

**SOFTWARE PARA SCANNERS DE REDES - WIRESHARK**

**VINICIUS SOUZA VASCONCELOS DOS  
SANTOS**

**VITOR FONSECA VERONEZI**

## SUMÁRIO

<b>1 – INTRODUÇÃO .....</b>	<b>3</b>
<b>2 – EMBASAMENTO TEÓRICO .....</b>	<b>5</b>
2.1 – Funcionamento da ferramenta.....	5
2.2 – Capturando senhas.....	8
2.3 – Detalhes dos cabeçalhos de dados entre camadas.....	11
<b>3 – CONCLUSÃO .....</b>	<b>15</b>
<b>REFERÊNCIAS .....</b>	<b>16</b>

## 1 – INTRODUÇÃO

Tanenbaum (2003) em seu livro “Redes de Computadores”, no capítulo 8 – “Segurança de redes” cita que durante as primeiras décadas da existência das redes de computadores, elas eram utilizadas principalmente por pesquisadores universitários, com a finalidade de enviar mensagens e também por funcionários de grandes empresas, para compartilhar impressoras e documentos. Sob estas condições, a segurança nunca precisou ter maiores cuidados. Hoje em dia, com milhões de usuários, transações bancárias, *softwares* de receita federal e sistemas que necessitam de cuidados maiores, as redes precisaram evoluir, gerando assim melhoras significativas em seus protocolos e algoritmos, tornando-a mais segura.

Tanenbaum (2003) e também Kurose e Ross (2013) abordam em seus livros, um capítulo direcionado a segurança de redes, onde abordam temas como: criptografias, algoritmos de chave assimétrica, assinaturas digitais, *Firewalls*, segurança em redes sem fio entre outros tópicos. Eles explicam como estes atingem as camadas e seu funcionamento interno. Ao final de cada capítulo no livro “*Redes de Computadores e a internet: uma abordagem top-down*” (6ª Edição) de Kurose e Ross (2013) é tratado um tópico chamado *Wireshark Lab*, onde este aborda uma ferramenta de *scanner* de rede.

*Scanners* de redes são programas utilizados para varrer os computadores em uma rede à procura de vulnerabilidade e/ou portas abertas, que possam receber possíveis ataques, podendo ocorrer tanto em servidores como em *hosts* finais. Eles buscam sistemas desprotegidos e trabalham de duas formas: (1) os *scanners* de portas (*portscanner*), que verificam as portas TCP/IP abertas de um sistema e os de (2) vulnerabilidades, que fazem a verificação das vulnerabilidades conhecidas nos programas que rodam no computador. Estes fazem testes necessários e com os resultados o administrador pode procurar melhorias no sistema.

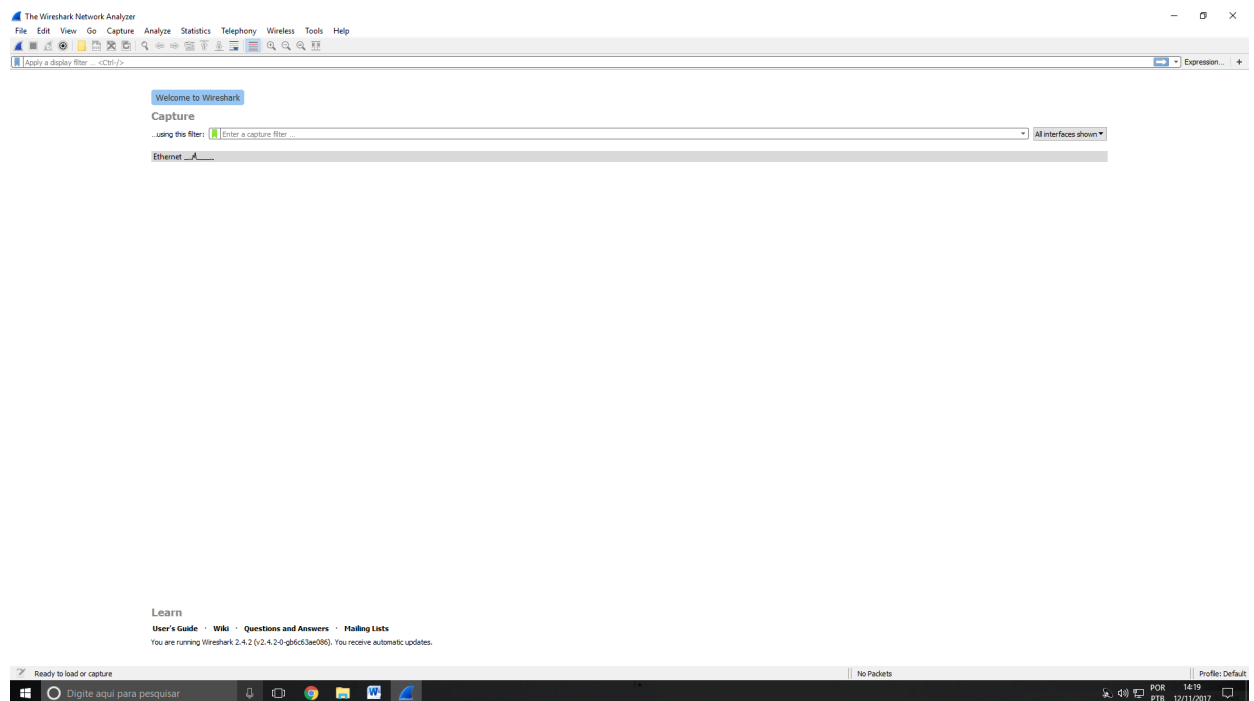
Como citado acima, estas ferramentas não são apenas para uso a fins de invadir e roubar dados pessoais, mas também tem o intuito de testar as vulnerabilidades de um programa, dando *feedbacks* necessários para fazer uma renovação na segurança de determinados sistemas.

Uma ferramenta muito utilizada no mundo *Linux* que também é citada no livro de Kurose e Ross (2013) é o *Wireshark*. Uma ferramenta muito poderosa e

utilizada mundialmente. Esta ferramenta permite verificar minuciosamente o que esta acontecendo em uma rede a nível microscópico de detalhes. Hoje em dia utilizam-se *Switches* e não mais *Hubs* dificultando um pouco mais a captura de dados (pacotes) em uma rede, já que os *Switches* não enviam dados para todas as portas (*broadcast*) como o *Hub* fazia.

Abaixo na Figura 1, mostra-se a tela inicial da ferramenta *Wireshark*. No capítulo 2 será abordado o seu funcionamento, terminando com uma conclusão no capítulo 3.

FIGURA 1 – Tela inicial da ferramenta *Wireshark*.



Fonte: Autor (2017).

## 2 – EMBASAMENTO TEÓRICO

A ferramenta *Wireshark* possui um rico conjunto de recursos como:

- Inspeção profunda de centenas de protocolos.
- Análise de captura ao vivo e off-line.
- Navegador padrão de pacotes de três painéis.
- Multi-plataforma: funciona no *Windows*, *Linux*, *MacOS*, *Solaris*, *FreeBSD*, *NetBSD* entre outros.
- Os filtros de exibição mais poderosos da indústria.
- Os dados ao vivo podem ser lidos de *Ethernet*, IEEE 802.11, PPP / HDLC, ATM, *Bluetooth*, USB, *Token Ring*, *Frame Relay*, FDDI e outros (dependendo da sua plataforma).
- Suporte de encriptação para muitos protocolos.
- Coloração pode ser aplicada na lista de pacotes para uma análise rápida e intuitiva.
- A saída pode ser exportada para XML, *PostScript*, CSV ou texto sem formatação.

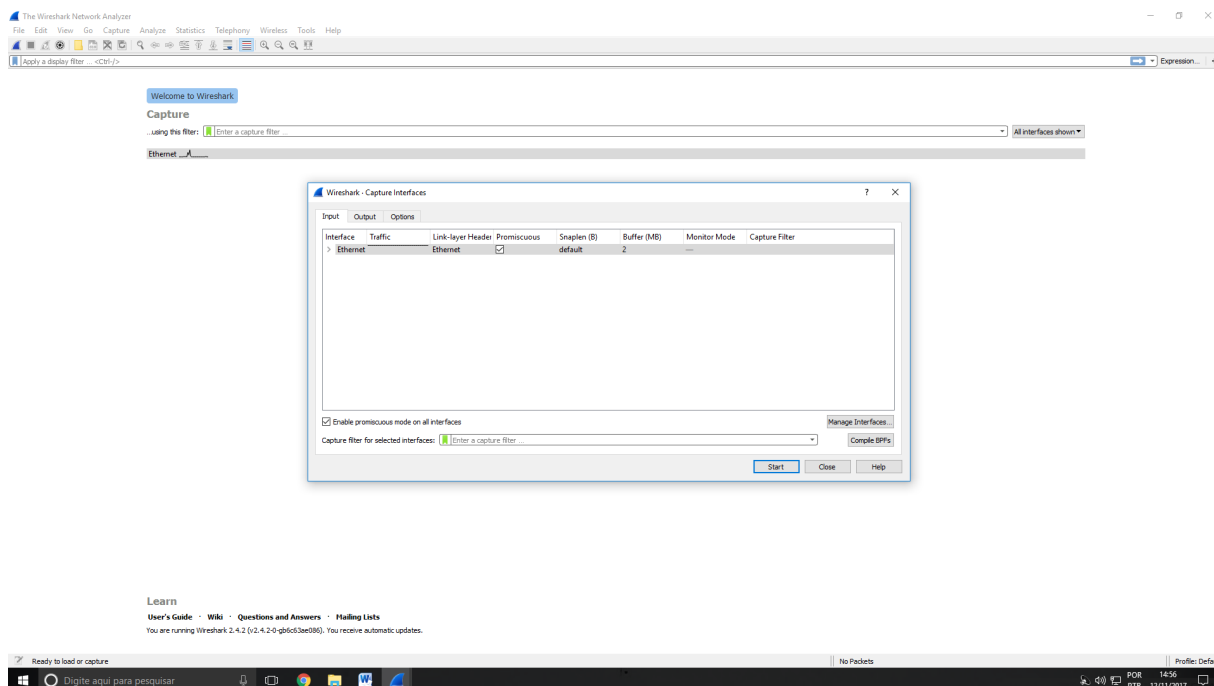
Algumas destas características poderão ser vistas nos tópicos abaixo.

### 2.1 – Funcionamento da ferramenta.

Após realizada a instalação da ferramenta, pode-se testá-la capturando alguns pacotes em uma rede em tempo real. A pesquisa mostra como a ferramenta faz a separação das camadas, diferenciação dos protocolos e o seu poder nas mãos de usuários mal-intencionados.

Para fazer uma varredura em uma rede com a ferramenta *Wireshark*, é necessário primeiramente escolher a interface de rede na qual se deseja iniciar o procedimento (encontra-se no menu superior, *Capture* e depois *Options*), como mostra a Figura 2. Após selecionar a(s) interface(s) (o computador pode ter mais de uma, exemplo, interface cabeada e uma placa *Wireless*), pressione o botão *Start* que o procedimento irá começar.

FIGURA 2 – Tela de escolha de interface(s).

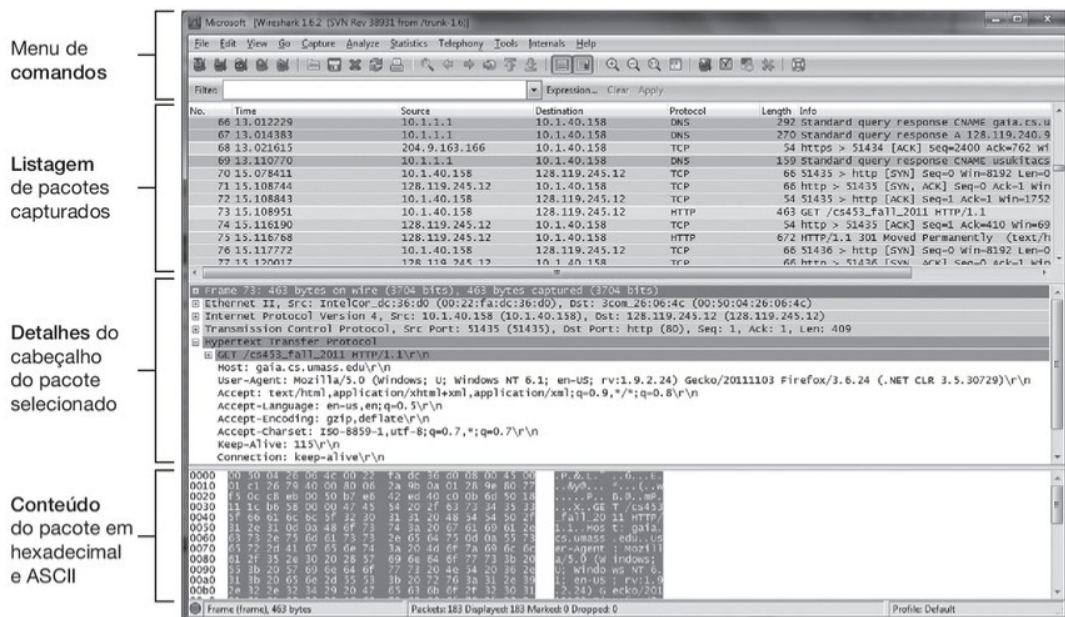


Fonte: Autor (2017).

Assim que a ferramenta começar a “farejar” os pacotes em sua rede, a seguinte tela irá aparecer como mostra a Figura 3. Kurose e Ross (2013) explica que a ferramenta básica para observar as mensagens trocadas entre entidades de protocolos em execução é chamada *de packet sniffer* (analisador de pacotes). Ela simplesmente “fareja” passivamente mensagens enviadas e recebidas por um computador e também exibe o conteúdo de vários campos de protocolos das mensagens capturadas.

Na parte de “listagem de pacotes capturados”, estão todos os pacotes que foram capturados durante a varredura. Ela traz uma tabela com os campos de *Time* (tempo), *Source* (origem), *Destination* (destino), *Protocol* (protocolo), *length* (tamanho) e *info* (informação), onde pode-se observar o IP de origem e destino, o protocolo que esta sendo usado (TCP, UDP, HTTP...), o tamanho e informações básicas. Assim que selecionada uma linha desta tabela, pode-se observar mais informações sobre este pacote através do menu “detalhes do cabeçalho do pacote selecionado” como mostra a Figura 4.

FIGURA 3 – Tela do analisador de pacotes.



Fonte: Livro do Kurose | Ross, Redes de computadores e a internet: uma abordagem top-down (6.ed, pág. 58, 2013).

FIGURA 4 – Menu de exibição do cabeçalho do pacote escolhido.

```
> Frame 240: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface 0
> Ethernet II, Src: Tp-LinkT_98:3a:88 (84:16:f9:98:3a:88), Dst: AsustekC_eb:46:66 (34:97:f6:eb:46:66)
> Internet Protocol Version 4, Src: 172.217.30.72, Dst: 192.168.0.104
> Transmission Control Protocol, Src Port: 80, Dst Port: 53511, Seq: 5721, Ack: 356, Len: 1430
> Hypertext Transfer Protocol
```

Fonte: Autor (2017).

Aqui mostra com detalhes minuciosos e microscópicos todas as informações que estão contidas dentro de um determinado pacote, os protocolos que ele utilizou, seu IP de origem e destino, endereço MAC, portas e muito mais. Esta parte da ferramenta traz as informações separadas em camadas como mostra a Figura 4, desde a camada de aplicação (*Hypertext Transfer Protocol* - HTTP) até a camada de enlace (na Figura 4, escrito *Frame*), passando pela camada de transporte e rede. Nos próximos tópicos a pesquisa vai abordar com mais detalhes a captura e extração de informação de um determinado pacote e como a segurança em redes é importante quando o assunto são dados pessoais dos usuários. O tópico abaixo traz um pequeno exemplo de como capturar uma senha com a ferramenta *Wireshark*.

## 2.2 – Capturando senhas.

Percebemos na Figura 3, uma área denominada “conteúdo de um pacote em hexadecimal e ASCII”. Nesta parte se traz o conteúdo que está dentro de um determinado pacote escolhido, como o protocolo da camada de aplicação que ele utiliza, seu HTML e TAGS, site acessado, cookies e alguns valores um pouco bagunçados como podemos ver na Figura 5.

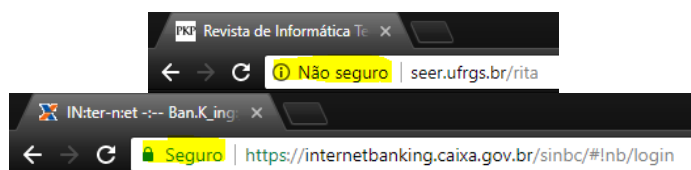
FIGURA 5 – Conteúdo de um pacote em hexadecimal.

```
0210 72 3a 20 68 74 74 70 3a 2f 2f 73 65 65 72 2e 75 r: http://seer.u
0220 66 72 67 73 2e 62 72 2f 72 69 74 61 0d 0a 41 63 frgs.br/rita.Ac
0230 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 cept-Encoding: g
0240 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a 41 63 zip, deflate.Ac
0250 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 70 cept-Language: p
0260 74 2d 42 52 2c 70 74 3b 71 3d 30 2e 38 2c 65 6e t-BR,pt;q=0.8,en
0270 2d 55 53 3b 71 3d 30 2e 36 2c 65 6e 3b 71 3d 30 -US;q=0.6,en;q=0
0280 2e 34 0d 0a 43 6f 6f 6b 69 65 3a 20 5f 5f 75 74 .4..Cookie: __ut
0290 6d 74 3d 31 3b 20 4f 4a 53 53 49 44 3d 65 65 73 mt=1; OJ SSID=ees
02a0 33 64 32 33 61 64 6c 31 71 35 74 6a 74 6f 35 34 3d23ad11 q5tjto54
02b0 6f 63 6c 6d 35 30 30 3b 20 5f 5f 75 74 6d 61 3d oc1m500; __utma=
02c0 32 30 31 30 34 38 37 34 38 2e 35 33 33 35 33 35 20104874 8.533535
02d0 38 39 31 2e 31 35 31 30 34 33 34 35 34 39 2e 31 891.1510 434549.1
02e0 35 31 30 34 33 34 35 34 39 2e 31 35 31 30 35 31 51043454 9.151051
02f0 34 35 37 36 2e 32 3b 20 5f 5f 75 74 6d 62 3d 32 4576.2; __utmb=2
0300 30 31 30 34 38 37 34 38 2e 33 2e 31 30 2e 31 35 01048748 .3.10.15
0310 31 30 35 31 34 35 37 36 3b 20 5f 5f 75 74 6d 63 10514576 ; __utmc
0320 3d 32 30 31 30 34 38 37 34 38 3b 20 5f 5f 75 74 =201048748; __ut
0330 6d 7a 3d 32 30 31 30 34 38 37 34 38 2e 31 35 31 mz=20104 8748.151
0340 30 35 31 34 35 37 36 2e 32 2e 32 2e 75 74 6d 63 0514576. 2.2.utmc
0350 73 72 3d 67 6f 6f 67 6c 65 7c 75 74 6d 63 63 6e sr=googl e|utmccn
0360 3d 28 6f 72 67 61 6e 69 63 29 7c 75 74 6d 63 6d =(organi c)|utmcm
0370 64 3d 6f 72 67 61 6e 69 63 7c 75 74 6d 63 74 72 d=organi c|utmctr
0380 3d 28 6e 6f 74 25 32 30 70 72 6f 76 69 64 65 64 =(not%20 provided
0390 29 0d 0a 0d 0a 75 73 65 72 6e 61 6d 65 3d 54 45 )....use rname=TE
03a0 53 54 45 26 70 61 73 73 77 6f 72 64 3d 31 32 33 STE&pass word=123
```

Fonte: Autor (2017).

No meio desta “bagunça” que se obtêm informações cruciais. Foram acessados dois sites para uma análise destes dados que são trazidos pela ferramenta. O primeiro, um site qualquer de uma revista de informática que não contém segurança de acesso. O site utiliza protocolo HTTP simples, sem criptografia dos dados. E o segundo, o site do *Internet Banking* da Caixa que contém protocolo HTTPS, certificados digitais, segurança com encriptação dos dados, entre outras técnicas, como mostra a Figura 6.

FIGURA 6 – Sites acessados, com e sem segurança de encriptação dos dados.



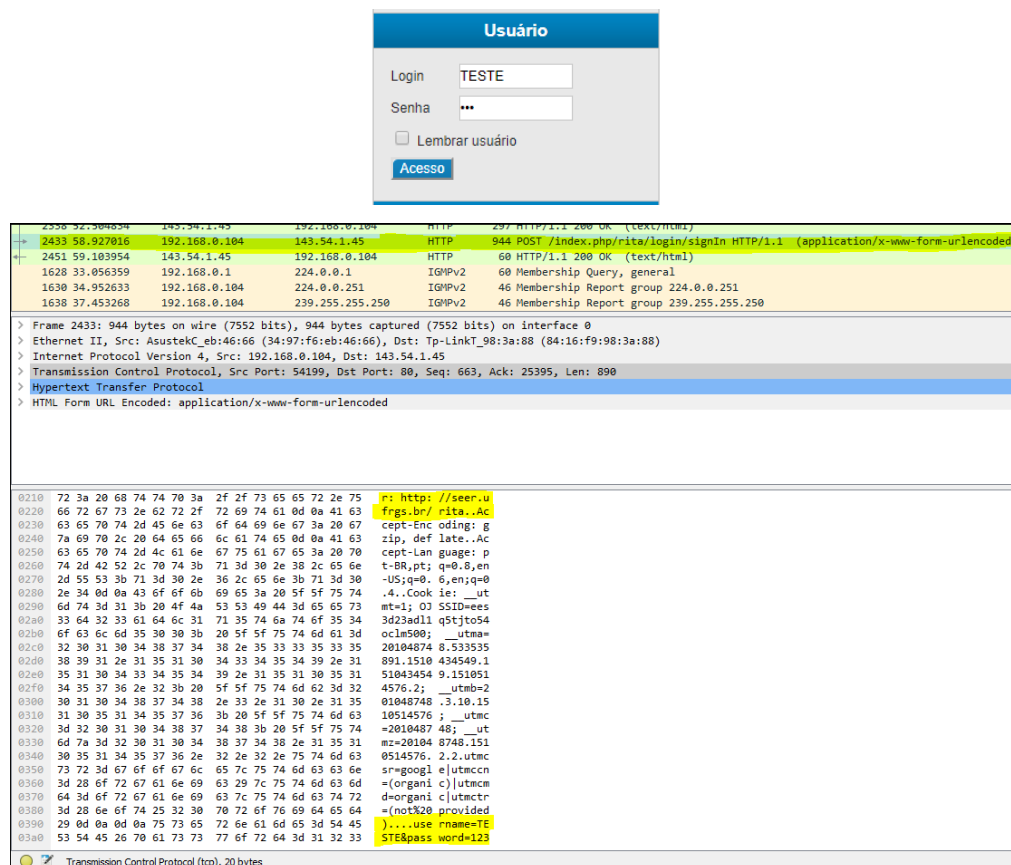
Fonte: Autor (2017).



Os sites que não contem encriptação de dados transferem seus dados em um arquivo de texto normal. Estes quando capturados podem ser facilmente abertos e lidos por qualquer um. Neles contem informação como o site que foi acessado e se tentar logar em uma conta, aparecerá o *login* e senha do usuário, como mostra a Figura 7, com um arquivo de dados capturado através da pesquisa feita no primeiro site da revista de informática.

Após a tentativa de logar com um usuário chamado **TESTE** e com a senha **123**, podemos analisar que estes aparecem de forma explicita no dado capturado pela ferramenta *Wireshark*, enquanto estamos efetuando uma varredura na rede, e utilizando o computador para acessar os sites qualquer.

FIGURA 7 – Login e senha capturados.



Fonte: Autor (2017).

Selecionada a linha da tabela do protocolo HTTP (da camada de aplicação) do tipo POST, tipo que envia dados para um determinado formulário gerando uma requisição, no caso da Figura 7, a tentativa de logar no site da revista de informática. Observando a imagem percebe-se facilmente os respectivos *login* e senha, marcados de marca texto na tela do *Wireshark*. Porém isto já não ocorre no

segundo exemplo de tentativa de acesso ao *Internet Banking* da Caixa, como mostrado na Figura 6. O site da Caixa oferece o protocolo na camada de aplicação chamado de HTTPS que tem encriptação de dados e certificados digitais, para prevenir contra este tipo de problema de roubos de dados pessoais.

Observando a Figura 8, percebe-se que com os dados criptografados a leitura e extração de informações não são tão simples, chegando a ser impossível. Mesmo que estes dados caiam nas mãos de uma potente máquina, é quase impossível (se não for impossível) descriptografar estes dados. Pegando o exemplo dos bancos, eles têm criptografias próprias que só a base de dados deles entende, com algoritmos poderosos e específicos, tornando as redes mais seguras.

FIGURA 8 – Tentativa de leitura dos dados do site da Caixa – *Internet Banking*.

No.	Time	Source	Destination	Protocol	Length	Info
1031	15.248153	fe80::a18d:4e5e:d24...	fe80::dd61:dc1b:384...	HTTP/X...	807	POST /b741dc33-bfa0-45b3-9570-1dd930cfdee/ HTTP/1.1
1032	15.248216	fe80::a18d:4e5e:d24...	fe80::dd61:dc1b:384...	HTTP/X...	807	POST /b741dc33-bfa0-45b3-9570-1dd930cfdee/ HTTP/1.1
1056	15.258651	fe80::dd61:dc1b:384...	fe80::a18d:4e5e:d24...	HTTP/X...	255	HTTP/1.1 200
Frame 1032: 807 bytes on wire (6456 bits), 807 bytes captured (6456 bits) on interface 0						
Ethernet II, Src: AsustekK_eb:46:66 (34:97:f6:eb:46:66), Dst: MitacInt_7b:20:7f (08:22:4d:7b:20:7f)						
Internet Protocol Version 6, Src: fe80::a18d:4e5e:d241:9703, Dst: fe80::dd61:dc1b:384e:3dc6						
Transmission Control Protocol, Src Port: 54309, Dst Port: 5357, Seq: 241, Ack: 1, Len: 733						
0000	00 22 4d 7b 20 7f 34 97	f6 eb 46 86 86 dd 60 09	.M{ .4. .FF..			
0010	09 2a 02 f1 06 80 fe 80	00 00 00 00 00 00 01 8d	*.....			
0020	4e 5e d2 41 97 03 fe 80	00 00 00 00 00 00 dd 61	N^A.....a			
0030	dc 1b 38 4e 3d c6 d4 25	14 ed 23 46 53 23 50 9e	..BN%.% .#F5AP.			
0040	cc 11 50 18 01 02 1b 2f	00 00 3c 3f 78 6d 6c 20	..P.... /...?xml			
0050	76 65 72 73 69 6f 6e 3d	22 31 2e 30 22 20 65 6e	version="1.0" en			
0060	63 6f 64 69 6e 67 3d 22	75 74 66 2d 38 22 3f 3e	coding=" utf-8"?>			
0070	3c 73 6f 61 70 3a 45 6e	76 65 6c 6f 70 65 20 78	<soap:en velope v			
0080	6d 6c 6e 73 3a 73 6f 61	70 3d 22 68 74 70 3a	mln:soa pe="http:			
0090	2f 2f 77 77 77 2e 77 33	2e 6f 72 67 2f 32 30 30	//www.w3 .org/200			
00a0	33 2f 30 35 2f 73 6f 61	70 2d 65 6e 76 65 6c 6f	3/05/soa p-envelo			
00b0	70 65 22 20 78 6d 6c 6e	73 3a 77 73 61 3d 22 68	pe" xmln s:wsa="h			
00c0	74 74 70 3a 2f 2f 73 63	68 65 6d 61 73 2e 78 6d	tp://sc hemas.xml			
00d0	6c 73 6f 61 70 2e 6f 72	67 2f 77 73 2f 32 30 30	lsoap.or g/ws/200			
00e0	34 2f 30 38 2f 61 64 64	72 65 73 73 69 6e 67 22	4/08/add ressing"			
00f0	20 78 6d 6c 6e 73 3a 6c	6d 73 3d 22 68 74 70 3a	xmln:s1 ms="http			
0100	3a 2f 2f 73 63 68 65 6d	61 73 2e 6d 69 63 72 6f	://schem as.micro			
0110	73 6f 66 74 2e 63 6f 6d	2f 77 69 6e 64 6f 77 73	soft.com /windows			
0120	2f 6c 6d 73 2f 32 30 30	37 2f 30 38 22 3e 3c 73	/lms/200 7/00">s			
0130	6f 61 70 3a 48 65 61 64	65 72 3e 3c 77 73 61 3a	oap:Head er>wsa:			
0140	54 6f 3e 75 72 6e 3a 75	75 69 64 3a 62 37 34 31	To>urn:u uid:b741			
0150	64 63 33 33 2d 62 66 61	30 2d 34 35 62 33 2d 39	dc33-bfa 0-45b3-9			
0160	35 37 30 2d 31 64 64 39	33 30 63 66 64 63 65 65	570-1dd9 30cfdee			
0170	3c 2f 77 73 61 3a 5f 6f	3e 3c 77 73 61 3a 41 63	</wsa:To>>wsa:Ac			
0180	74 69 6f 6e 3e 68 74 74	70 3a 2f 2f 73 63 68 65	tionhtt p://sche			
0190	6d 61 73 2e 78 6d 6c 73	6f 61 70 2e 6f 72 67 2f	mas.xmls oap.org/			
01a0	77 73 2f 32 30 30 34 2f	30 39 2f 74 72 61 6e 73	ws/2004/ 09/trans			
01b0	66 65 72 2f 47 65 74 3c	2f 77 73 61 3a 41 63 74	fer/Get< /wsa:Act			
01c0	69 6f 6e 3e 3c 77 73 61	3a 40 65 73 73 61 67 65	ion>wsa :Message			
01d0	49 44 3e 75 72 6e 3a 75	75 69 64 3a 64 61 32 37	IDurn:u uid:da27			
01e0	38 35 62 63 2d 62 39 35	32 2d 34 32 32 62 2d 39	85bc-b95 2-422b-9			
01f0	38 62 64 2d 62 61 66 3a	65 36 64 39 39 37 32 63	8bd-baf4 e6d9972c			
0200	3c 2f 77 73 61 3a 4d 65	73 73 61 67 65 49 44 3e	</wsa:Me ssageID>			
0210	3c 77 73 61 3a 52 65 70	6c 79 54 6f 3e 3c 77 73	<wsa:Rep lyTo><ws			
0220	61 3a 41 64 64 72 65 73	73 3e 68 74 70 3a 2f	a:Address s:http:			
0230	2f 73 63 68 65 6d 61 73	2e 78 6d 6c 73 6f 61 70	/schemas .xmlsoap			
0240	2e 6f 72 67 2f 77 73 2f	32 30 30 34 2f 30 38 2f	.org/ws/ 2004/08/			
0250	61 64 64 72 65 73 69 6e	67 2f 72 6f 6c 65 2f	addressi ng/role/			
0260	61 6e 6f 6e 79 6d 6f 75	73 3c 2f 77 73 61 3a 41	anonymou s</wsa:A			
0270	64 64 72 65 73 73 3e 3c	2f 77 73 61 3a 52 65 70	ddress>< /wsa:Rep			
0280	6c 79 54 6f 3e 3c 77 73	61 3a 46 72 6f 6d 3e 3c	lyTo>ws a:From<			
0290	77 73 61 3a 41 64 64 72	65 73 73 3e 75 72 6e 3a	wsa:Addr ess>urn:			
02a0	75 75 69 64 3a 33 31 37	64 32 31 65 36 2d 35 63	uid:317 d21e6-5c			
02b0	32 32 2d 34 3a 66 30 2d	62 65 34 61 2d 62 31 38	22-44f0- be4a-b18			
02c0	30 62 33 38 39 32 64 38	33 3c 2f 77 73 61 3a 41	0b3892d8 3</wsa:A			
02d0	64 64 72 65 73 73 3e 3c	2f 77 73 61 3a 46 72 6f	ddress> /wsa:Fro			
02e0	6d 3e 3c 6c 6d 73 3a 4c	61 72 67 65 4d 65 74 61	m<classL argeMeta			
02f0	64 61 74 61 53 75 70 70	6f 72 74 2f 3e 3c 2f 73	dataSupp ort></s			
0300	6f 61 70 3a 48 65 61 64	65 72 3e 3c 73 6f 61 70	oap:Head er><soap			
0310	3a 42 6f 64 79 2f 3e 3c	2f 73 6f 61 70 3a 45 6e	:Body/>< /soap:En			
Frame (807 bytes)    Reassembled TCP (973 bytes) Transmission Control Protocol (tcp), 20 bytes						

Fonte: Autor (2017).

Após observar estes dois exemplos das falhas de segurança em determinados sites, um terceiro exemplo foi escolhido para explicar melhor como a ferramenta *Wireshark* trabalha sobre as camadas do modelo TCP/IP (o mais usado mundialmente) e como ela trata e exhibe as informações que são trazidas dentro dos

pacotes. O próximo tópico mostrará isto na tentativa de acesso ao Aprender Unoeste no site oficial da Unoeste.

## 2.3 – Detalhes dos cabeçalhos de dados entre camadas.

Após acessar o site [www.unoeste.br](http://www.unoeste.br) e logar com um usuário e senha na área do aluno (Aprender Unoeste) obtivemos os seguintes dados mostrados na Figura 9.

FIGURA 9 – Cabeçalho de dados do site da Unoeste.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)
Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0 > Ethernet II, Src: AsustekC_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT_98:3a:88 (84:16:f9:98:3a:88) > Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4 > Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928						
0000	69 67 69 6e 3a 20 68 74 74 70 3a 2f 2f 77 77 77	igin: http://www				
0000	2e 75 6e 6f 65 73 74 65 2e 62 72 0d 0a 55 70 67	.unoeste.br..Upg				
0000	72 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65	rade-Inte cure-Re				
0000	71 75 65 73 74 73 3a 20 31 0d 0a 43 6f 6e 74 65	quests: 1..Conte				
0000	6e 74 2d 54 79 70 65 3a 20 61 70 70 6c 69 63 61	nt-Type: applica				
0100	74 69 6f 6e 2f 78 2d 77 77 77 2d 66 6f 72 6d 2d	tion/x-w ww-form-				
0110	75 72 6c 65 6e 63 6f 64 65 64 0d 0a 55 73 65 72	urlencod ed..User				
0120	2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f	-Agent: Mozilla/				
0130	35 2e 30 20 28 57 69 6e 64 6f 77 73 20 4e 54 20	5.0 (win dows NT				
0140	31 30 2e 30 3b 20 57 69 6e 36 34 3b 20 78 36 34	10.0; Wl n64; x64				
0150	29 20 41 70 70 6c 65 57 65 62 4b 69 74 2f 35 33	) AppleW ebKit/53				
0160	37 2e 33 36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b	7.36 (KH TML, lik				
0170	65 20 47 65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f	e Gecko) Chrome/				
0180	36 31 2e 30 2e 33 31 36 33 2e 31 30 30 20 53 61	61.0.316 3.100 5a				
0190	66 61 72 69 2f 35 33 37 2e 33 36 0d 0a 41 63 63	fari/537 .36..Acc				
01a0	65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 61	ept: tex t/html,a				
01b0	70 70 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c	pplicati on/xhtml				
01c0	2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e	=xml,app lication				
01d0	2f 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65	/xml;q=0 .9,image				
01e0	2f 77 65 62 70 2c 69 6d 61 67 65 2f 61 70 6e 67	/webp,im age/apng				
01f0	2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 52 65 66 65	,*/*;q=0 .8..Refe				
0200	72 65 72 3a 20 68 74 74 70 3a 2f 2f 77 77 77 2e	rer: htt p://www.				
0210	75 6e 6f 65 73 74 65 2e 62 72 2f 4c 6f 67 69 6e	unoeste. br/Login				
0220	2f 49 6e 64 65 78 3f 72 61 4d 61 74 3d 31 30 31	/Index?n aMet=101				
0230	34 32 36 36 36 39 26 70 3d 31 33 31 35 34 39 38	426669&p -1315498				
0240	39 38 31 31 35 30 32 39 39 35 30 0d 0a 41 63 63	98115029 950..Acc				
0250	65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a	ept-Enco ding: gz				
0260	69 70 2c 20 64 65 66 6c 61 74 65 0d 0a 41 63 63	ip, defl ate..Acc				
0270	65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 70 74	ept-Lang uage: pt				
0280	2d 42 52 2c 70 74 3b 71 3d 30 2e 38 2c 65 6e 2d	-BR,pt;q =0.8,en-				
0290	55 33 20 5f 75 74 6d 62 3d 32 36 32 30 35 35	US;q=0.8 ,en;q=0.				
02a0	34 0d 0a 43 6f 6f 6b 69 65 3a 20 5f 5f 75 74 6d	4..Cooki e: _utm				
02b0	74 5f 55 41 2d 31 35 35 35 31 30 35 2d 31 3d 31	t_UA=155 5105-1=1				
02c0	3b 20 5f 5f 75 74 6d 61 3d 32 36 32 30 35 35 38	; _utma =2620558				
02d0	38 31 2e 31 30 35 36 31 34 37 39 32 36 2e 31 35	81.10561 47926.15				
02e0	30 39 35 37 32 31 30 35 2e 31 35 31 30 35 31 33	09572105 .1510513				
02f0	36 32 37 2e 31 35 31 30 35 31 34 39 31 33 2e 32	627.1510 514913.2				
0300	39 3b 20 5f 5f 75 74 6d 62 3d 32 36 32 30 35 35	9; _utb =262055				
0310	38 38 11 2e 37 2e 31 30 2e 31 35 31 30 35 31 34	881.7.10 .1510514				
0320	39 31 33 3b 20 5f 5f 75 74 6d 63 3d 32 36 32 30	913; _u tmc=2620				
0330	35 35 38 38 31 3b 20 5f 5f 75 74 6d 6a 7a 3d 32 36	55881; _ utmc=26				
0340	32 30 35 35 38 38 31 2e 31 35 31 30 35 31 34 39	2055881. 15105149				
0350	31 33 2e 32 39 2e 32 2e 75 74 6d 63 73 72 3d 67	13.29.2. utmcsc=g				
0360	6f 6f 6f 6c 65 7c 75 74 6d 63 63 6e 3d 28 6f 72	oogle ut mcn=(or				
0370	67 61 6e 69 63 29 7c 75 74 6d 63 6d 64 3d 6f 72	ganic) u tmcnd=or				
0380	67 61 6e 69 63 7c 75 74 6d 63 74 72 3d 28 6e 6f	ganic ut mctr=(no				
0390	74 25 32 30 70 72 6f 76 69 64 65 64 29 0d 0a 0d	t%20prov ided)...				
03a0	0a 6c 6f 67 69 6e 3d 31 30 31 34 32 36 36 36 39	.login=1 01426669				
03b0	26 73 65 6e 68 61 3d 74 65 73 74 65 31 32 33 26	&senha=t este123&				
03c0	62 74 6e 43 6f 6e 66 69 72 6d 61 72 3d 43 6f 6e	btnConfi rmar=Con				
03d0	66 69 72 6d 61 72	firmar				

Fonte: Autor (2017).

A primeira coisa que podemos observar é que este site não contém segurança como já vimos no tópico “2.2 – Capturando senhas”, facilmente podemos observar o campo *login* com o valor 101426669 e o campo senha com o valor teste123. Isto também pode ser observado na Figura 10, localizada nos “detalhes do cabeçalho do pacote selecionado” na aba *HTML Form URL Encoded*.

FIGURA 10 – Detalhes do pacote.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)

> Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0  
 > Ethernet II, Src: AsustekC\_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT\_98:3a:88 (84:16:f9:98:3a:88)  
 > Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4  
 > Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928  
 > Hypertext Transfer Protocol  
 > HTML Form URL Encoded: application/x-www-form-urlencoded  
 > Form item: "login" = "101426669"  
 > Form item: "senha" = "teste123"  
 > Form item: "btnConfirmar" = "Confirmar"

Fonte: Autor (2017).

A próxima camada a ser analisada, é a camada de aplicação na aba *Hypertext Transfer Protocol* mostrado na Figura 11. Esta aba traz algumas informações como o protocolo utilizado (no caso da Figura 11 o HTTP), o tipo (POST ou GET (entre outros)), algumas *tags* HTML, o site acessado, a sua linguagem e o tamanho do dado em *bytes*.

FIGURA 11 - *Hypertext Transfer Protocol*.

> POST /Login/Logar HTTP/1.1\r\n Host: www.unoeste.br\r\n Connection: keep-alive\r\n Content-Length: 53\r\n Cache-Control: max-age=0\r\n Origin: http://www.unoeste.br\r\n Upgrade-Insecure-Requests: 1\r\n Content-Type: application/x-www-form-urlencoded\r\n User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36\r\n Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n Referer: http://www.unoeste.br/Login/Index?raMat=101426669&p=131549898115029950\r\n Accept-Encoding: gzip, deflate\r\n Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4\r\n [truncated]Cookie: __utmt_UA-1555105-1=1; __utma=262055881.1056147926.1509572105.1510513627.1510514913.29; __utmb=262055881.7.10\r\n <a href="http://www.unoeste.br/Login/Logar">[Full request URI: http://www.unoeste.br/Login/Logar]</a> [HTTP request 4/5] <a href="#">[Prev request in frame: 1242]</a> <a href="#">[Response in frame: 1302]</a> <a href="#">[Next request in frame: 1303]</a> File Data: 53 bytes
--

Fonte: Autor (2017).

A Figura 12 mostra a camada de transporte. Além de especificar o protocolo utilizado (o TCP como já mostra no nome da aba *Transmission Control Protocol*) ele especifica as portas de origem e destino, seu ACK e tamanho, *checksum* e *flags* além de outras informações complementares.

FIGURA 12 – Aba *Transmission Control Protocol*.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)

>	Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0
>	Ethernet II, Src: AsustekC_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT_98:3a:88 (84:16:f9:98:3a:88)
>	Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4
▼	Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928
	Source Port: 54393
	Destination Port: 80
	[Stream index: 10]
	[TCP Segment Len: 928]
	Sequence number: 2084 (relative sequence number)
	[Next sequence number: 3012 (relative sequence number)]
	Acknowledgment number: 16694 (relative ack number)
	0101 .... = Header Length: 20 bytes (5)
>	Flags: 0x018 (PSH, ACK)
	Window size value: 258
	[Calculated window size: 66048]
	[Window size scaling factor: 256]
	Checksum: 0xfcfc2 [unverified]
	[Checksum Status: Unverified]
	Urgent pointer: 0
>	[SEQ/ACK analysis]
	TCP payload (928 bytes)
>	Hypertext Transfer Protocol
>	HTML Form URL Encoded: application/x-www-form-urlencoded

Fonte: Autor (2017).

A aba *Internet Protocol Version 4*, referente a camada de rede, podemos facilmente ver o protocolo utilizado o IPV4, o IP de origem (minha máquina) e destino (Servidor da Unoeste) da requisição, algumas *flags*, *offset* e os respectivos tamanhos dos dados em *bytes* exibidos na Figura 13.

FIGURA 13 – Dados da camada de rede.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)

>	Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0
>	Ethernet II, Src: AsustekC_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT_98:3a:88 (84:16:f9:98:3a:88)
▼	Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4
	0100 .... = Version: 4
	.... 0101 = Header Length: 20 bytes (5)
▼	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
	0000 00.. = Differentiated Services Codepoint: Default (0)
	.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
	Total Length: 968
	Identification: 0x62df (25311)
▼	Flags: 0x02 (Don't Fragment)
	0... .... = Reserved bit: Not set
	.1.. .... = Don't fragment: Set
	..0. .... = More fragments: Not set
	Fragment offset: 0
	Time to live: 128
	Protocol: TCP (6)
	Header checksum: 0x00b9 [validation disabled]
	[Header checksum status: Unverified]
	Source: 192.168.0.104
	Destination: 177.131.33.4
	[Source GeoIP: Unknown]
	[Destination GeoIP: Unknown]
>	Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928
>	Hypertext Transfer Protocol
>	HTML Form URL Encoded: application/x-www-form-urlencoded

Fonte: Autor (2017).

Por ultimo a camada de enlace na aba denominada de *Ethernet II*, onde pode se notar os endereços MAC das máquinas e seus respectivos roteadores /

comutadores, tanto de origem quanto destino todos em hexadecimal como mostra a Figura 14. E na aba *Frame* como mostra a Figura 15 são exibidas as interfaces, tamanho dos dados e tempos, alguns protocolos e também portas utilizadas podendo fazer uma analogia com a camada física.

FIGURA 14 – Detalhes dos MAC.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)

> Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0

▼ Ethernet II, Src: AsustekC\_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT\_98:3a:88 (84:16:f9:98:3a:88)

    Destination: Tp-LinkT\_98:3a:88 (84:16:f9:98:3a:88)

        Address: Tp-LinkT\_98:3a:88 (84:16:f9:98:3a:88)

            .... 0. .... = LG bit: Globally unique address (factory default)

            .... 0. .... = IG bit: Individual address (unicast)

    Source: AsustekC\_eb:46:66 (34:97:f6:eb:46:66)

        Address: AsustekC\_eb:46:66 (34:97:f6:eb:46:66)

            .... 0. .... = LG bit: Globally unique address (factory default)

            .... 0. .... = IG bit: Individual address (unicast)

        Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4

> Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928

> Hypertext Transfer Protocol

> HTML Form URL Encoded: application/x-www-form-urlencoded

Fonte: Autor (2017).

FIGURA 15 – Detalhes dos *frames*.

No.	Time	Source	Destination	Protocol	Length	Info
1250	23.894568	177.131.33.4	192.168.0.104	HTTP	1367	HTTP/1.1 200 OK (text/html)
1300	29.878681	192.168.0.104	177.131.33.4	HTTP	982	POST /Login/Logar HTTP/1.1 (application/x-www-form-urlencoded)
1302	30.000166	177.131.33.4	192.168.0.104	HTTP	594	HTTP/1.1 302 Found (text/html)

▼ Frame 1300: 982 bytes on wire (7856 bits), 982 bytes captured (7856 bits) on interface 0

    Interface id: 0 (\Device\NPF\_{2489421F-3272-43C2-B816-6E93A9B6756C})

        Interface name: \Device\NPF\_{2489421F-3272-43C2-B816-6E93A9B6756C}

        Encapsulation type: Ethernet (1)

        Arrival Time: Nov 12, 2017 17:50:16.672912000 Horário brasileiro de verão

        [Time shift for this packet: 0.00000000 seconds]

        Epoch Time: 1510516216.672912000 seconds

        [Time delta from previous captured frame: 1.192189000 seconds]

        [Time delta from previous displayed frame: 1.192189000 seconds]

        [Time since reference or first frame: 29.878681000 seconds]

        Frame Number: 1300

        Frame Length: 982 bytes (7856 bits)

        Capture Length: 982 bytes (7856 bits)

        [Frame is marked: False]

        [Frame is ignored: False]

        [Protocols in frame: eth:ethertype:ip:tcp:http:urlencoded-form]

        [Coloring Rule String: http || tcp.port == 80 || http2]

> Ethernet II, Src: AsustekC\_eb:46:66 (34:97:f6:eb:46:66), Dst: Tp-LinkT\_98:3a:88 (84:16:f9:98:3a:88)

> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 177.131.33.4

> Transmission Control Protocol, Src Port: 54393, Dst Port: 80, Seq: 2084, Ack: 16694, Len: 928

> Hypertext Transfer Protocol

> HTML Form URL Encoded: application/x-www-form-urlencoded

Fonte: Autor (2017).

### 3 – CONCLUSÃO

Esta pesquisa abordou um pouco sobre a ferramenta de escaneamento de redes *Wireshark*, trazendo alguns exemplos de sites sem segurança e como a ferramenta tratou isto. Em meio ao mundo quase totalmente digital, e com um fluxo de informação exorbitante passando todos os dias nas redes de computadores espalhadas pelo mundo, vimos que a segurança é algo indispensável, pois os dados certos nas mãos de pessoas erradas causam um grande estrago.

Utilizando a ferramenta observamos o poder de detalhamento que ela traz dentro dela, muito bem organizada, com o mínimo de falhas possível e muito fácil de utilizar. A pesquisa abordou alguns tópicos simples, mas foi o suficiente para se observar o quão completo é este programa.

## REFERÊNCIAS

KUROSE, J. F; ROSS, K. W. **Redes de computadores e a internet**: uma abordagem top-down.6.ed. São Paulo: Pearson Education do Brasil, 2013.

TANENBAUM, A. S. **Redes de computadores**. 4.ed. Rio de Janeiro: Elsevier Editora, 2003.

DIEGO MACÊDO – ANALISTA DE T.I. **Introdução ao Wireshark**: Detecção e captura de tráfego em redes. 2016. Disponível em <<http://www.diegomacedo.com.br/introducao-ao-wireshark-deteccao-e-captura-de-trafego-em-redes/>>. Acessado em: 11 Nov. 2017.

LOGANDO TI. **Scanners de rede (software)**: Portas e vulnerabilidades. 2011. Disponível em: <<http://www.logandoti.com/scanners-de-rede-software-portas-e-vulnerabilidades/>>. Acessado em: 11 Nov. 2017.

UNDER-LINUX.ORG. **Wireshark – Parte 1 – Análise de Tráfego e Captura de Senhas**. 2012. Disponível em: < <https://under-linux.org/entry.php?b=2969>>. Acessado em: 11 Nov. 2017.

WIRESHARK. **Wireshark**. 2017. Disponível em: <<https://www.wireshark.org/>>. Acessado em: 11 Nov. 2017.