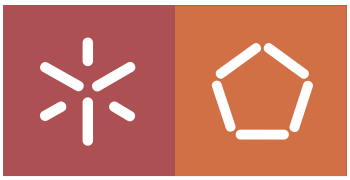
83%
Trabalho muito bom. Assinalo apenas algumas lacunas: detalhes da análise do tráfego não desejado está um pouco limitada porque é feita apenas na perspetiva de sessões (-15%); a análise estatistica indicada não apresenta todas as avidências (-2%).



Universidade do Minho
Escola de Engenharia

GESTÃO E VIRTUALIZAÇÃO DE REDES
SEGURANÇA EM REDES
NETWORK SECURITY
(SR) - HOMEWORK TP4

ANÁLISE DE TRÁFEGO

GRUPO 2

A85308 Filipe Miguel Teixeira Freitas Guimarães
A79799 Gonçalo Nogueira Costeira
A84912 Joana Isabel Afonso Gomes
A75480 Marco Matias Pereira Gonçalves
A42040 Miriam Miranda Pinto
A57041 Simão Pedro Santa Cruz Oliveira

Braga,
30 de novembro de 2020

Conteúdo

1 Introdução e Contextualização

Neste trabalho prático foi-nos proposta a aplicação do conhecimento adquirido nas aulas de Seguranças em Redes relativamente ao tópico de **Análise de Tráfego**, sendo o objectivo realizar isso mesmo, uma análise de tráfico, paralelamente identificando os componentes presentes (e as suas características) na captura dada.

Para esta análise foi precisa uma alguma contextualização teórica com os conceitos que estão interligados a este tema, passando pelo **tráfego** de rede, as várias **camadas de rede** do **modelo OSI**, e os diversos protocolos que nelas existem.

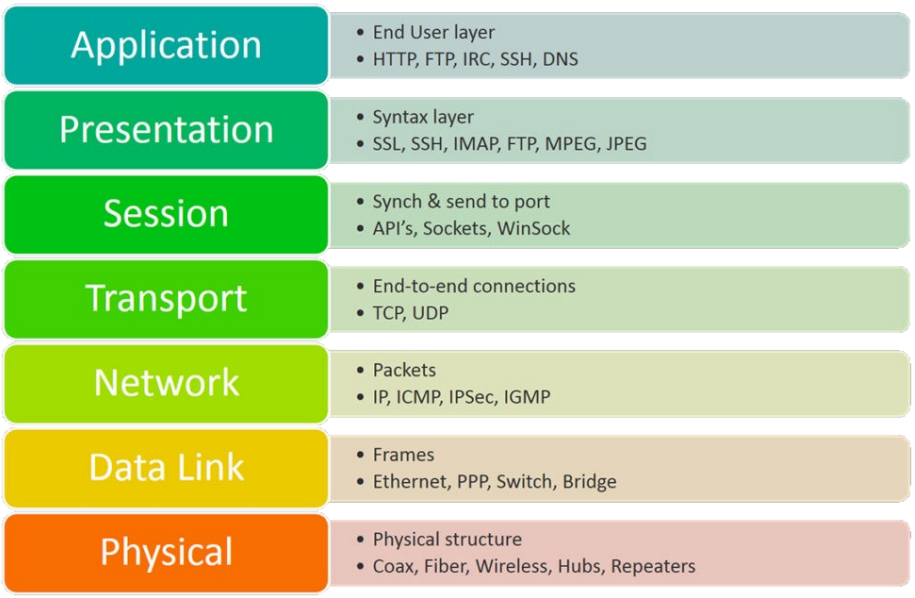


Figura 1: Camadas de rede do modelo OSI e alguns dos seus protocolos.
Fonte: `https://asmed.com/comptia-network-osi-model/`

Simultaneamente, foi necessário uma familiarização pelo grupo com as várias funcionalidades que o Wireshark oferece, recorrendo essencialmente à sua documentação.

Analisámos portanto a captura de tráfego que nos foi disponibilizada, registando várias vertentes do tráfego, recorrendo essencialmente às funcionalidades de estatística do programa, levando à elaboração das tarefas propostas.

2 Resposta ao problema proposto

2.1 Home net = 193.137.8.0/24

Teremos como endereço de rede interna o endereço 193.137.8.0/24, ou seja internamente serão usados os ultimos 8 bits para endereçamento de hosts.
Podemos verificar os seguintes endereços da rede local presentes na captura:

- 193.137.8.95
- 193.137.8.106
- 193.137.8.114
- 193.137.8.138
- 193.137.8.142
- 193.137.8.157
- 193.137.8.215

Usando o filtro `ip.addr == 193.137.8.0/24` podemos observar apenas os pacotes que circulam com endereços da Home net.

Através da **Hierarquia de Protocolos** do Wireshark, conseguimos observar que a interface capturada foi a ***Ethernet***, como podemos ver na Figura 2.

O que querem dizer com isto?

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	563	100.0	185889	9731	0	0	0
Ethernet	100.0	563	4.2	7882	412	0	0	0
Internet Protocol Version 4	99.1	558	6.0	11160	584	0	0	0
User Datagram Protocol	3.2	18	0.1	144	7	0	0	0
NetBIOS Name Service	0.2	1	0.0	62	3	1	62	3
Domain Name System	0.4	2	0.0	90	4	2	90	4
Data	2.7	15	0.7	1361	71	15	1361	71
Transmission Control Protocol	94.3	531	88.2	163936	8581	333	73290	3836
Telnet	5.3	30	0.2	287	15	30	287	15
NetBIOS Session Service	13.1	74	4.4	8117	424	0	0	0
SMB (Server Message Block Protocol)	13.1	74	4.2	7821	409	74	7821	409
Malformed Packet	0.2	1	0.0	0	0	1	0	0
Hypertext Transfer Protocol	15.5	87	54.1	100641	5268	75	73719	3859
Line-based text data	0.9	5	9.4	17551	918	5	18770	982
JPEG File Interchange Format	0.2	1	0.2	373	19	1	373	19
Data	0.4	2	1.5	2844	148	2	3220	168
Compuserve GIF	0.7	4	1.6	3009	157	4	3259	170
File Transfer Protocol (FTP)	0.9	5	0.1	136	7	5	0	0
Data	0.2	1	0.0	1	0	1	1	0
Internet Control Message Protocol	1.6	9	0.3	624	32	9	624	32
Configuration Test Protocol (loopback)	0.4	2	0.0	92	4	0	0	0
Data	0.4	2	0.0	80	4	2	80	4
Address Resolution Protocol	0.5	3	0.1	120	6	3	120	6

Figura 2: Protocol Hierarchy do Wireshark

2.2 Estratégia de análise

1) Primeiramente, com recurso à opção ***Endpoints*** das *Statistics* do Wireshark, conseguimos identificar os vários IPs usados na captura.

Era bom terem incluido as evidências que recolheram, juntamente com conclusões/observações.

2) Com a opção ***Conversations***, também de *Statistics*, é possível analisar o **tráfego TCP**

2.1) Ordenamos as *frames* por ordem temporal (carregando na opção *Rel. Start*, i.e., relative start)

2.2) Estudamos o conteúdo de cada *stream* (recorrendo à ferramenta *Follow Stream*)

3) Ainda com a mesma opção, é possível analisar o **tráfego UDP**

3.1) Ordenamos as *frames* por ordem temporal

3.2) Estudamos o conteúdo de cada *stream* (recorrendo à ferramenta *Follow Stream*)

4) Analisar outros protocolos, não TCP ou UDP, com auxílio de filtros do Wireshark.

2.3 Síntese de análise

2.3.1 Tráfego TCP

Nº sessão & streams	Tempo(s)	Src / Dest	Comentário
1 Streams 0 - 15	1.457307 a 3.90386 **Duração:** 2.4466	**Src:** 193.137.8.106 portas [1137-1153] **Dest:** 193.137.8.215 porta 80	Na stream 0 é estabelecida uma conexão TCP, resultando na criação de uma sessão HTTP. As 15 restantes streams presentes nesta primeira sessão correpondem a requests dos vários constituintes da página, para possibilitar a sua correta visualização. Ao longo desta conexão, podemos visualizar pacotes TCP DUP (TCP duplicate ACKs). Isto dá-se devido à receção de pacotes fora de ordem pelo server, ou quando há perda de pacotes. Foram transacionados 337 pacotes, perfazendo um total de 153.916k bytes
2 Stream 16	17.040244 a 82.494240 **Duração:** 65.4540	**Scr**: 193.137.8.106 porta 1153 **Dest:** 66.24.91.17 porta 80	Há uma sessão HTTP, entre o cliente e o servidor (mail.google.com), sendo observável o endereço de email com que o login foi efetuado (eu.nuno@gmail.com). (Cf. Anexo 1) Sendo HTTP (não HTTPs), é possível ver excertos das mensagens de e-mail trocadas, que não são devidamente encriptadas. (Cf. Anexo 1) No final da stream, é terminada a sesssão com um pedido [RST,ACK] (possivelmente devido a um timeout). (Cf. Anexo 2) Foram transacionados 9 pacotes de dados, perfazendo um total de 2842 bytes. ou fecho da janela

Nº sessão & streams	Tempo(s)	Src / Dest	Comentário
3 Stream 17	23.819398 a 35.186798 **Duração:** 11.3674	**Src:** 193.137.8.106 porta 1154 **Dest:** 193.137.8.95 porta 21	(Cf. Anexo 3) Há uma sessão FTP, entre o cliente e o servidor ("piano.dsi.uminho.pt"). Existe uma tentativa de autenticação com o user anonymous, mas essa tentativa é rejeitada pelo server com o código "530 - User anonymous unknown". Este utilizador anonymous no FTP é uma maneira de os utilizadores não terem de se identificar ao servidor, muito usado em servers públicos. O facto de o server ter rejeitado este login significa que já está protegido contra tentativas de ataque com este método. No fim o user fecha a ligação, o que é representado com o code 221. A duração desta stream justifica-se pelo tempo que o user demora a inserir os seus dados de autenticação e o tempo que demora até decidir sair. Foram transacionados 14 pacotes de dados, perfazendo um total de 918 bytes.
4 Stream 18	54.257406 a 82.846006 **Duração:** 28.5886	**Src:** 193.137.8.106 porta 1156 **Dest:** 193.137.8.95 porta 23	Há uma sessão TELNET entre o servidor e o cliente. Há uma tentativa de login com o user "guest"e a password "guest"na página "piano.dsi.uminho.pt" (mais uma vez podemos ver estes dados pois o protocolo não cobre a sua encriptação), sendo que estas credenciais são inválidas e a autenticação é negada. (Cf. Anexo 4) Foram transacionados 53 pacotes de dados, perfazendo um total de 3239 bytes.
5 Stream 19	97.001824 a 106.000724 **Duração:** 8.9989	**Src:** 87.28.58.222 porta 11132 **Dest:** 193.137.8.157 porta 30797	(Cf. Anexo 5) O host 87.28.58.222 tenta estabelecer uma ligação através do envio de pedidos SYN. Não havendo resposta, dão-se timeouts e são enviados pedidos de retransmissão (TCP retransmission). Foram transacionados 3 pacotes de dados, perfazendo um total de 186 bytes.
6 Stream 20	98.607890 a 107.60159 **Duração:** 8.9937	**Src:** 87.28.58.222 porta 11139 **Dest:** 193.137.8.157porta 443	Assim como na sessão anterior, o host 87.28.58.222 tenta estabelecer uma ligação através do envio de pedidos SYN, na porta 443 do server destino. Não havendo resposta, dão-se timeouts, e são enviados pedidos de retransmissão (TCP retransmission). Foram transacionados 3 pacotes de dados, perfazendo um total de 186 bytes.

Não é uma

E anteriormente (357, 358, 365) já tinha havido uma tentativa semelhante, de outra máquina! Não apanharam, porque tentaram analisar isto como stream.

idem

Nº sessão & streams	Tempo(s)	Src / Dest	Comentário
7 Stream 21	100.221796 a 109.203696 **Duração:** 8.9819	**Src:** 87.28.58.222 porta 11141 **Dest:** 193.137.8.157 porta 80	Tal como nas últimas duas sessões, o host 87.28.58.222 tenta estabelecer uma ligação através do envio de pedidos SYN, agora na porta 80 do server destino. Não havendo resposta, dão-se timeouts, e são enviados pedidos de retransmissão (TCP retransmission). Foram transacionados 3 pacotes de dados, perfazendo um total de 186 bytes.
8 Stream 22	137.534994 a 137.997794 **Duração:** 0.4628	**Src:** 193.137.8.106 porta 1157 **Dest:** 66.249.91.17 porta 80	Assim como na sessão 2 (stream 16), há uma sessão HTTP, entre o cliente e o servidor (mail.google.com), sendo observável o endereço de email com que o login foi efetuado (eu.nuno@gmail.com), em que é possível ver excertos das mensagens de e-mail trocadas. Porém, em constraste com a sessão 2, nesta não há um [RST,ACK] que traduz o fecho da sessão. (Cf. Anexo 6) (Isto pode dar aso a um ataque conhecido como *cookie hijacking* ou *session hijacking*, que se traduz na exploração de uma sessão de computador válida para obter acesso não autorizado a informações ou serviços do sistema) Foram transacionados 8 pacotes de dados perfazendo um total de 2788 bytes.
9 Stream 23	143.664551 a 152.818857 **Duração:** 9.1543	**Src:** 193.137.8.106 porta 1158 **Dest:** 193.137.8.142 porta 445	Há uma sessão SMB, que se trata de uma partilha de ficheiros, propósito mais comum deste protocolo. (Cf. Anexo 7) Se analisarmos atentamente a stream com o filtro tcp.stream do Wireshark, podemos detetar tentativas de login com user root ("\"), (nos pacotes 472 e 508), que são negadas pelo server. (Cf. Anexo 8) É ainda vísivel uma tentativa de login pelo user "BOCASJNR\hsantos"(pacote 480), que tentou aceder ao ficheiro "\AutoRun.inf"(confrontar pacote 485) na diretoria "\\TROMBONE\SOFT", mas no pacote 485 vemos um erro "STATUS_OBJECT_NAME_NOT_FOUND", indicando que não teve sucesso. (Cf. Anexo 9) É possível ainda ver alguns pacotes perdidos com tentativas de retransmissão por TCP Retransmission. Foram transacionados 98 pacotes de dados perfazendo um total de 17k bytes.
10 Stream 24	143.676901 a 143.721001 **Duração:** 0.0441	**Src:** 193.137.8.106 porta 1159 **Dest:** 193.137.8.142 porta 139	(Cf. Anexo 10) Há uma tentativa de estabelecimento de ligação mas é terminada com um pacote RST (TCP Reset). RST acontece quando chega um pacote TCP não esperado a um host, o que quase sempre indica uma atividade maliciosa, como tentativas de hijack à conexão. Foram transacionados 3 pacotes de dados, perfazendo um total de 178 bytes.

ou simplesmente porque a captura

2.3.2 Tráfego UDP

Nº sessão & streams	Tempo(s)	Src / Dest	Comentário
0	23.779650 a 23.792078 **Duração:** 0.0124	Src: 193.137.8.106 porta 1030 Dest: 193.137.8.142 porta 53	É Pedido de DNS referente à tradução do endereço piano.dsi.uminho.pt para o seu IP. Teve como resposta do servidor de DNS da Universidade do Minho (IP) o IP 192.137.8.95. Este pedido surgiu quando na stream TCP 17 é efetuada uma ligação FTP. Foram transacionados 3 pacotes de dados, perfazendo um total de 450 bytes.
7	143.672084 **Duração:** 0.0000	Src: 193.137.8.106 porta 137 Dest: 193.137.8.142 porta 137	Este pacote é referente à stream TCP 24, ou seja, ao serviço SMB e tem como propósito fazer o mesmo trabalho que um serviço de DNS (traduzir de nome para IP), mas do protocolo NBNS, que pertence ao serviço NetBIOS.

As streams 1-6 não apresentam informação relevante ou quaisquer dados decifráveis, para podermos analisar e tirar conclusões. Porém, no decorrer dessas sessões, foram transacionados 15 pacotes de dados, perfazendo um total de 1991 bytes.

2.3.3 Outros protocolos

Analisamos de seguida protocolos com um reduzido tráfego nesta captura, quando comparados com o TCP e o UDP.

- **ARP**

  Este protocolo existe nos pacotes 2, 349 e 351. O ARP (**Address Resolution Protocol**) é usado para descobrir endereços da camada de Ligação de Dados, tais como endereços MAC. Na Figura 3 apresentamos um exemplo para a descoberta do endereço MAC, neste caso no pacote 2.

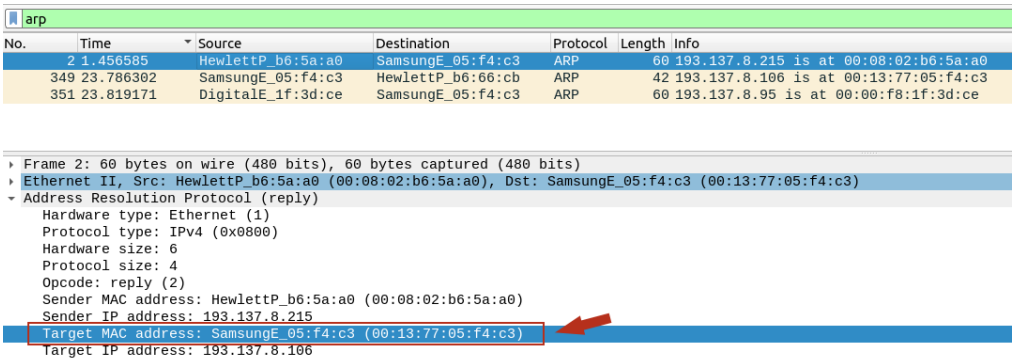


Figura 3: Pacotes com protocolo ARP

- **ICMP**

  Ao observarmos todos os pacotes **ICMP** através do filtro `icmp` podemos observar que estes pacotes representam *ping's*. Neste caso apenas encontramos 9 pacotes (1,6% da totalidade), em que destes 9 apenas 1 é um pacote de resposta (No.460) a um *ping request* (No.459) ou seja apenas temos uma resposta de 12,5% dos *ping's*

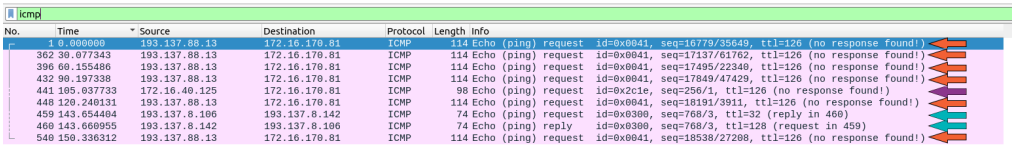


Figura 4: Pacotes com protocolo ICMP

* Vários Ping Request de 193.137.88.13 para 172.16.170.81 para testar a conectividade, sem obtenção de qualquer resposta.

* Ping Request para testar a conectividade noutro endereços, sem obtenção de qualquer resposta.

* **Conexão bem sucedida.** Efetuou-se um Ping Request, obteve-se um Ping Reply.

- **FTP**

Ao observarmos todos os pacotes **FTP** através do filtro `ftp` reparamos que estes apenas foram utilizados para o IP `193.137.8.95` aceder ao `piano.dsi.uminho.pt`. Como se trata de uma comunicação não cifrada podemos observar que o utilizador tenta intoduzir um *User* **anonymous** mas recebe a resposta a dizer que este *User* não existe e decide terminar a sessão.

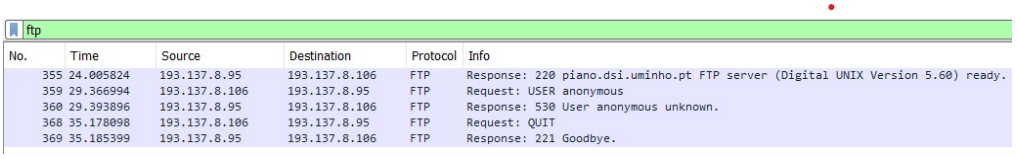


Figura 5: Pacotes com protocolo FTP

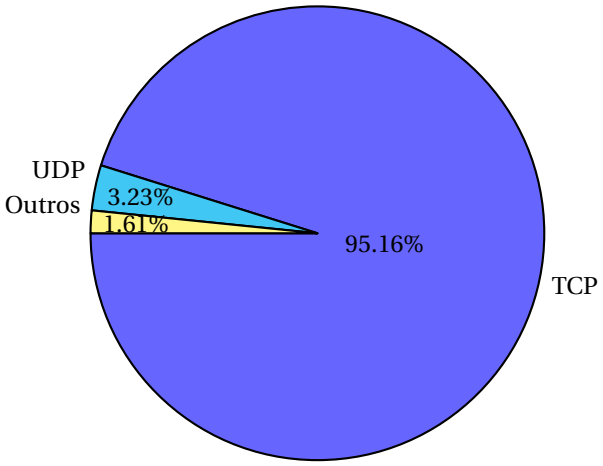
Mas não conseguiram uma visão sequencial e de flows (IPs) do tráfego "não desejado"...

3 Estatísticas

Através do Wireshark conseguimos recolher alguns dados importantes que nos permitiram fazer uma análise mais eficiente.

- **Data da captura:** 2007-10-16 17:36:28
- **Duração:** 00:02:32
- **Encapsulamento:** Ethernet
- **Número total de pacotes:** 563
- **Total de bytes:** 185889
- **Tamanho máximo de cada pacote:** 65535 bytes

Analisando os protocolos verificou-se que o protocolo predominante é o TCP, seguido pelo UDP e por fim outro tipo de protocolos. Tal como podemos ver no *pie chart* seguinte:



Durante a análise estatística achamos relevante perceber quais as portas utilizadas para a comunicação e obtivemos os seguintes resultados.

Para o protocolo TCP foram usadas as seguintes portas:

- **403:** Porta standard para comunicação HTTPS;
- **80:** Esta porta é usada maioritariamente para comunicação HTTP;
- **21:** Costuma estar associada a FTP, um protocolo de transferência de ficheiros não seguro, visto que a informação é enviada em texto;
- **23:** Tipicamente usada pelo protocolo Telnet;
- **445:** Usada por versões mais recentes do SMB para transmissão de ficheiros ou pelo Microsoft-DS *Active Directory*;
- **139:** Serviço de sessão da NetBIOS;
- **11132:** Sem informação relevante.

Para o protocolo UDP foram usadas as seguintes portas:

- **30797:** Sem informação relevante;
- **53:** Utilizada pelo DNS;
- **137:** Serviço de naming da NetBIOS, usado para registar nomes.

4 Conclusão e Análise de Resultados

Após mais um trabalho prático realizado, desta vez envolvendo análise de tráfego, foi possível, com recurso à útil ferramenta do Wireshark, analisar várias vertentes do tráfego, como por exemplo (um fator que consideramos muito relevante) potenciais vulnerabilidades (como acesso a email, password e outros dados), concluindo onde é provável ocorrerem ataques.

Felizmente o Wireshark apresenta bastante documentação, tornando mais acessível a sua utilização. Assim, consideramos que o grupo adquiriu competências diversificadas ao nível do tema abordado e, como referido pelo docente, outra "sensibilidade"na análise de tráfego, de forma a conseguir detectar intuitivamente onde pode ser comprometida a informação a ser transmitida.

5 Webgrafia

- https://whatis.techtarget.com/
- https://www.wireshark.org/docs
- https://owasp.org/www-community/attacks/

6 Anexos

```
GET /mail/?ui=pb&tlt=115a67ba1f3 HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; GNotify 1.0.25.0)
Host: mail.google.com
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: GV=10115a9980eb7-6dec08d9f1dfd9f8b6151b4afca9a9f4; __utma=173272373.870200234.1175466843.1175466843.1175466843.1; gmailchat=eu.nuno@gmail.com/104902;
S=gmail=2BOv8-GZ5SSMzZQu8Lkpwg:gmail_yj=7gECZkY1yCB029onOc6XoQ:gmproxy=jd3pMfLR7ic:gmproxy_yj=kKwdq9Ty1Gc:gmproxy_yj_sub=W8G-7fLAEk0;
PREF=ID=77a64c6088f3ea1c:TM=1167250258:LM=1167250258:S=USGBHPfj9i5dGP-M;
SID=DQAAAHkAAAAzEF93tOOV4RhNJywuq0A1qO5Kl5D7fy7oHDx6tbyp6nrUjce49LXvmTkdYgVSBhdtsnX0poppXnB9nv56CSWQ9q2mEV_nXu0ibW2phTgj3Rc6YXkRAlyGjUl1F-
ByEOqDnPr0zp16S8nAnj6oJLkIL-9XdD0ZpU7Fc2VkDtbnWQ

HTTP/1.1 200 OK
Cache-control: no-cache, no-store
Pragma: no-cache
Content-Type: application/octet-stream
Content-Length: 1422
Server: GFE/1.3
Date: Tue, 16 Oct 2007 16:36:34 GMT


.................."...^all...^i...^u..7
1
.membership@techtarget.com.au..SearchStorage ANZ.........ETape encryption; Exchange Backup; 4PB of tape; Google's storage plans..0SearchStorage ANZ : Weekly Site RoundUp
&hellip;...
.................."...^all...^i...^u..(
"
.program@mentornet.net.    MentorNet..........[MentorNet DS]: Interviews..eFor Match with Sam Bartels) Henrique, Giving a good interview is a skill that we often learn
&hellip;...
.................."...^all...^i...^u..4
.
.membership@techtarget.com.au..TechTarget ANZ.........HNew White Papers: Intelligent storage; Security risk reduction; and more..jTechTarget ANZ : White Paper Update
Welcome to TechTarget ANZ&#39;s White Paper Update, featuring &hellip;...
.................."...^all...^i...^u..8
2
.program.team@mentornet.net..MentorNet Program T...........MentorNet: Take Action..aDear Henrique, Here are some opportunities to take action with MentorNet: 1) Be an
early &hellip;...
.................."...^all...^i...^u..7
1
.membership@techtarget.com.au..SearchStorage ANZ.........Storage basics; Mozying along; Expand your SAN..0SearchStorage ANZ : Weekly Site RoundUp &hellip;...
.................."...^all...^i...^u..5
/
.jennifercg@mentornet.net..Jennifer Chou-Green.........7Share your opinions and experiences about careers in IT..jDear Henrique, For the November issue of MentorNet News.
we plan to feature careers in IT. Please &hellip;.....
```

Anexo 1

tcp.stream eq 16

No.	Time	Source	Destination	Protocol	Length	Info
340	17.040244	193.137.8.106	66.249.91.17	TCP	62	1153 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1260 SACK_PERM=1
341	17.112653	66.249.91.17	193.137.8.106	TCP	60	80 → 1153 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1260
342	17.112805	193.137.8.106	66.249.91.17	TCP	54	1153 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
343	17.113087	193.137.8.106	66.249.91.17	HTTP	780	GET /mail/?ui=pb&tlt=115a67ba1f3 HTTP/1.1
344	17.195859	66.249.91.17	193.137.8.106	TCP	60	80 → 1153 [ACK] Seq=1 Ack=727 Win=6786 Len=0
345	17.495952	66.249.91.17	193.137.8.106	TCP	1314	80 → 1153 [ACK] Seq=1 Ack=727 Win=6786 Len=1260 [TCP segment of a reassembled PDU]
346	17.496102	66.249.91.17	193.137.8.106	HTTP	404	HTTP/1.1 200 OK
347	17.496282	193.137.8.106	66.249.91.17	TCP	54	1153 → 80 [ACK] Seq=727 Ack=1611 Win=65535 Len=0
425	82.494246	193.137.8.106	66.249.91.17	TCP	54	1153 → 80 [RST, ACK] Seq=727 Ack=1611 Win=0 Len=0

Anexo 2

```
220 piano.dsi.uminho.pt FTP server (Digital UNIX Version 5.60) ready.
USER anonymous
530 User anonymous unknown.
QUIT
221 Goodbye.
```

Anexo 3

```
..... ..#..'..$.......... ..#..'..$..'.........P......'.......ANSI..........!..........!.........
.

.
Digital UNIX (piano.dsi.uminho.pt) (ttyp1)
.

.
.

....login: guest
guest
Password:guest

Login incorrect


Wait for login retry ...

Login incorrect
login: .^D..
```

Anexo 4

tcp.stream eq 19

No.	Time	Source	Destination	Protocol	Length	Info
435	97.001824	87.28.58.222	193.137.8.157	TCP	62	11132 → 30797 [SYN] Seq=0 Win=17520 Len=0 MSS=460 SACK_PERM=1
437	100.005686	87.28.58.222	193.137.8.157	TCP	62	[TCP Retransmission] 11132 → 30797 [SYN] Seq=0 Win=17520 Len=0 MSS=460 SACK_PERM=1
442	106.000754	87.28.58.222	193.137.8.157	TCP	62	[TCP Retransmission] 11132 → 30797 [SYN] Seq=0 Win=17520 Len=0 MSS=460 SACK_PERM=1

Anexo 5

tcp.stream eq 22

No.	Time	Source	Destination	Protocol	Length	Info
451	137.534994	193.137.8.106	66.249.91.17	TCP	62	1157 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1260 SACK_PERM=1
452	137.624306	66.249.91.17	193.137.8.106	TCP	60	80 → 1157 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1260
453	137.624498	193.137.8.106	66.249.91.17	TCP	54	1157 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
454	137.624683	193.137.8.106	66.249.91.17	HTTP	780	GET /mail/?ui=pb&tlt=115a67ba1f3 HTTP/1.1
455	137.718582	66.249.91.17	193.137.8.106	TCP	60	80 → 1157 [ACK] Seq=1 Ack=727 Win=6786 Len=0
456	137.994238	66.249.91.17	193.137.8.106	TCP	1314	80 → 1157 [ACK] Seq=1 Ack=727 Win=6786 Len=1260 [TCP segment of a reassembled PDU]
457	137.994275	66.249.91.17	193.137.8.106	HTTP	404	HTTP/1.1 200 OK
458	137.997816	193.137.8.106	66.249.91.17	TCP	54	1157 → 80 [ACK] Seq=727 Ack=1611 Win=65535 Len=0

[RST,ACK]

Anexo 6

472	143.797519	193.137.8.106	193.137.8.142	SMB	332	Session Setup AndX Request, NTLMSSP_AUTH, User: \
473	143.884862	193.137.8.142	193.137.8.106	SMB	184	Session Setup AndX Response
474	143.885291	193.137.8.106	193.137.8.142	SMB	140	Tree Connect AndX Request, Path: \\TROMBONE\IPC$
475	143.910381	193.137.8.142	193.137.8.106	SMB	93	Tree Connect AndX Response, Error: STATUS_ACCESS_DENIED

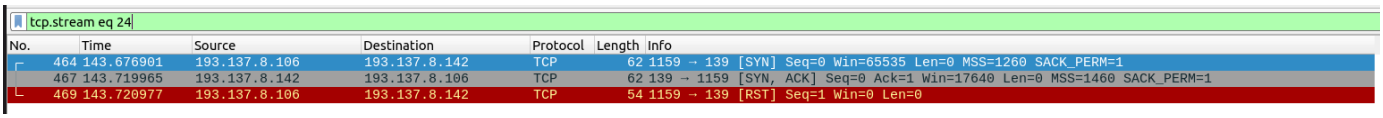
Anexo 7

480	143.997633	193.137.8.106	193.137.8.142	SMB	414	Session Setup AndX Request, NTLMSSP_AUTH, User: BOCASJNR\hsantos
481	144.088862	193.137.8.142	193.137.8.106	SMB	184	Session Setup AndX Response
482	144.089906	193.137.8.106	193.137.8.142	SMB	140	Tree Connect AndX Request, Path: \\TROMBONE\SOFT
483	144.105432	193.137.8.142	193.137.8.106	SMB	120	Tree Connect AndX Response
484	144.290936	193.137.8.106	193.137.8.142	SMB	158	Trans2 Request, QUERY_PATH_INFO, Query File Basic Info, Path: \AutoRun.inf
485	144.309540	193.137.8.142	193.137.8.106	SMB	93	Trans2 Response, QUERY_PATH_INFO, Error: STATUS_OBJECT_NAME_NOT_FOUND

Anexo 8

495	144.483434	193.137.8.142	193.137.8.106	TCP	1314	445 → 1158 [ACK] Seq=1879 Ack=1967 Win=17058 Len=1260 [TCP segment of a reassembled PDU]
496	144.483619	193.137.8.142	193.137.8.106	TCP	1314	[TCP Previous segment not captured] 445 → 1158 [ACK] Seq=4399 Ack=1967 Win=17058 Len=126…
497	144.483689	193.137.8.142	193.137.8.106	TCP	154	445 → 1158 [PSH, ACK] Seq=5659 Ack=1967 Win=17058 Len=100 [TCP segment of a reassembled …
498	144.483904	193.137.8.106	193.137.8.142	TCP	66	1158 → 445 [ACK] Seq=1967 Ack=3139 Win=65535 Len=0 SLE=4399 SRE=5659
499	144.483943	193.137.8.106	193.137.8.142	TCP	66	[TCP Dup ACK 498#1] 1158 → 445 [ACK] Seq=1967 Ack=3139 Win=65535 Len=0 SLE=4399 SRE=5759
500	144.773654	193.137.8.142	193.137.8.106	TCP	1314	[TCP Retransmission] 445 → 1158 [ACK] Seq=3139 Ack=1967 Win=17058 Len=1260
501	144.773874	193.137.8.106	193.137.8.142	TCP	54	1158 → 445 [ACK] Seq=1967 Ack=5759 Win=65535 Len=0

Anexo 9



tcp.stream eq 24

No.	Time	Source	Destination	Protocol	Length	Info
464	143.676901	193.137.8.106	193.137.8.142	TCP	62	1159 → 139 [SYN] Seq=0 Win=65535 Len=0 MSS=1260 SACK_PERM=1
467	143.719965	193.137.8.142	193.137.8.106	TCP	62	139 → 1159 [SYN, ACK] Seq=0 Ack=1 Win=17640 Len=0 MSS=1460 SACK_PERM=1
469	143.720977	193.137.8.106	193.137.8.142	TCP	54	1159 → 139 [RST] Seq=1 Win=0 Len=0

Anexo 10