### Grupo de Resposta a Incidentes de Segurança – GRIS Departamento de Ciência da Computação Universidade Federal do Rio de Janeiro

# Encriptação no meebo

Estamos seguros mesmo?

Por: Manoel Fernando de Sousa Domingues Junior

Rio de Janeiro – Brasil 2009



#### Universidade Federal do Rio de Janeiro

Instituto de Matemática Departamento de Ciência da Computação Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ – Brasil

Encriptação no meebo GRIS-2009-A-NUM

Manoel Fernando de Sousa Domingues Junior

#### V. 1.0.

A versão mais recente deste documento pode ser obtida na página oficial do GRIS.

Este documento é Copyright© 2009 GRIS. Ele pode ser livremente copiado desde que sejam respeitadas as seguintes condições:

É permitido fazer e distribuir cópias inalteradas deste documento completo ou em partes, contanto que esta nota de copyright e distribuição seja mantida em todas as cópias, e que a distribuição não tenha fins comerciais. Se este documento for distribuído apenas em partes, instruções de como obtê-lo por completo devem ser incluídas. É vedada a distribuição de versões modificadas deste documento, bem como a comercialização de cópias, sem a permissão expressa do GRIS.

Embora todos os cuidados tenham sido tomados na preparação deste documento, o GRIS não garante a correção absoluta das informações nele contidas, nem se responsabiliza por eventuais conseqüências que possam advir do seu uso.



Grupo de Resposta a Incidentes de Segurança

# Índice

1- Introdução	4					
1.1 – O que é o meebo?						
2 – Indo aos experimentos.						
2.1 – Procedimentos	5					
2.2 – Ambiente dos experimentos	5					
2.3 – Executando o teste						
3 – Conclusões a cerca do teste	7					
4 – Criptografia no meebo	8					
4.1 - o SSL	8					
4.2 – o TLS	9					
5 – Referências						
Índias de ilustrasãos						
Indice de ilustrações						
Página inicial do meebo em julho de 2009.	4					
Topologia de rede para experimento						
Captura dos pacotes transferidos entre o cliente (10.10.10.197) e meebo (208.81.191.110) com						
navegador compatível com SSL e TLS.	6					
Pacotes com conteúdo criptografado	6					
Captura dos pacotes transferidos entre o cliente (10.10.10.197) e meebo (208.81.191.110) com						
navegador sem suporte a SSL e TLS	7					
Pacote com login em texto puro	8					



# 1- Introdução

Com a grande massificação do uso da internet, diversas formas de comunicação foram criadas, porém a que mais se intensificou foi a feita através de mensageiros instantâneos.

Com o surgimento da web 2.0 em meados de 2004 e seu conceito de web como plataforma, começaram a ser criados webmessengers, que são na verdade, serviços que rodam na nuvem dispensando a instalação de aplicativos localmente.

### 1.1 - O que é o meebo?

O meebo é um serviço da web 2.0 fundado em 2005, que permite que usuários de diversas redes de mensageiros instantâneos se comuniquem sem ter que fazer a instalação e utilização de aplicativos localmente. Seu principal diferencial é sua interface de fácil utilização e o fato de poder reunir amigos em uma lista para se comunicar em tempo real utilizando diferentes protocolos de comunicação instantânea.

Hoje o meebo conta com mais de 50 milhões de pessoas partilhando mais de 5 bilhões de mensagens por mês, além de ser uma das redes socias em maior crescimento na web.<sup>[1]</sup>



*Ilustração 1: Página inicial do meebo em julho de 2009*[1] Dados retirados da página do meebo na internet em 29 de julho de 2009

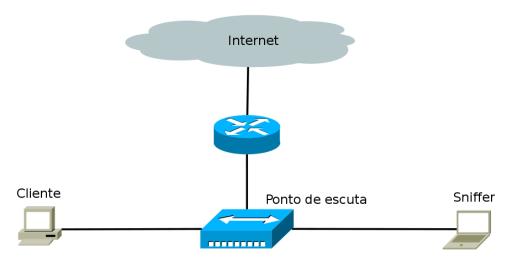
### 2 – Indo aos experimentos

#### 2.1 - Procedimentos

Os procedimentos realizados no teste consistiram em montar um ambiente de rede controlado em que seria colocado um computador executando um sniffer na rede e outro se conectando ao serviço meebo.

### 2.2 - Ambiente dos experimentos

Para executar os testes de segurança foi montada uma topologia de rede em ambiente controlado seguindo o seguinte modelo.



PC que vai se conectar ao meebo

PC que vai escutar a rede em modo promíscuo

Ilustração 2: Topologia de rede para experimento

Softwares usados em cada computador:

- O computador que vai se conectar ao meebo está rodando linux e usando como navegadores o Mozilla Firefox 3.5.2 e Opera 10.00, ambos com o SSL e TLS, ora habilitados, ora desabilitados;
- O computador que vai escutar a rede em modo promiscuo está rodando linux e usando como sniffer o Wireshark 1.2.1.

#### 2.3 - Executando o teste

Já com o software sniffer aberto e em modo de captura, nos conectamos ao meebo, e inserimos um login e senha.

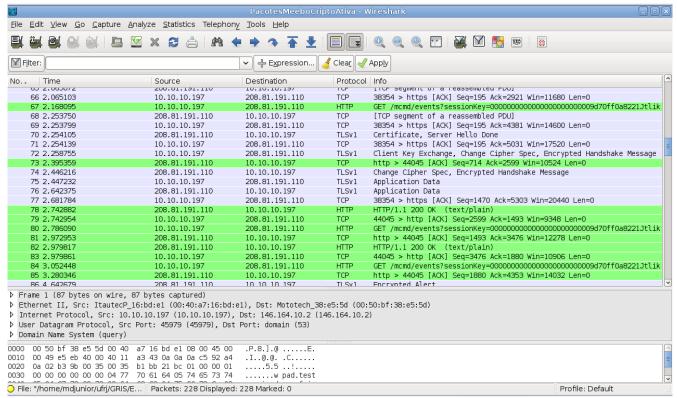


Ilustração 3: Captura dos pacotes transferidos entre o cliente (10.10.10.197) e meebo (208.81.191.110) com navegador compatível com SSL e TLS

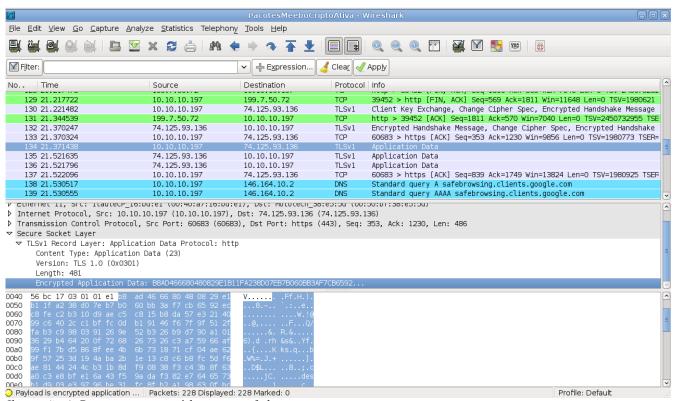


Ilustração 4: Pacotes com conteúdo criptografado

# 3 - Conclusões a cerca do teste

Tendo sido realizados inumeros testes para comprovar a veracidade dos resultados aqui representados, temos que o meebo utiliza criptografia (TLS 1.0, como se pode observar na ilustração 5) somente para o campo no qual diz respeito a senha de usuário, se o navegador for compatível com esse padrão. Caso o navegador não seja compatível com o TLS, o meebo utiliza um JavaScript em sua página para criptografar a senha com sua chave pública.

					D = = = 00 - = = 0 =	Winner Committee	
<b>Z</b>				acotesMeeboCriptol	Desativada	- Wireshark	
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> r	alyze <u>S</u> tatistics Telephon <u>y</u>	<u>T</u> ools <u>H</u> elp			
			Z X 2 📤   24 💠	→ → 🛧 🖢			
<b>¥</b> F	ilter:			→ Expression	of Clear €	Apply	
No		Time	Source	Destination	Protocol		
		2.002/72	200.01.131.110	10.10.10.137	TO	11ttp > 37302 [310, Act] 3cq-0 Ack-1 WIII-3070 Edil-0 133-1700	
		2.802801	10.10.10.197	208.81.191.110	TCP	54902 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0	
		2.802926	10.10.10.197	208.81.191.110	HTTP	POST /mcmd/login HTTP/1.1 (application/x-www-form-urlencoded)	
		2.952564	208.81.191.110	10.10.10.197	TCP	http > 54901 [ACK] Seq=713 Ack=2593 Win=11008 Len=0 TSV=1171572512 TS	
		2.992455	208.81.191.110	10.10.10.197	TCP	http > 54902 [ACK] Seq=1 Ack=1406 Win=8430 Len=0	
		2.998523	208.81.191.110	10.10.10.197	HTTP	HTTP/1.1 200 OK (text/plain)	
		2.998568	10.10.10.197	208.81.191.110	TCP	54902 > http [ACK] Seq=1406 Ack=236 Win=6432 Len=0	
		3.099391	208.81.191.110	10.10.10.197	HTTP	HTTP/1.1 200 OK (text/plain)	
		3.116528	10.10.10.197	208.81.191.110	HTTP	GET /mcmd/events?sessionKey=000000000000000000000000d70ff0a5f53UJrAY	
		3.138697	10.10.10.197	208.81.191.110	TCP	54901 > http [ACK] Seq=2593 Ack=1402 Win=9472 Len=0 TSV=2297012 TSER=	
		3.305176	208.81.191.110	10.10.10.197	HTTP	HTTP/1.1 200 OK (text/plain)	
		3.305266	10.10.10.197	208.81.191.110	TCP	54902 > http [ACK] Seq=2281 Ack=623 Win=7504 Len=0	
		3.405351	10.10.10.197	208.81.191.110	HTTP	GET /mcmd/events?sessionKey=000000000000000000000000000000000000	
	41	3.592606	208.81.191.110	10.10.10.197	TCP	http > 54901 [ACK] Seq=1402 Ack=3468 Win=12800 Len=0 TSV=1171573152 T	
	42 !	5.402568	68.142.101.254	10.10.10.197	TCP	http > 52054 [FIN, ACK] Seq=338 Ack=1627 Win=201264 Len=0 TSV=3627893	
	43 !	5.423928	208.81.191.110	10.10.10.197	TCP	http > 54902 [FIN, ACK] Seq=623 Ack=2281 Win=11240 Len=0	
	44 !	5.441914	10.10.10.197	68.142.101.254	TCP	52054 > http [ACK] Seq=1627 Ack=339 Win=8064 Len=0 TSV=2299315 TSER=3	
	45 !	5.463652	10.10.10.197	208.81.191.110	TCP	54902 > http [ACK] Seq=2281 Ack=624 Win=7504 Len=0	
	46	17.897882	10.10.10.197	208.81.191.110	TCP	54902 > http [FIN, ACK] Seq=2281 Ack=624 Win=7504 Len=0	
	47	17.897944	10.10.10.197	68.142.101.254	TCP	52054 > http [FIN, ACK] Seq=1627 Ack=339 Win=8064 Len=0 TSV=2311771 T	
	48	18.031840	68.142.101.254	10.10.10.197	TCP	http > 52054 [ACK] Seq=339 Ack=1628 Win=201264 Len=0 TSV=3627906335 T	
	49	18.084012	208.81.191.110	10.10.10.197	TCP	http > 54902 [ACK] Seg=624 Ack=2282 Win=11240 Len=0	
D Fr	ame	1 (74 bytes on wire, 7	4 bytes captured)				
Final II, Src: Itautecy 16:bd:e1 (00:40:a7:16:bd:e1), Dst: Mototech 38:e5:5d (00:50:bf:38:e5:5d)							
▶ Internet Protocol, Src: 10.10.10.197 (10.10.10.197), Dst: 68.142.101.254 (68.142.101.254) ▶ Transmission Control Protocol, Src Port: 52054 (52054), Dst Port: http (80), Seq: 0, Len: 0							
P 11	uiisii	1331011 CONTENDE PROTOCO	c, 510 1011. 52054 (52054)	, bac rore. Heep (or	,, seq. 0,	Lon. V	
					************		
				.P.8.].@E.			
				. <q.@.@d.< td=""><td></td><td></td></q.@.@d.<>			
			71 ec 00 00 00 00 a0 02	eV.Py. q			
0030	16	d0 ba 2e 00 00 02 04	05 b4 04 02 08 0a 00 23	#			
File	e: "/h	ome/mdjunior/ufrj/GRIS/E.	Packets: 65 Displayed: 65	Marked: 0		Profile: Default	

Ilustração 5: Captura dos pacotes transferidos entre o cliente (10.10.10.197) e meebo (208.81.191.110) com navegador sem suporte a SSL e TLS

Um fato importante de se observar, é que ao contrário do que diz a documentação do meebo, somente a senha é criptografada ficando o login em texto puro.

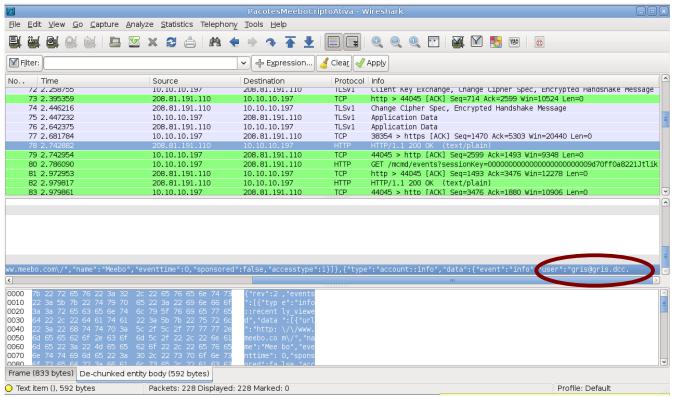


Ilustração 6: Pacote com login em texto puro

De acordo com o próprio site, ele não criptografa a página e as conversas para garantir a velocidade e uma melhor experiencia de uso. Para aqueles que preferem sacrificar a velocidade em busca de maior segurança, o serviço meebo disponibiliza o endereço: <a href="https://www.meebo.com">https://www.meebo.com</a>, em que todo conteúdo é criptografado.

Outro fato interessante é que nem o descontinuado webmessenger da microsoft, o que era mais usado no Brasil, apresentava opções envolvendo criptografia de todo conteúdo.

## 4 – Criptografia no meebo

#### 4.1 - o SSL

Com o uso crescente das redes de computadores nos anos 90, principalmente a internet, aumentou também a necessidade da criação de canais seguros para comunicação de dados sensíveis, dentro de redes que em sua maioria não garantem privacidade, integridade e autenticidade dos dados que nelas trafegam.

Assim foi criado o protocolo SSL, que provê privacidade, integridade e autenticidade dos dados que trafegam entre duas aplicações na rede. Isto ocorre através da autenticação das partes envolvidas e da criptografia dos dados transmitidos entre as partes. Esse protocolo ajuda a prevenir que intermediários entre as duas pontas da

comunicação tenham acesso indevido ou falsifiquem as informações que estão sendo transmitidas.

O SSL, atualmente na versão 3, funciona de forma simples: o servidor envia seu certificado digital para o cliente e este confere sua autenticidade, caso a autenticidade seja contestada a conexão é encerrada, se este for autêntico o cliente envia a requisição da chave pública do servidor, com a chave pública o cliente criptografa as informações e as envia para o servidor, somente o servidor com sua chave privada pode descriptografar as informações. Dessa forma temos a autenticidade, confidencialidade e integridade dos dados transferidos.

#### 4.2 - o TLS

O TLS 1.0 é uma evolução do SSL 3.0. Suas principais diferenças são:

- TLS é padronizado pelas RFC 2246 (v1.0) e RFC 4346 (v1.1)
- TLS usa o algoritmo keyed-Hashing for Message Authentication Code (HMAC) enquanto o SSL apenas Message Authentication Code (MAC). O algoritmo HMAC produz hashes mais seguros que o algoritmo MAC
- No TLS nem sempre é necessário recorrer à raiz de uma AC (Autoridade de Certificação) para usar uma certificação. Pode ser usada uma autoridade intermediária
- Novas mensagens de alerta
- O algoritmo Fortezza de criptografia não é suportado, pois não é aberto ao público. (Política da IETF)
- Diferenças em alguns campos dos cabeçalhos

No mais suas características principais permanecem iguais ao SSL 3.0.

### 5 – Referências

- Documentação oficial do meebo <a href="http://www.meebo.com/security/">http://www.meebo.com/security/</a>
- Grupo de Teleinformática e Automação <a href="http://www.gta.ufrj.br">http://www.gta.ufrj.br</a>
- Wikipedia: A enciclopédia livre <a href="http://pt.wikipedia.org">http://pt.wikipedia.org</a>
- RFC 5246 <a href="http://www.rfc-editor.org/rfc/rfc5246.txt">http://www.rfc-editor.org/rfc/rfc5246.txt</a>