MJ – Departamento de Polícia Federal
Coordenação de Tecnologia da Informação – CTI

Pen-test de Aplicações Web: Técnicas e Ferramentas

Ivo de Carvalho Peixinho
Perito Criminal Federal

Agenda

1. Introdução
2. Ferramentas
3. Metodologia
4. Conclusões

Introdução

- Aplicações *Web*
  - Desenvolvidas em cima do protocolo HTTP e dos *Web Servers.*
  - Utilizam diversas linguagens e tecnologias (PHP, ASP, .NET, J2EE, Applets Java, ActiveX, CSS, DOM, SOAP, XML, C#, AJAX, SQL, RMDBS, Perl, Python, Ruby, Cookies, HTML, Javascript, Flash, ISAPI, WebDAV, CGI, ColdFusion, etc).
  - Crescimento exponencial das aplicações
  - **Crescimento dos ataques**

Introdução

- Segurança de Aplicações *Web*
  - SSL/TLS como panacéia de segurança
  - Protocolo HTTP inseguro
    - Inadequado para aplicações web (*stateless*)
  - Clientes (*Browsers*) inseguros
    - Usuário pode manipular dados
    - Usuário pode enviar dados arbitrários
  - Proteção de perímetro insuficiente
    - Filtros na porta 80/443?

Introdução

- Análise de Vulnerabilidades
  - Pode ser automatizado por ferramentas
  - Bons para aplicações *off the shelf* (Webmail, blogs, etc).
  - Podem não encontrar todas as vulnerabilidades da aplicação (variações)
  - Podem gerar falso-positivos
- Pen-Testing
  - Requer conhecimento e tempo por parte do analista
  - Capaz de encontrar vulnerabilidades mais complexas
  - Comprovação das vulnerabilidades (sem falso-positivos)

Ferramentas




- Análise de Vulnerabilidades
  - W3AF - http://w3af.sourceforge.net/
  - Nikto - http://cirt.net/nikto2
  - Nessus - http://www.nessus.org/
  - Httprint - http://www.net-square.com/httprint/
- Pen-Test
  - Burp Suite - http://www.portswigger.net/suite/
  - WebScarab - http://www.owasp.org/
  - JAD - http://www.varaneckas.com/jad
  - Paros proxy - http://www.parosproxy.org/
  - WebGoat - http://www.owasp.org/

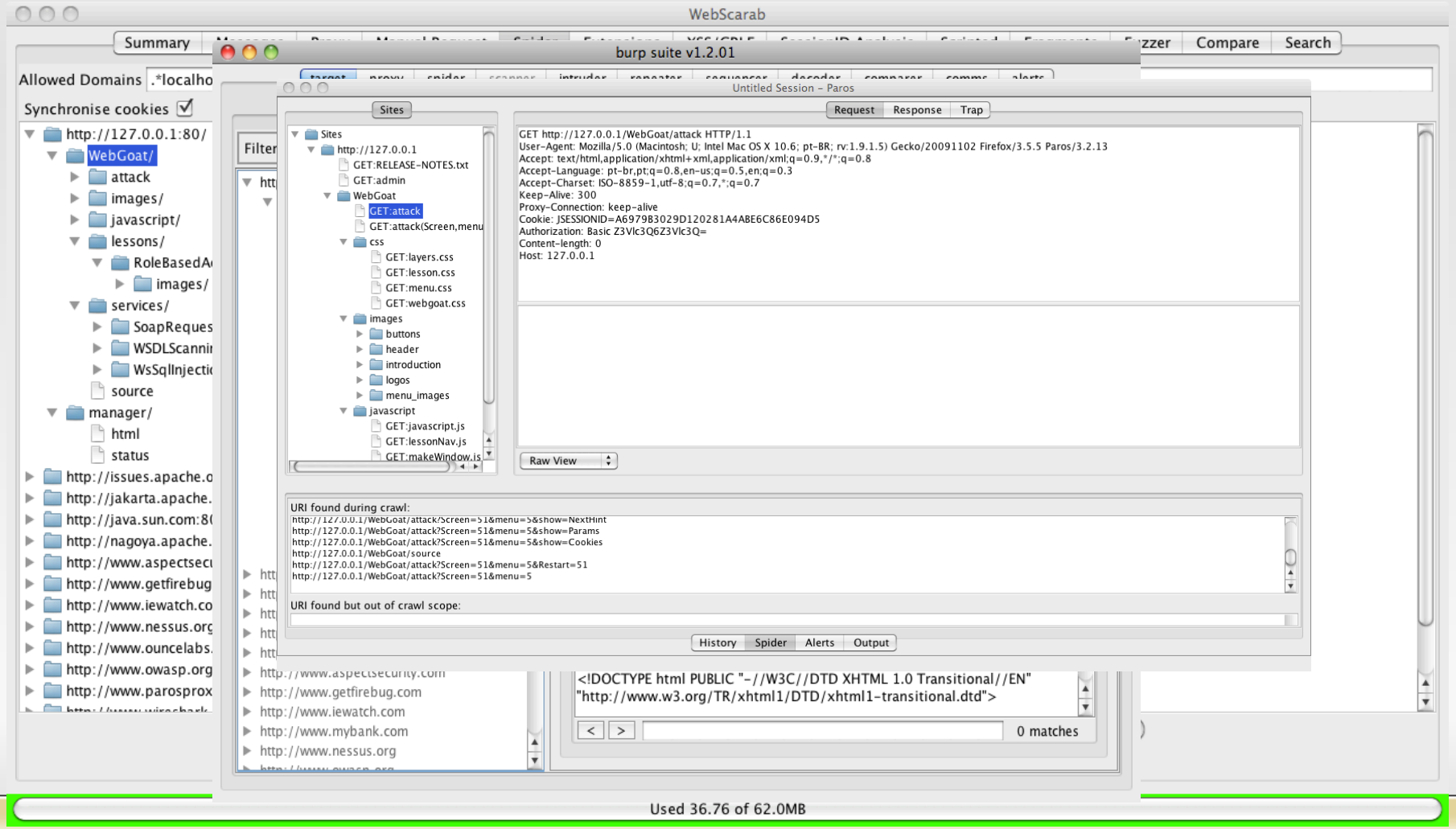
Metodologia

1. Mapear o conteúdo da aplicação
2. Analisar a aplicação e o servidor
3. Testar controles no lado do cliente
4. Testar mecanismos de autenticação
5. Testar mecanismos de gerenciamento de sessão
6. Testar controles de acesso
7. Testar vulnerabilidades nos parâmetros de entrada
8. Testar vulnerabilidades do Web Server

Metodologia

- Mapear o conteúdo da aplicação
  - Web Spidering
    - Varrer links da aplicação
    - Montar “mapa da aplicação”
    - Ferramentas automáticas
      - Burp Suite
      - WebScarab
      - Paros Proxy

Web Spidering

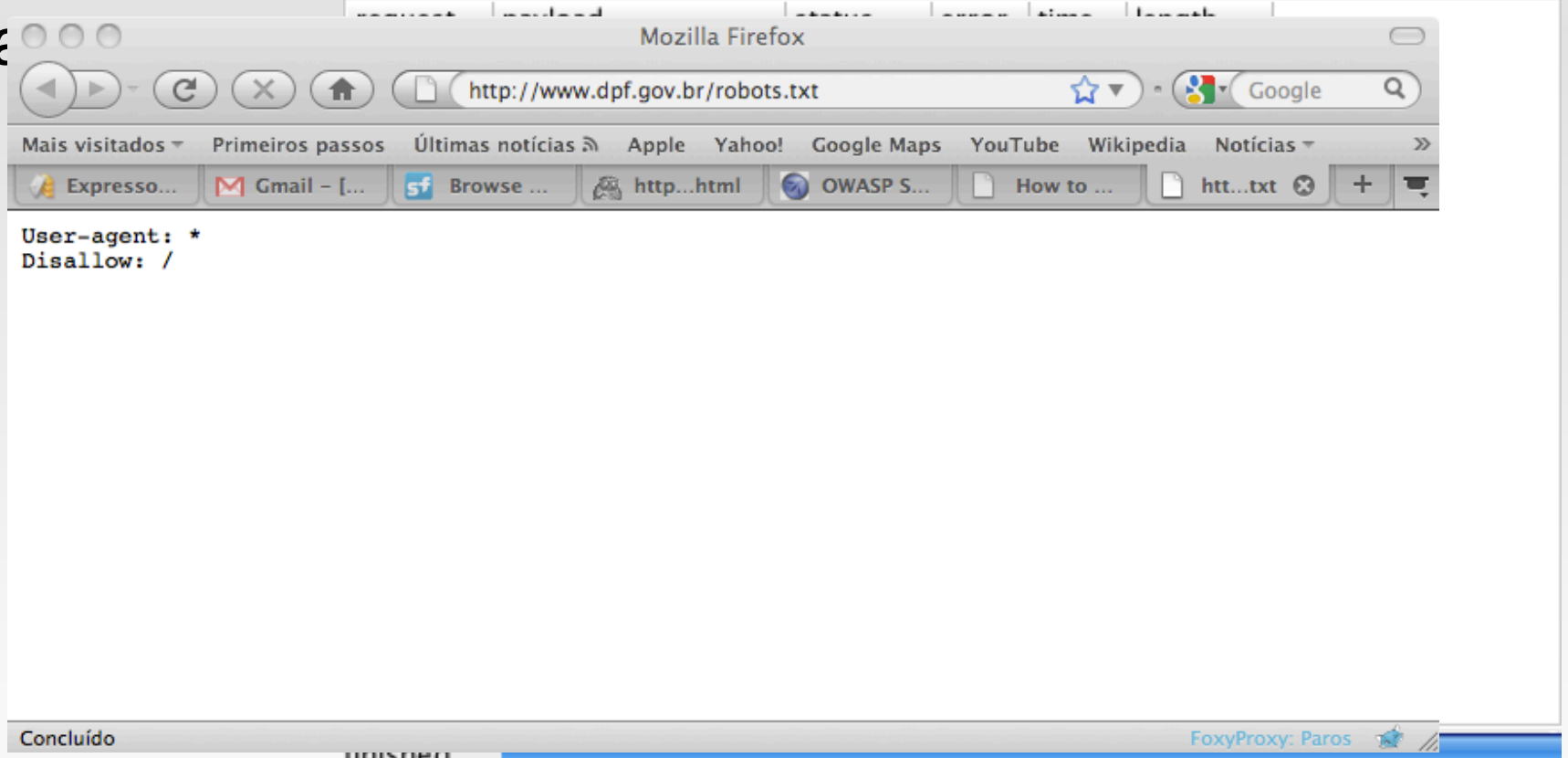


Metodologia

- Mapear o conteúdo da aplicação
  - Web Spidering automático
    - Desvantgens
      - Métodos de navegação não usuais
      - Funções com validação de dados
      - Gerenciamento de sessões
  - Spidering manual
    - Ferramentas de interceptação / proxy
    - Acesso manual à aplicação

Metodologia

- Ma
  - intruder attack 1



Metodologia

- Analisar a aplicação
  - Pontos de entrada de dados
    - URL's
    - Parametros de formulários (POST)
    - Query Strings
    - Cookies
    - Encapsulamento de dados
    - Parâmetros no caminho da URL
  - Verificar se a aplicação responde diferente para novos parâmetros

Metodologia

- A

Object reference not set to an instance of an object.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.NullReferenceException: Object reference not set to an instance of an object.

Source Error:

```
An unhandled exception was generated during the execution of the current web request.
Information regarding the origin and location of the exception can be identified using the
exception stack trace below.
```

Stack Trace:

```
[NullReferenceException: Object reference not set to an instance of an object.]
   nStuff.WebDevInfo.DevInfoModule.OnApplicationEndRequest(Object sender, EventArgs e) +63
   System.Web.SyncEventExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() +92
   System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously) +64
```

Version Information: Microsoft .NET Framework Version:2.0.50727.832; ASP.NET Version:2.0.50727.832

Errors while loading page from application

Metodologia

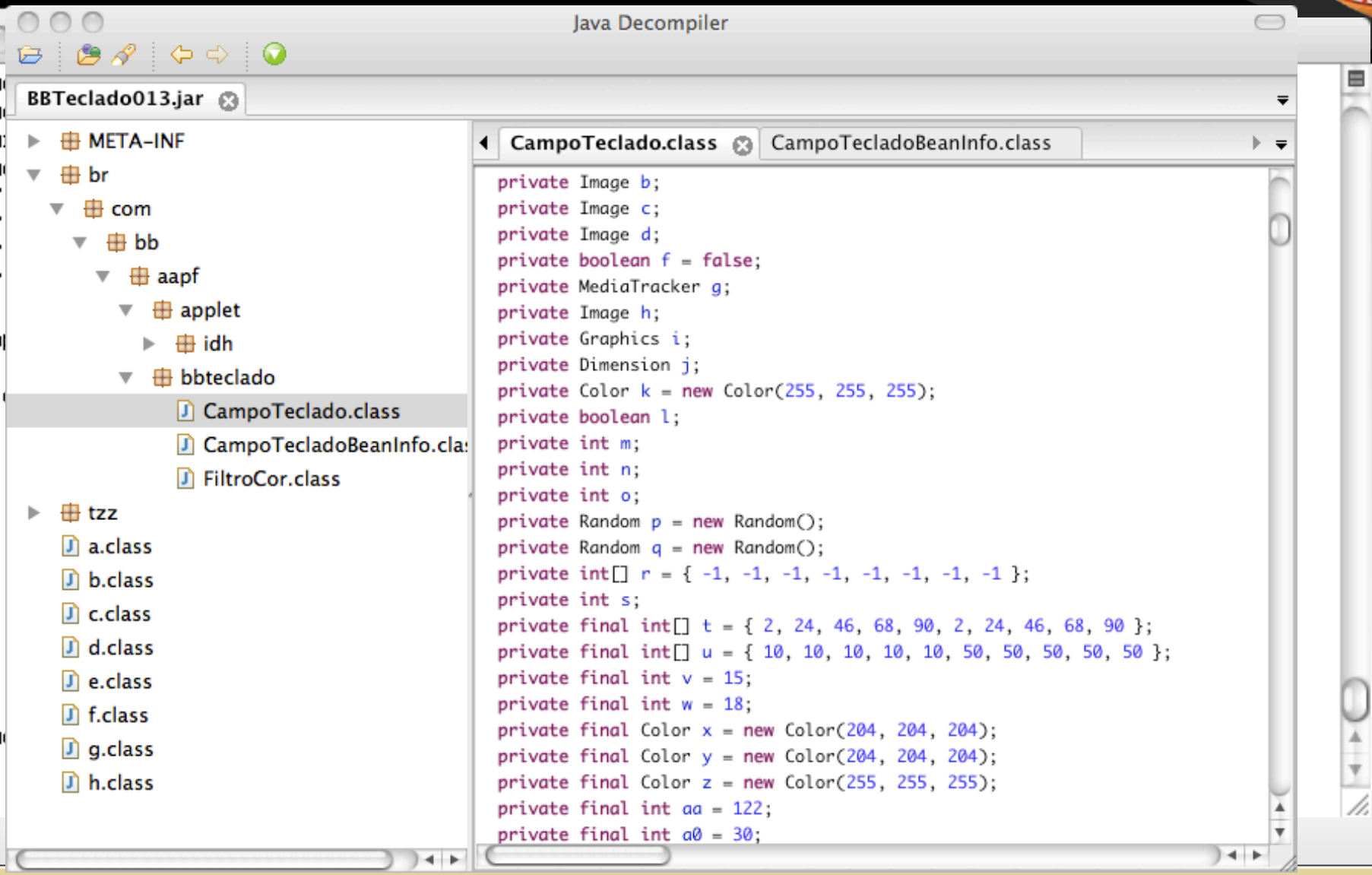
- Possível superfície de ataque
  - Bancos de dados -> Injeção SQL
  - Envio/download de arquivos -> Vulnerabilidades de *path traversal*
  - Apresentação de dados enviados pelo usuário -> XSS
  - Autenticação -> enumeração de usuários, senhas fracas, força bruta
  - Mensagens de erro -> Vazamento de informação
  - Componentes nativos -> *buffer overflows*
  - Componentes de terceiros -> Vulnerabilidades conhecidas

Metodologia

- Mapear a aplicação – Resumo
  - Enumerar conteúdo visível e funcionalidade
  - Uso de força bruta e inferência para localizar conteúdo escondido
  - Análise da aplicação
    - Funcionalidade, comportamento e mecanismos de segurança
  - Verificação da superfície de ataque

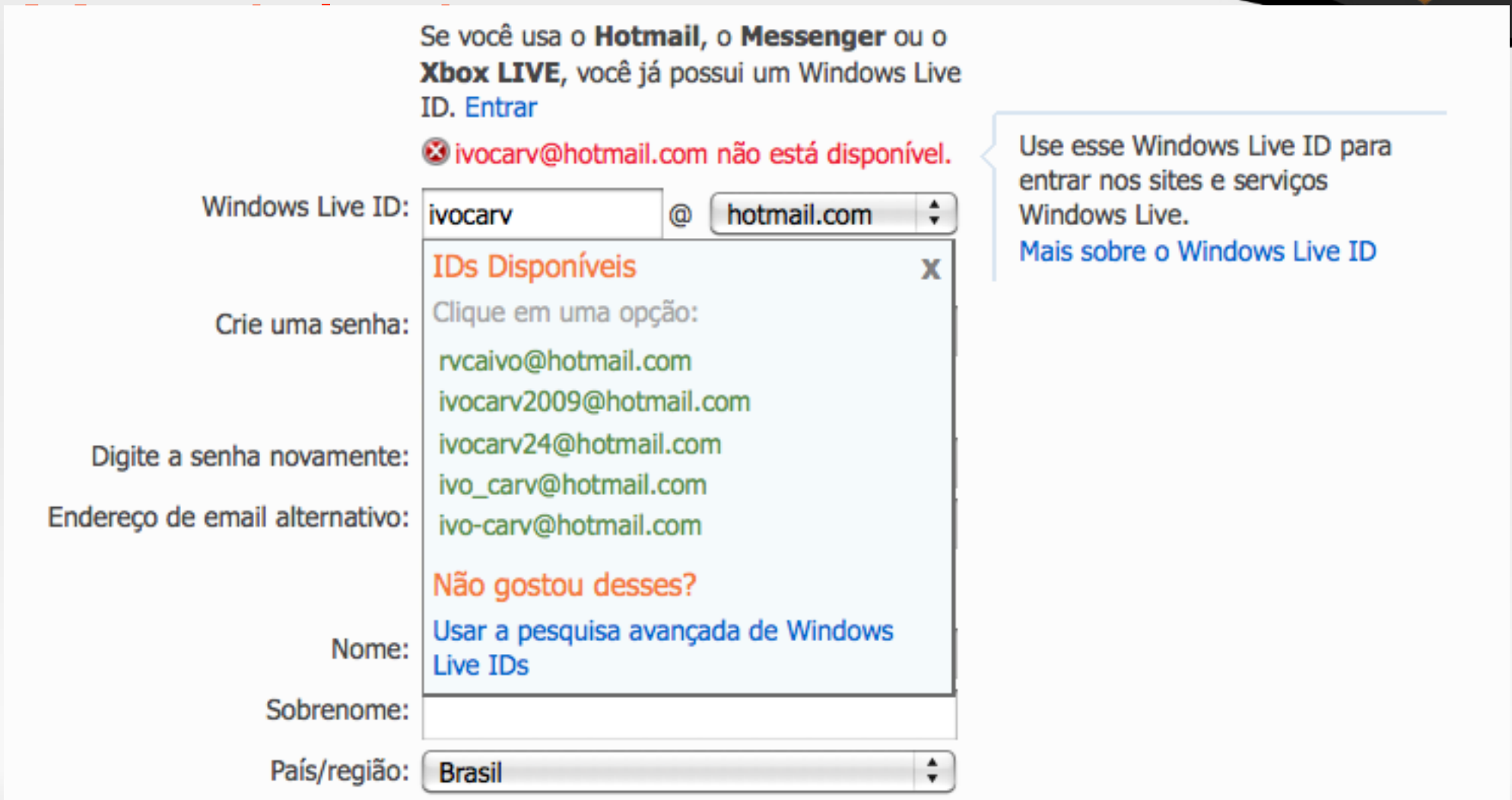
Metodologia

- Testar controles no lado do cliente
  - Campos de formulário escondidos
  - Parametros em URL
  - Dados obfuscados
    - Base64, hex, etc
  - Limites de tamanho em formulários
  - Validação Javascript (ex: CPF)
  - Atributos desabilitados (disabled="true")
  - Componentes compilados (Applets Java, etc)
    - Teclado Virtual

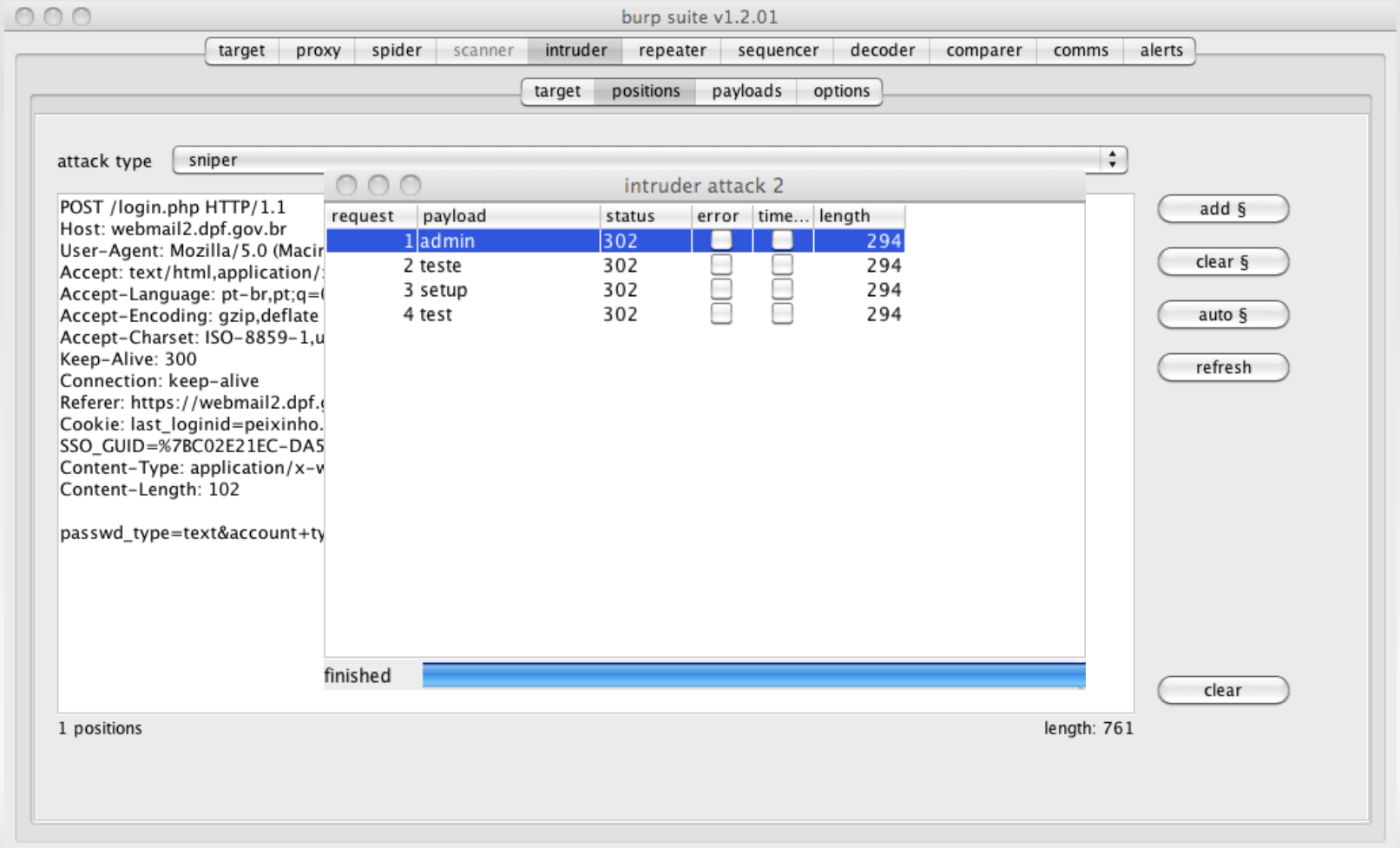

Java Decompiler
BBTeclado013.jar
META-INF
br
com
bb
aapf
applet
idh
bbteclado
CampoTeclado.class
CampoTecladoBeanInfo.cla
FiltroCor.class
tzz
a.class
b.class
c.class
d.class
e.class
f.class
g.class
h.class
CampoTeclado.class
CampoTecladoBeanInfo.class
private Image b;
private Image c;
private Image d;
private boolean f = false;
private MediaTracker g;
private Image h;
private Graphics i;
private Dimension j;
private Color k = new Color(255, 255, 255);
private boolean l;
private int m;
private int n;
private int o;
private Random p = new Random();
private Random q = new Random();
private int[] r = { -1, -1, -1, -1, -1, -1, -1, -1 };
private int s;
private final int[] t = { 2, 24, 46, 68, 90, 2, 24, 46, 68, 90 };
private final int[] u = { 10, 10, 10, 10, 10, 50, 50, 50, 50, 50 };
private final int v = 15;
private final int w = 18;
private final Color x = new Color(204, 204, 204);
private final Color y = new Color(204, 204, 204);
private final Color z = new Color(255, 255, 255);
private final int aa = 122;
private final int a0 = 30;

Metodologia

- Testar controles no lado do cliente
  - Outros componentes compilados
    - Flash
      - Flasm / Flare
    - Silverlight
      - Decompilável (dll em C#)
    - Active X
      - Decompilável se for em C#
      - Debugger (funções exportáveis - JavaScript)


Se você usa o Hotmail, o Messenger ou o Xbox LIVE, você já possui um Windows Live ID. Entrar
ivocarv@hotmail.com não está disponível.
Windows Live ID:
ivocarv
@
hotmail.com
IDs Disponíveis
Clique em uma opção:
rvcaivo@hotmail.com
ivocarv2009@hotmail.com
ivocarv24@hotmail.com
ivo_carv@hotmail.com
ivo-carv@hotmail.com
Não gostou desses?
Usar a pesquisa avançada de Windows Live IDs
Crie uma senha:
Digite a senha novamente:
Endereço de email alternativo:
Nome:
Sobrenome:
País/região:
Brasil
Use esse Windows Live ID para entrar nos sites e serviços Windows Live.
Mais sobre o Windows Live ID

Metodologia



burp suite v1.2.01

target | proxy | spider | scanner | intruder | repeater | sequencer | decoder | comparer | comms | alerts

intercept | options | history

Filter: hiding CSS, image and general binary content

#	host	method	URL	params	mod	status	length	MIME type	extension	title	SSL	IP	cookies	tim
686	http://webmail2.dpf.g...	GET	/phpgwapi/templates/default/js/simple_show_hi...	☐	☐	200	4519	script	js		☐	200.169.41.65		00
687	http://webmail2.dpf.g...	GET	/phpgwapi/templates/default/js/cookieManager.js	☐	☐	200	1960	script	js		☐	200.169.41.65		00
697	http://webmail2.dpf.g...	GET	/home/templates/default/images/navbar.png	☐	☐	200	36721	HTML	png	ExpressoMail...	☐	200.169.41.65		00
707	http://webmail2.dpf.g...	GET	/home/templates/default/images/navbar.png	☐	☐	200	36721	HTML	png	ExpressoMail...	☐	200.169.41.65		00
708	http://webmail2.dpf.g...	GET	/logout.php	☐	☐	302	905	HTML	php		☐	200.169.41.65	sessio...	00
715	http://webmail2.dpf.g...	GET	/login.php?cd=1	☑	☐	302	300	HTML	php		☐	200.169.41.65	conta...	00
716	https://mail.google.com	POST	/mail/channel/bind?VER=6&it=1636172&at=xn...	☑	☐	200	341	text			☑	74.125.91.17		00
717	https://webmail2.dpf....	GET	/login.php?cd=1	☑	☐	200	4661	HTML	php	ExpressoMail...	☑	200.169.41.65	conta...	00
718	https://mail.google.com	POST	/mail/channel/bind?VER=6&it=1649&at=xn3j2v...	☑	☐	200	341	text			☑	74.125.91.17		00
719	https://webmail2.dpf....	POST	/login.php	☑	☐	302	324	HTML	php		☑	200.169.41.65	conta...	00
720	https://webmail2.dpf....	GET	/login.php?cd=5	☑	☐	200	4657	HTML	php	ExpressoMail...	☑	200.169.41.65		00
721	http://safebrowsing.cli...	POST	/safebrowsing/downloads?client=navclient-auto-...	☑	☐	200	795	text			☐	74.125.91.138		00
722	http://safebrowsing-ca...	GET	/safebrowsing/rd/goog-malware-shavar_s_228...	☐	☐	200	5916				☐	74.125.9.83		00

request | response

raw | headers | hex

```
HTTP/1.1 302 Found
Date: Fri, 04 Dec 2009 02:04:16 GMT
Server: Apache
Set-Cookie: contador=2
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Location: /login.php?cd=5
Vary: Accept-Encoding
Keep-Alive: timeout=120
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
Content-Length: 2
```

< > 0 matches

Metodologia

- Manipulação insegura de credenciais
  - Credenciais enviadas por HTTP
  - Credenciais enviadas por HTTPS, porém o formulário de login é enviado via HTTP (verificação da origem)
  - Credenciais armazenadas em cookies
  - Credenciais passadas na URL (Query String)
    - Usar sempre POST
  - Credenciais passadas do servidor para o cliente (remember me)

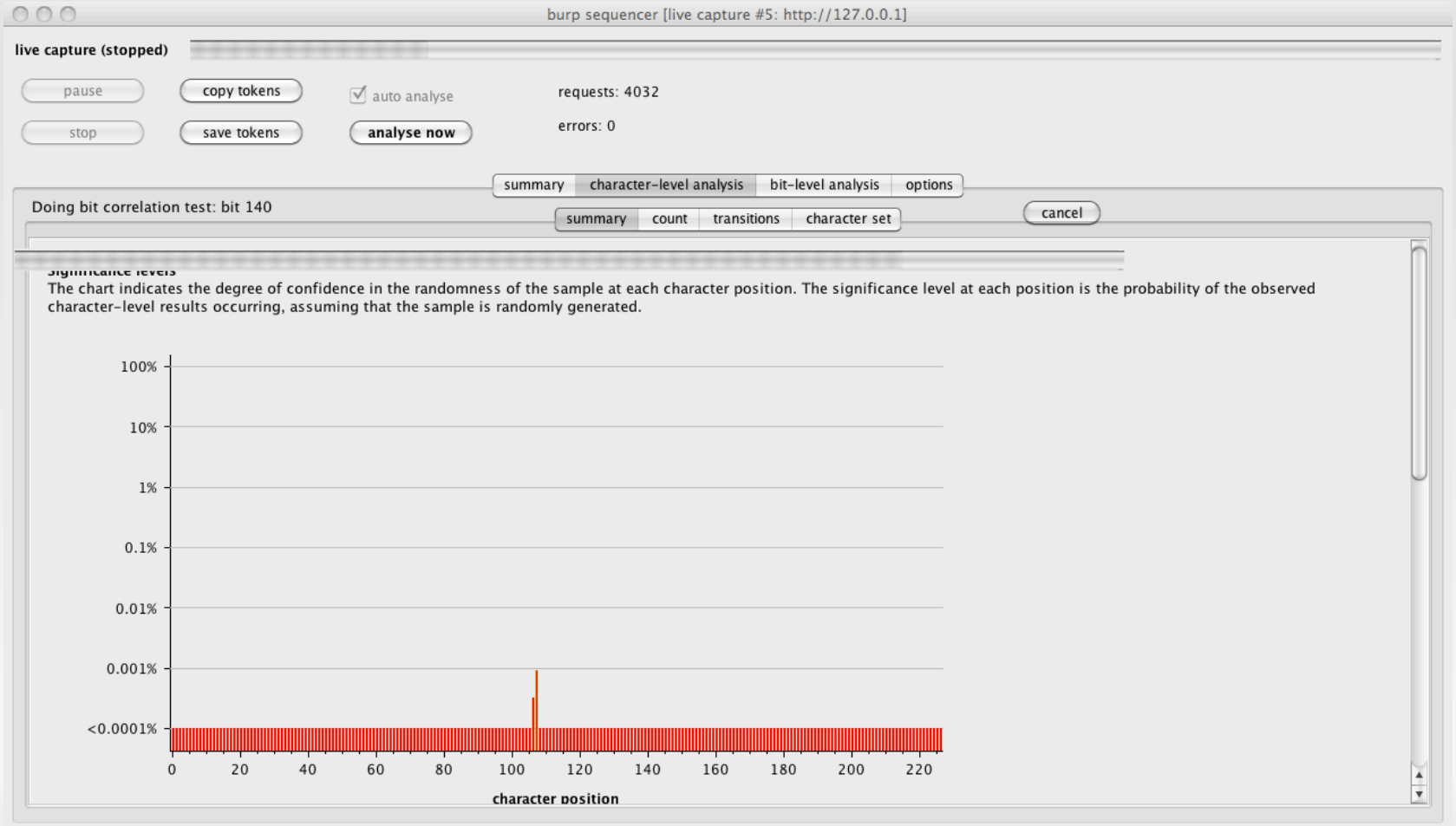
Metodologia

- Testar mecanismos de gerenciamento de sessão
  - Mecanismos de estado de sessão
    - HTTP é *stateless*
    - Necessidade de identificar sessões
    - Atribuição de *token* para cada sessão
    - *Token* reenviado em cada requisição
    - Vulnerabilidades
      - *Tokens* gerados de forma insegura
      - *Tokens* tratados de forma insegura

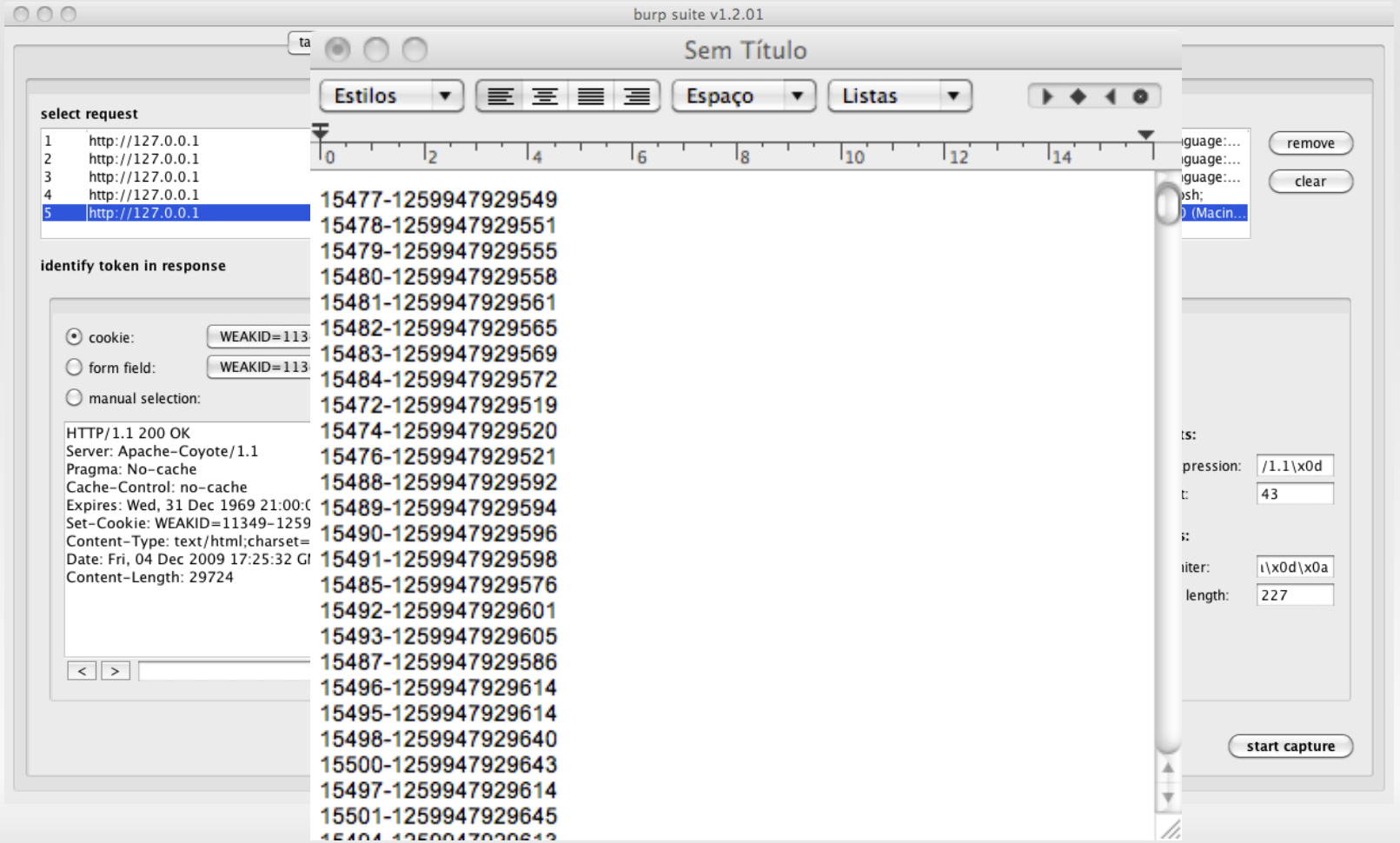
Metodologia

- Testar mecanismos de gerenciamento de sessão
  - Geração de *tokens*
    - *Tokens* com dados da sessão (ex: IP, username, etc).
    - *Tokens* codificados
      - Base64
      - Hexadecimal
  - *Tokens* previsíveis (aleatoriedade insegura)
  - *Tokens* cifrados (possibilidade de *tamper/ECB*)

Metodologia

- 

Metodologia



Metodologia

- Testar mecanismos de gerenciamento de sessão
  - Manipulação de *tokens*
    - Revelação de *tokens*
      - Trechos do site em HTTP
      - Tokens na URL
      - Captura de tokens em cookies (XSS)
    - Finalização de sessão
      - Funções de *logout* defeituosas

Metodologia

- Testar controles de acesso
  - URL's de acesso privilegiado públicas
    - URL's não são secretas!!
  - Controles baseados em identificação
    - Especificam recusos e ações acessados
    - Identificadores tambem não são secretos
  - Controles de acesso múltiplos
    - Apenas o primeiro estágio seguro
  - Arquivos estáticos

Metodologia

- Testar controles de acesso
  - Arquivos estáticos
    - Compras online de arquivos
    - URL estática
    - Acessível em contextos não autorizados
  - Métodos inseguros
    - admin=true

Metodologia

- Testar vulnerabilidades nos parâmetros de entrada
  - Injeção SQL
    - ‘ or 1=1
    - ‘ UNION SELECT …--
    - SELSELECTECT
    - Etc…

Metodologia

- Testar vulnerabilidades nos parâmetros de entrada
  - Injeção de comandos de sistema
    - ` & | ; < > \
    - script.cgi?a=b | cat /etc/passwd
    - <parametro> && dir c:\

Metodologia

- Testar vulnerabilidades do Web Server
  - Credenciais Padrão
    - interfaces de administração
  - Conteúdo padrão
    - phpinfo, test.cgi, etc.
  - Listagens de diretório
  - Virtual Hosting mal configurado (permissões)
  - Servidores Web vulneráveis (atualizações)

Conclusões

- Crescimento exponencial das aplicações Web
- Pouca cultura de segurança em aplicações Web
- SSL não torna seu site seguro
- URL's são públicas!!!
  - Use POST
- Filtros de pacote ineficientes para proteção do perímetro
  - Mod_security é uma boa idéia!!
- Pen-test importante para aplicações *home made*
  - Necessita de tempo e conhecimento (treino)
  - Ferramentas ajudam

Ivo de Carvalho Peixinho
Perito Criminal Federal
peixinho.icp @ dpf.gov.br