+] Good headers. Already used in your website. Good job!
[+] Good headers. We recommend apply it
[-] Bad headers. We recommend remove it

http://hack.me
[-] Server
[-] Cookie not HTTP Only
[-] Cookie not over SSL/TLS
[-] Cookie not only from SameSite
[+] Referrer-Policy
[+] X-Content-Type-Options
```

Referências

Desenvolvimento Seguro e Hardening

- Curso da ESR: Hardening em Linux (SEG10) - Instrutor: Italo Valcy ♥
- Guia de Desenvolvimento Seguro da STI - Autor: Kaio Porto
- Hackers Expostos 7 - Segredos e Soluções para a Segurança de Redes
- Hacking Exposed 3 - Web Applications

h2t

- [Cert Bahia Projects: h2t](#) - Desenvolvedor: Gildásio Júnior

Informações Sobre Phishing

- [Catálogo de Fraudes da RNP](#)

Referências

SSRF

- [Acutenix Server-Side-Request-Forgery Vulnerability](#)
- [Medium SSRF-Server-Side-Request-Forgery-Types-And-Ways-To-Exploit-It](#)

Broken Authentication

- [Owasp Broken Authentication and Session Management](#)
- [Hdivsecurity Owasp Broken Authentication and Session Management](#)
- [Sitelock Owasp-top-10-broken-authentication-session-management](#)
- [Resources InfosecInstitute](#)

Open Redirect

- [Detectify The Real Impact of An Open Redirect](#)
- [Trustwave Understanding and Discovering Open Redirect Vulnerabilities](#)

Referências

Algoritmos de Criptografia

- [Blog Avira MD5 The Broken Algorithm](#)
- [Zdnet SHA-1 Collision Attacks are Now Actually Practical](#)
- [The SSL Store Difference SHA-1 SHA-2 SHA-256 Hash Algorithms](#)

Hardening de Servidor Web

- [Apachecon](#)
- [Deadliest WebAttacks](#)
- [Acodez Web-Server-Hardening-Checklist](#)
- [Geekflare Apache-Web-Server-Hardening-Security](#)
- [Mythemeshop Understanding Server Security](#)

Agradecimentos



fbarros@ufba.br

katharine.schramm@ufba.br