

repositório do roteiro: <https://github.com/AndreCorreaSantos/Roteiro1.git>

Roteiro 1

1)

Intrusion detection systems (IDS) são ferramentas de segurança de internet usadas para monitorar tráfego e dispositivos contra atividades suspeitas e maliciosas envolvendo segurança da informação.

Exemplos de ferramentas que são utilizadas como IDS:

- ManageEngine Log360
- SolarWinds Security Event Manager

2)

A principal diferença entre um serviço de IDS e IPS está no fato de que o IDS apenas monitora o tráfego e pode alertar o usuário contra ameaças, já o IPS chega a filtrar e impedir a passagem de pacotes caso o serviço os considere suspeitos. Ambos os serviços utilizam bases de dados com assinaturas conhecidas de ameaças digitais para cumprir suas respectivas funções.

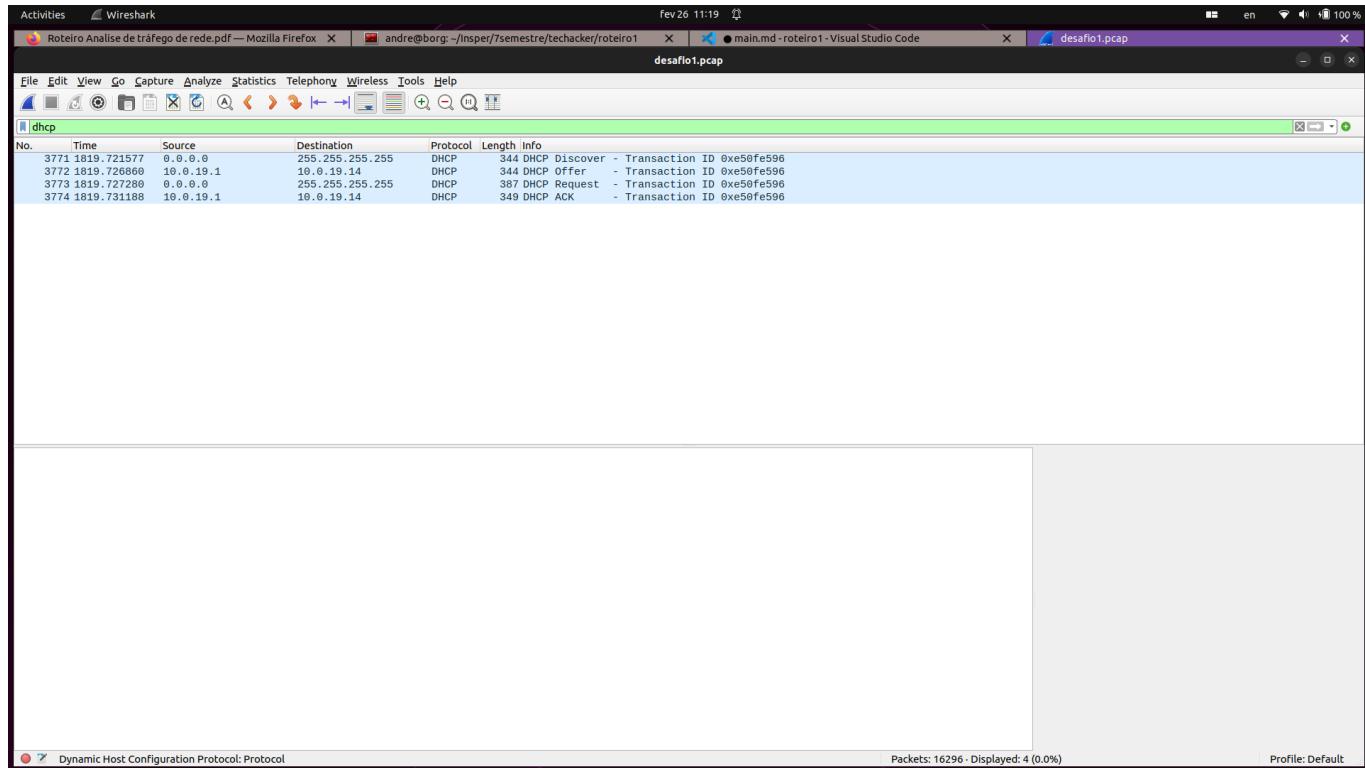
Desafio 1

3)

Utilizando Wireshark com display filter de "DHCP", podemos filtrar os pacotes e observamos quatro requests que pertencem ao protocolo DHCP: 3771 1819.721577 0.0.0.0 255.255.255.255 DHCP 344 DHCP Discover - Transaction ID 0xe50fe596 3772 1819.726860 10.0.19.1 10.0.19.14 DHCP 344 DHCP Offer - Transaction ID 0xe50fe596 3773 1819.727280 0.0.0.0 255.255.255.255 DHCP 387 DHCP Request - Transaction ID 0xe50fe596 3774 1819.731188 10.0.19.1 10.0.19.14 DHCP 349 DHCP ACK - Transaction ID 0xe50fe596

Observa-se que esses quatro pacotes seguem uma lógica tradicional onde o computador manda um request no endereço de broadcast da rede para descobrir o servidor responsável por implementar o DHCP, em seguida o servidor responde com uma oferta, o cliente requisita o IP e, em seguida, o IP é associado ao PC.

Conclui-se que o servidor que implementa DHCP na rede capturada possui endereço 10.0.19.1.



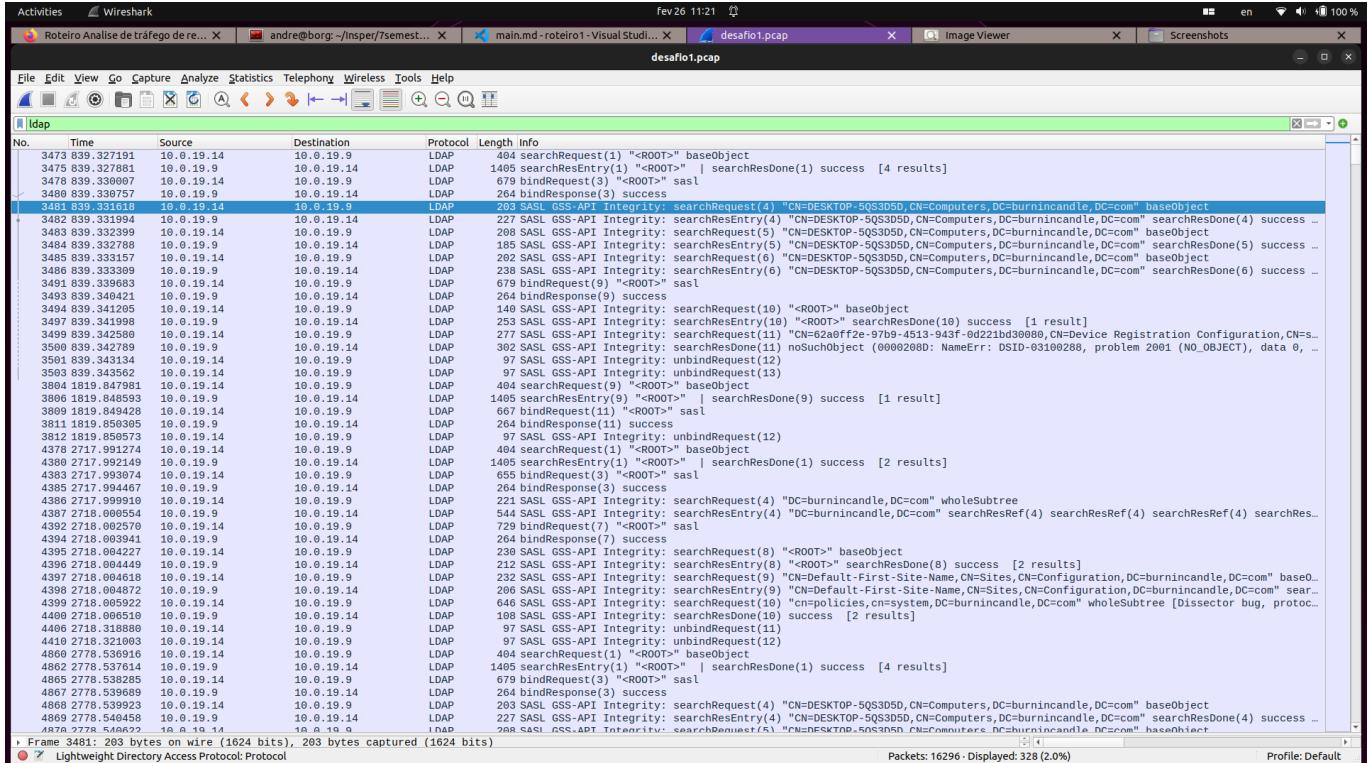
4)

A partir desse tutorial, <https://unit42.paloaltonetworks.com/using-wireshark-identifying-hosts-and-users/>, descobri que o controlador de domínio costuma utilizar o protocolo LDAP. Utilizando o display filter do Wireshark, filtrei os requests para apenas os que seguem o protocolo LDAP.

A grande maioria dos logs do protocolo LDAP que encontrei seguiam o seguinte padrão:

```
3481 839.331618 10.0.19.14 10.0.19.9 LDAP 203 SASL GSS-API Integrity: searchRequest(4) "CN=DESKTOP-5QS3D5D,CN=Computers,DC=burnincandle,DC=com" baseObject 3482 839.331994 10.0.19.9 10.0.19.14 LDAP 227 SASL GSS-API Integrity: searchResEntry(4) "CN=DESKTOP-5QS3D5D,CN=Computers,DC=burnincandle,DC=com" searchResDone(4) success [4 results]
```

Ou seja, o cliente de número 10.0.19.14 inicia a conversa e requisita algo para o servidor de IP 10.0.19.9, que responde com um resultado. Assim, é possível concluir que o controlador de domínio dessa rede possui IP 10.0.19.9, com nome burnincandle.



5)

Observando os requests, notei que o único computador que aparetava se comunicar com o domain controller era o computador de IP 10.0.19.14. Utilizando o display filter para filtrar todos os requests que chegam e saem do IP em questão e pertencem ao protocolo HTTP, notei que em certo momento essa máquina recebe um gzip.

498 24.690866 188.166.154.118 10.0.19.14 HTTP 678 HTTP/1.1 200 OK (application/gzip)

Além disso, a maioria dos outros requests identificados com esse filtro são relacionados a Windows updates, fora outros dois:

9272 7555.029469 104.80.96.219 10.0.19.14 HTTP 317 HTTP/1.1 304 Not Modified

Esse primeiro que vai para x1.c.lencr.org, que parece suspeito em um primeiro momento, contudo, mediante pesquisa, aparenta ser algo natural:

https://www.reddit.com/r/safing/comments/ra5t4j/question REGARDING_x1clencrorg_domain_found_in/.

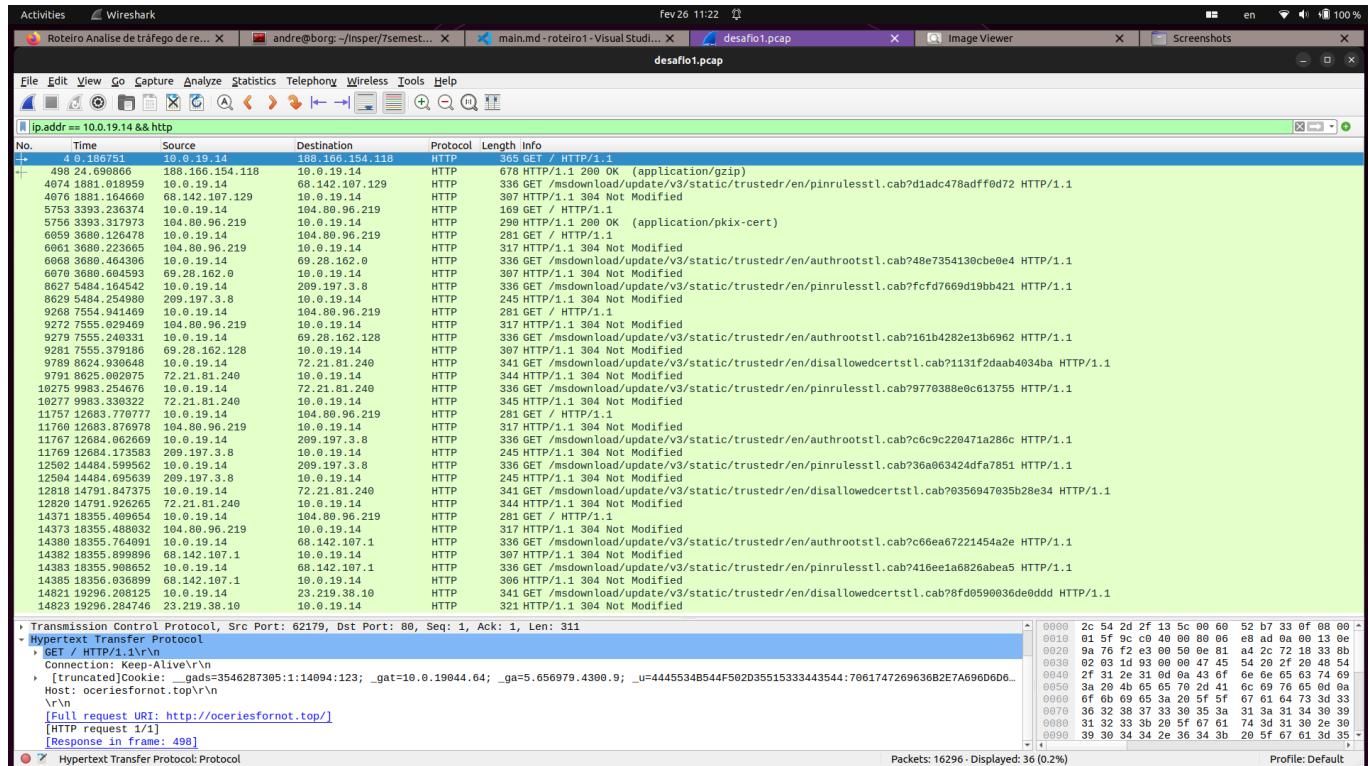
4 0.186751 10.0.19.14 188.166.154.118 HTTP 365 GET / HTTP/1.1

Esse segundo request, entretanto, vai para o domínio [Full request URI: http://oceriesfornot.top/].

Pesquisando sobre esse domínio, é fácil encontrar sua relação à atividades maliciosas:

<https://threatfox.abuse.ch/ioc/394377/>.

Voltando ao primeiro request que recebia um gzip, nota-se que o host que enviou esse arquivo é o mesmo host malicioso do segundo request; dessa forma, podemos supor que o cliente instalou sem querer algum arquivo malicioso desse endereço e agora o malware instalou-se na máquina e está mandando informações do cliente via HTTP/GET para o host malicioso.



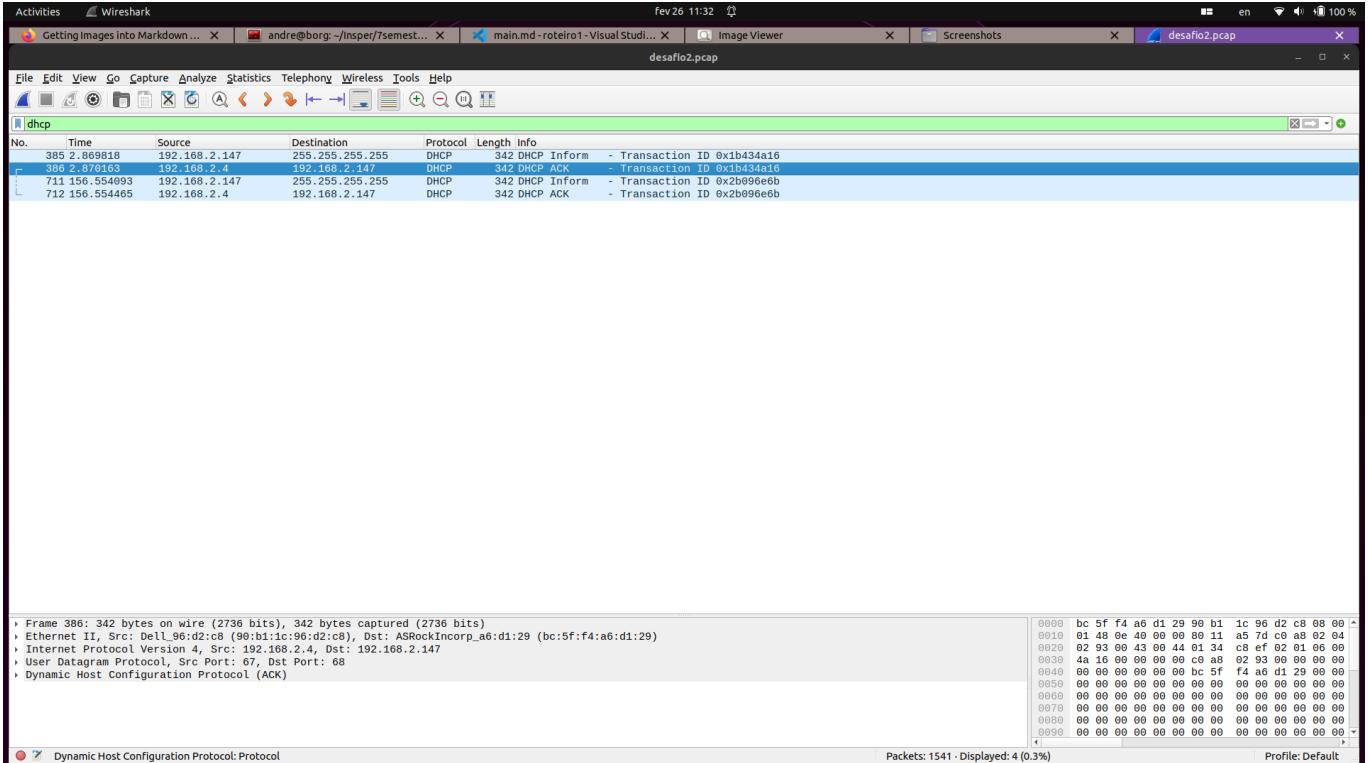
Desafio 2

6)

Para encontrar o MAC address do host 192.168.2.4, primeiramente tentei filtrar por requests do protocolo arp, entretanto não haviam requests desse protocolo nos logs. Em seguida filtrei por requests do protocolo dhcp e encontrei um DHCP ACK.

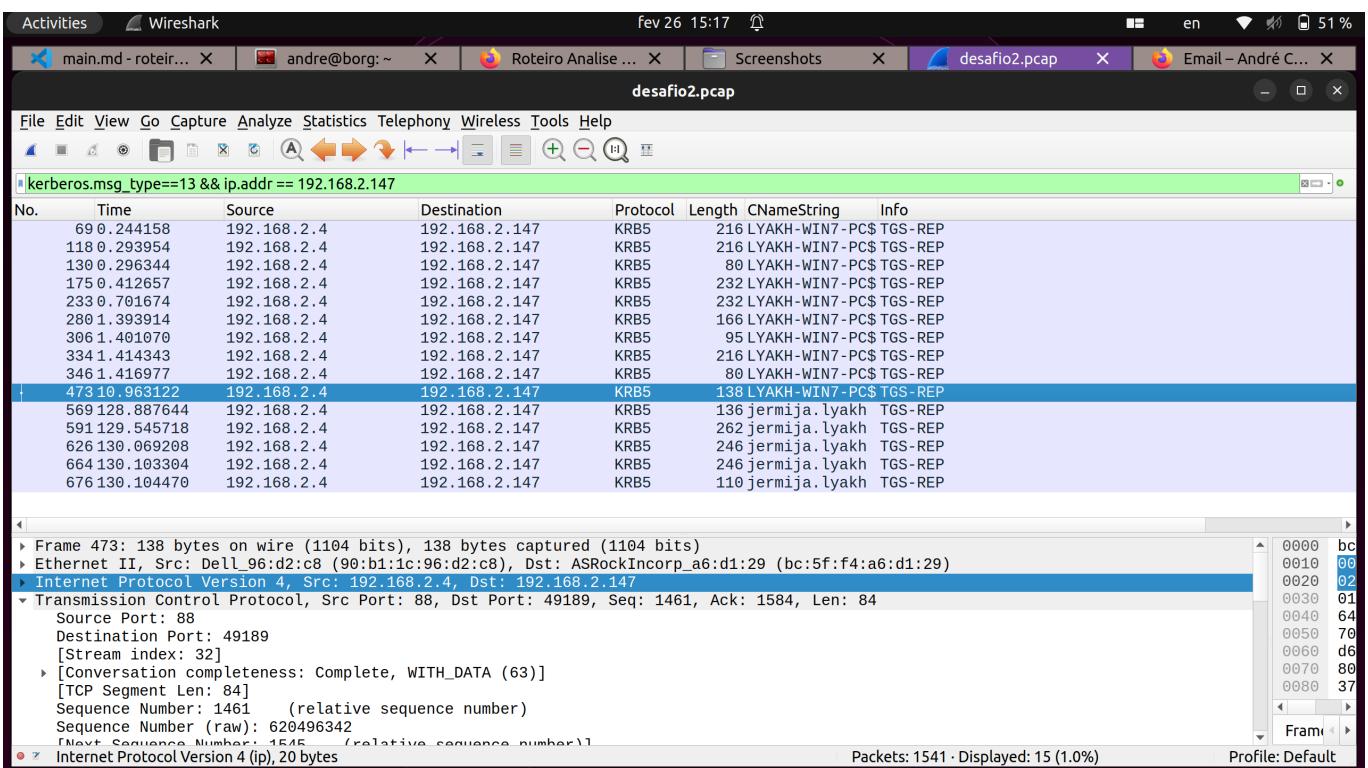
386 2.870163 192.168.2.4 192.168.2.147 DHCP 342 DHCP ACK - Transaction ID 0x1b434a16

no aba de Source do nível de Ethernet II do request foi possível identificar o MAC address da máquina em questão: Address: Dell_96:d2:c8 (90:b1:1c:96:d2:c8)--> MAC = 90:b1:1c:96:d2:c8.



7)

Para encontrar o hostname associado ao IP 192.168.2.147 utilizei o display filter com o filtro: `kerberos.msg_type==13 && ip.addr == 192.168.2.147`, para filtrar somente requests do tipo TGS-REP enviados para o ip de interesse. Com esse filtro ao investigar a variável cname de um dos requests, encontrei o nome do host associado ao IP. LYAKH-WIN7-PC\$



8)

Filtrando os requests pelo protocolo kerberos, reparei que existem quatro principais tipos de requests: AS-REQ, AS-REP, TGS-REQ e TGS-REP. pesquisando um pouco descobri que essas são etapas normais de autenticação por meio do kerberos. As etapas AS tem a ver com a conversa inicial entre o usuário e o servidor de autenticação e as etapas TGS tem a ver com a requisição de um ticket por parte do usuário e a resposta do servidor com o ticket do usuário. Em especial na etapa TGS-REP o servidor devolve um ticket para o usuário e nessa resposta, sob a variável cname - client name, é possível encontrar o nome do usuário windows em questão - essa mesma lógica foi utilizada na questão anterior.

CNameString: jermija.lyakh

para encontrar os requests do tipo TGS-REP, utilizei o display filter: `kerberos.msg_type==13 && ip.addr == 192.168.2.147`.

O filtro utilizado foi o mesmo da questão anterior.

| No. | Time | Source | Destination | Protocol | Length | CNameString | Info |
|------------------|------|-------------|---------------|----------|--------|-------------------------|------|
| 69 0. 244158 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 216 | LYAKH-WIN7-PC\$ TGS-REP | |
| 118 0. 293954 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 216 | LYAKH-WIN7-PC\$ TGS-REP | |
| 130 0. 296344 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 80 | LYAKH-WIN7-PC\$ TGS-REP | |
| 175 0. 412657 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 232 | LYAKH-WIN7-PC\$ TGS-REP | |
| 233 0. 701674 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 232 | LYAKH-WIN7-PC\$ TGS-REP | |
| 280 1. 393914 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 166 | LYAKH-WIN7-PC\$ TGS-REP | |
| 306 1. 401070 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 95 | LYAKH-WIN7-PC\$ TGS-REP | |
| 334 1. 414343 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 216 | LYAKH-WIN7-PC\$ TGS-REP | |
| 346 1. 416977 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 80 | LYAKH-WIN7-PC\$ TGS-REP | |
| + 473 10. 963122 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 138 | LYAKH-WIN7-PC\$ TGS-REP | |
| 569 128. 887644 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 136 | jermija.lyakh TGS-REP | |
| 591 129. 545718 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 262 | jermija.lyakh TGS-REP | |
| 626 130. 069208 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 246 | jermija.lyakh TGS-REP | |
| 664 130. 103304 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 246 | jermija.lyakh TGS-REP | |
| 676 130. 104470 | | 192.168.2.4 | 192.168.2.147 | KRB5 | 110 | jermija.lyakh TGS-REP | |

9)

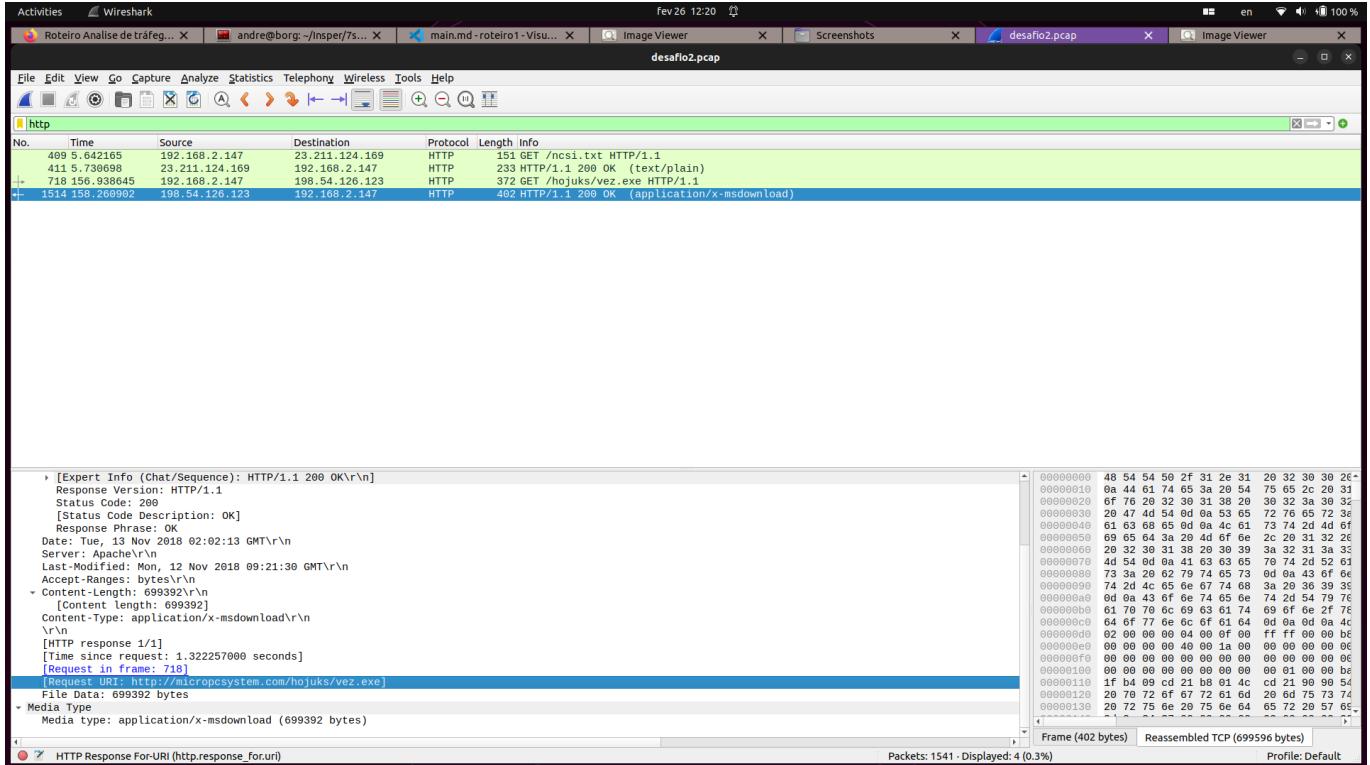
Kerberos é um protocolo de autenticação em redes de computadores baseado em tickets que permite que nós se comuniquem sobre uma rede não segura e comprovem sua identidade um ao outro de maneira segura. O protocolo foi desenvolvido pelo MIT e utiliza criptografia simétrica para autenticar usuários e serviços, evitando que senhas sejam enviadas sem criptografia pela rede.

10)

Filtrando os logs por "http" no display filter, encontram-se poucos http requests. Observando-os nota-se o último request apresenta uma URI de destino na qual há a requisição de um recurso com extensão .exe - que é um arquivo executável windows.

[Request URI: <http://micropcsystem.com/hojuks/vez.exe>]

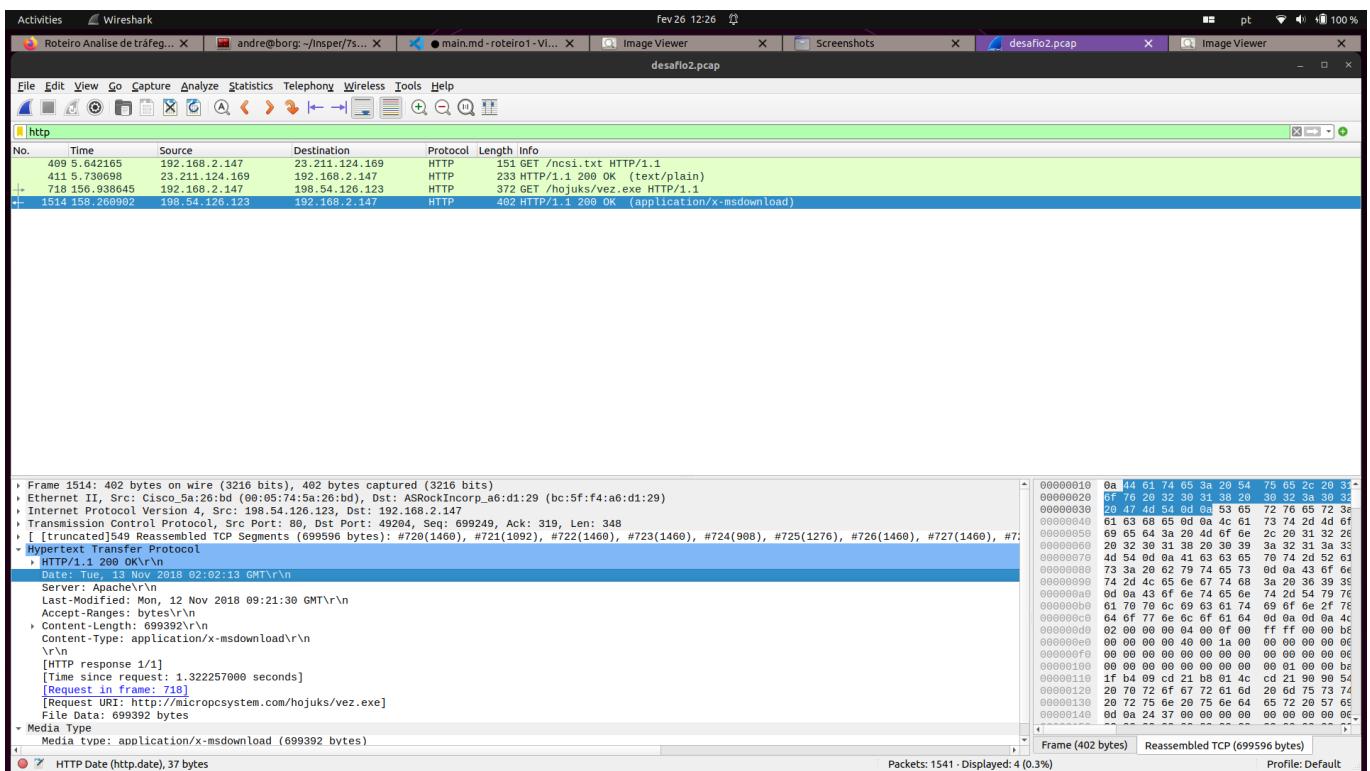
A url responsável pelo arquivo é: <http://micropcsystem.com/hojuks/vez.exe> e o nome do arquivo é: "vez.exe".



11)

Investigando o request da questão anterior, podemos encontrar a data e horário na Sessão "Hypertext Transfer Protocol":

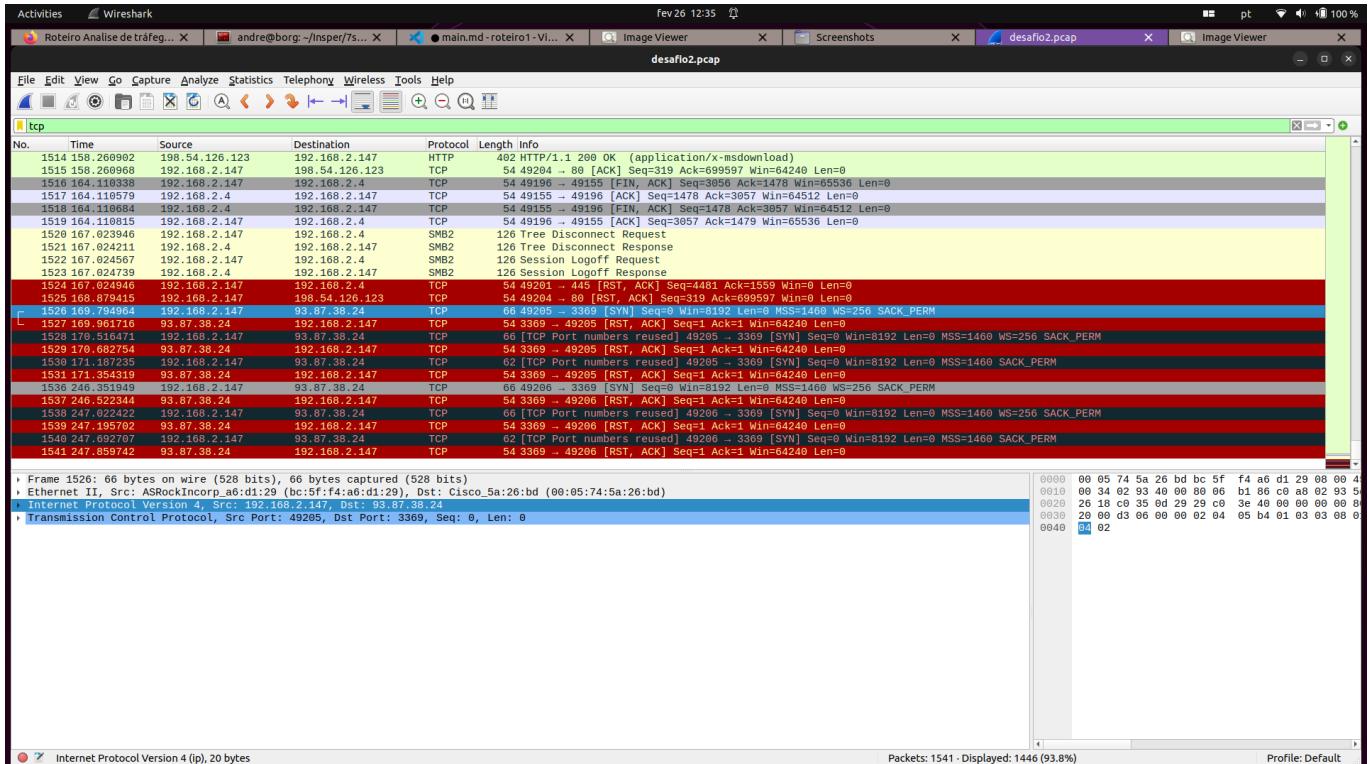
Date: Tue, 13 Nov 2018 02:02:13 GMT\r\nn



12)

Após receber o executável a máquina contaminada tenta estabelecer uma conexão TCP com 93.87.38.24, enviando um SYN para esse endereço.

1526 169.794964 192.168.2.147 93.87.38.24 TCP 66 49205 → 3369 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM

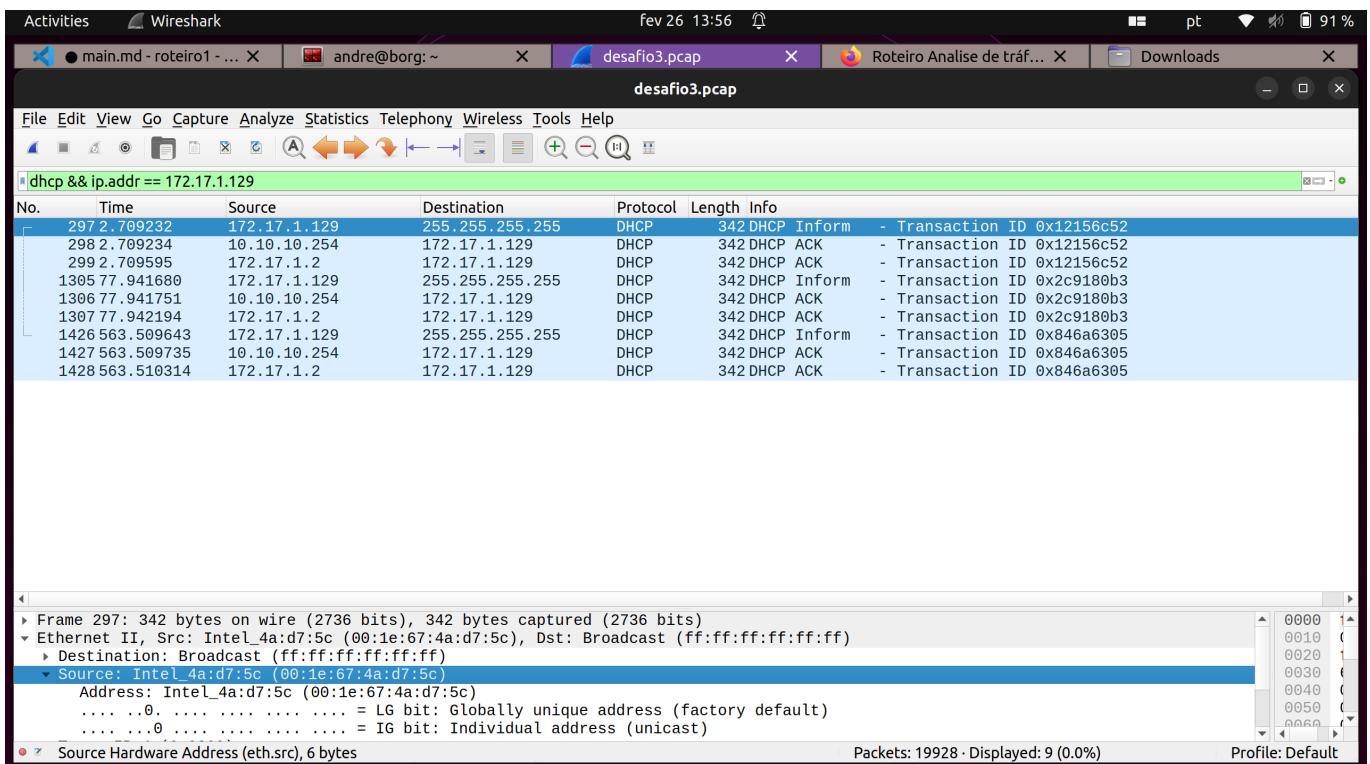


Desafio 3

13)

Filtrando os requests para apenas protocolo dhcp e ip.addr == 172.17.1.129, ao inspecionar um request que tem origem no ip de interesse é possível encontrar o MAC address na camada ETHERNET II do pacote.

Source: Intel_4a:d7:5c (00:1e:67:4a:d7:5c) --> MAC = 00:1e:67:4a:d7:5c



14)

Utilizando uma lógica similar à questão 8, filtrei os requests: `kerberos.msg_type == 13 && ip.dst == 172.17.1.129`, de modo a visualizar apenas pacotes do tipo TGS-REP enviados para o IP de interesse. Inspecionando um dos requests encontrados, é possível encontrar o nome do usuário associado ao IP sob o campo Cname.

CNameString: NALYVAIKO-PC\$

Nome do usuário: NALYVAIKO-PC\$

| No. | Time | Source | Destination | Protocol | Length | CNameString |
|------|-------------|------------|--------------|----------|--------|--------------------|
| 68 | 0.228168 | 172.17.1.2 | 172.17.1.129 | KRB5 | 214 | NALYVAIKO-PC\$ |
| 124 | 0.332099 | 172.17.1.2 | 172.17.1.129 | KRB5 | 214 | NALYVAIKO-PC\$ |
| 136 | 0.333533 | 172.17.1.2 | 172.17.1.129 | KRB5 | 78 | NALYVAIKO-PC\$ |
| 174 | 0.374457 | 172.17.1.2 | 172.17.1.129 | KRB5 | 230 | NALYVAIKO-PC\$ |
| 236 | 0.659168 | 172.17.1.2 | 172.17.1.129 | KRB5 | 230 | NALYVAIKO-PC\$ |
| 347 | 5.858052 | 172.17.1.2 | 172.17.1.129 | KRB5 | 162 | NALYVAIKO-PC\$ |
| 377 | 5.869203 | 172.17.1.2 | 172.17.1.129 | KRB5 | 91 | NALYVAIKO-PC\$ |
| 401 | 5.873197 | 172.17.1.2 | 172.17.1.129 | KRB5 | 214 | NALYVAIKO-PC\$ |
| 413 | 5.874679 | 172.17.1.2 | 172.17.1.129 | KRB5 | 78 | NALYVAIKO-PC\$ |
| 520 | 10.952912 | 172.17.1.2 | 172.17.1.129 | KRB5 | 134 | NALYVAIKO-PC\$ |
| 556 | 16.470573 | 172.17.1.2 | 172.17.1.129 | KRB5 | 176 | innochka.nalyvaiko |
| 570 | 16.535842 | 172.17.1.2 | 172.17.1.129 | KRB5 | 304 | innochka.nalyvaiko |
| 608 | 17.078991 | 172.17.1.2 | 172.17.1.129 | KRB5 | 288 | innochka.nalyvaiko |
| 644 | 17.101890 | 172.17.1.2 | 172.17.1.129 | KRB5 | 288 | innochka.nalyvaiko |
| 656 | 17.103204 | 172.17.1.2 | 172.17.1.129 | KRB5 | 152 | innochka.nalyvaiko |
| 8166 | 5767.006399 | 172.17.1.2 | 172.17.1.129 | KRB5 | 91 | NALYVAIKO-PC\$ |

Frame 556: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits)
 ▶ Ethernet II, Src: Dell_98:d5:f2 (f0:4d:a2:98:d5:f2), Dst: Intel_4a:d7:5c (00:1e:67:4a:d7:5c)
 ▶ Internet Protocol Version 4, Src: 172.17.1.2, Dst: 172.17.1.129
 ▶ Transmission Control Protocol, Src Port: 88, Dst Port: 49194, Seq: 1461, Ack: 1555, Len: 122
 ▶ [2 Reassembled TCP Segments (1582 bytes): #555(1460), #556(122)]
 ▶ Kerberos

15)

Utilizando o mesmo filtro da conta anterior, inspecionando alguns requests abaixo podemos encontrar o nome da conta de usuário do windows no ip de interesse:

nome: innochka.nalyvaiko

The screenshot shows the Wireshark interface with the file "desafio3.pcap" loaded. A search filter is applied: "kerberos.msg_type == 13 && ip.dst == 172.17.1.129". The list of captured frames shows multiple Kerberos messages (KRB5) between source IP 172.17.1.2 and destination IP 172.17.1.129. The details pane displays the structure of a Kerberos message frame, including fields like CNameString and a base64-encoded payload. The bytes pane shows the raw hex and ASCII data of the captured frame.

16)

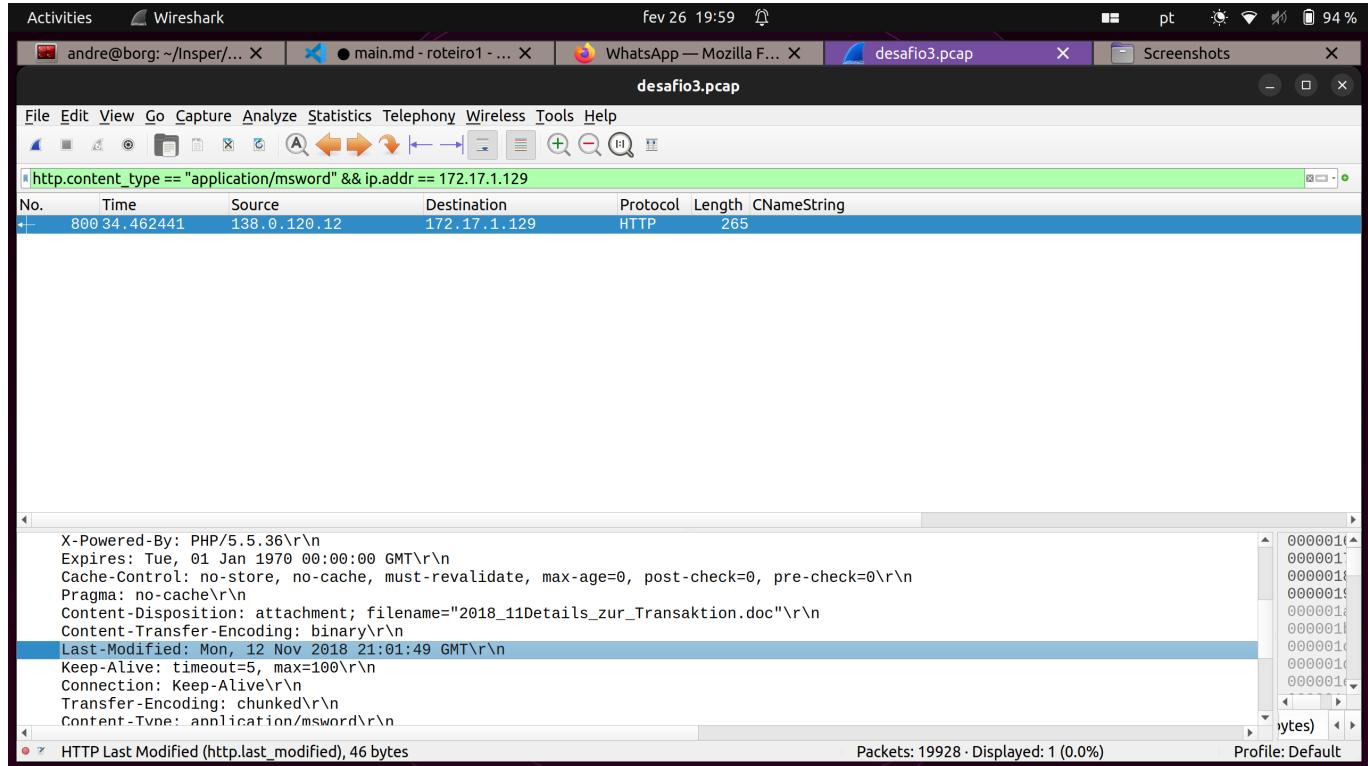
Filtrando os logs para apenas requests relacionados ao ip de interesse, no protocolo http e com content_type igual a application/msword, foi possível encontrar o request em que o url que retornou um documento word. <http://ifcingenieria.cl/QpX8It/BIZ/Firmenkunden/>

The screenshot shows the Wireshark interface with the file "desafio3.pcap" loaded. A search filter is applied: "http.content_type == \"application/msword\" && ip.addr == 172.17.1.129". The list of captured frames shows one single HTTP request (HTTP/1.1) from source IP 138.0.120.12 to destination IP 172.17.1.129, with a length of 265 bytes. The details pane shows the request headers, including Transfer-Encoding: chunked, Content-Type: application/msword, and the URL http://ifcingenieria.cl/QpX8It/BIZ/Firmenkunden/. The bytes pane shows the raw HTTP request message.

17)

Podemos ver a data e hora em que a URL foi criada no campo Last-Modified da sessão Hypertext Transfer Protocol.

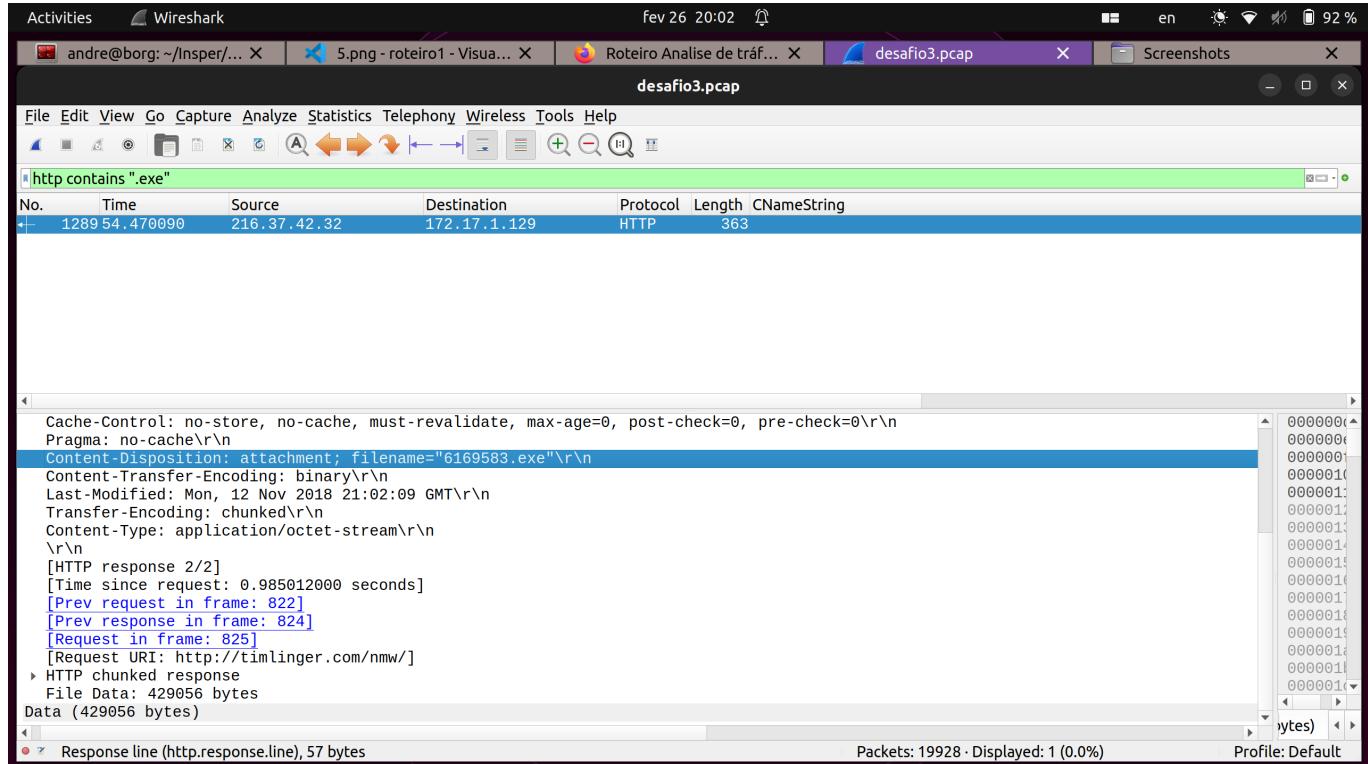
Last-Modified: Mon, 12 Nov 2018 21:01:49 GMT



18)

Filtrando todos requests http que contém .exe foi possível encontrar um request no qual o ip de interesse tenta baixar um arquivo executável.

URI: <http://timlinger.com/nmw/>



19)

Inspecionando os requests do protocolo kerberos, é possível observar que o ip 192.168.1.216 envia pedidos para 192.168.1.2 requisitando algo e o servidor responde com um ticket. A partir disso, é possível concluir que o ip do domain controller é 192.168.1.2.

| No. | Time | Source | Destination | Protocol | Length | CNameString | SNameString |
|-------|------------|---------------|---------------|----------|--------|-----------------|--|
| 1016 | 42.892696 | 192.168.1.216 | 192.168.1.2 | LDAP | 605 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1277 | 103.483774 | 192.168.1.216 | 192.168.1.2 | DCERPC | 772 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1304 | 103.496187 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1317 | 103.502560 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1751 | 1.791968 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMY... | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1771 | .792087 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMY... | cifs, SPOONWATCH-DC.spoonwatch.net |
| 3181 | .929590 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMY... | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 4453 | .385321 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMY... | cifs, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 89142 | .553870 | 192.168.1.2 | 192.168.1.216 | KRB5 | 345 | steve.smith | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 97642 | .781592 | 192.168.1.2 | 192.168.1.216 | KRB5 | 299 | steve.smith | cifs, SPOONWATCH-DC |

```

pvno: 5
msg-type: krb-tgs-rep (13)
crealm: SPOONWATCH.NET
  cname
    name-type: KRB5-NT-PRINCIPAL (1)
    > cname-string: 1 item
  ticket
    tkt-vno: 5
    realm: SPOONWATCH.NET
  sname
    name-type: KRB5-NT-SRV-INST (2)
    > sname-string: 3 items
      SNameString: LDAP
      SNameString: SPOONWATCH-DC.spoonwatch.net
      SNameString: spoonwatch.net
    > enc-part
  > enc-part
  
```

Packets: 5880 · Displayed: 32 (0.5%) · Profile: Default

20)

Observando o cname String do pedido de ticket do protocolo kerberos, é possível encontrar o nome do usuário associado ao ip de interesse.

User account: steve.smith

| No. | Time | Source | Destination | Protocol | Length | CNameString | SNameString |
|-------|------------|---------------|---------------|----------|--------|-------------------|--|
| 1016 | 42.892696 | 192.168.1.216 | 192.168.1.2 | LDAP | 605 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1277 | 103.483774 | 192.168.1.216 | 192.168.1.2 | DCERPC | 772 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1304 | 103.496187 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1317 | 103.502560 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1751 | 1.791968 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMYN02\$ | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 1771 | .792087 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMYN02\$ | cifs, SPOONWATCH-DC.spoonwatch.net |
| 3181 | .929590 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMYN02\$ | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 4453 | .385321 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMYN02\$ | cifs, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 89142 | .553870 | 192.168.1.2 | 192.168.1.216 | KRB5 | 345 | steve.smith | LDAP, SPOONWATCH-DC.spoonwatch.net, spoonwatch |
| 97642 | .781592 | 192.168.1.2 | 192.168.1.216 | KRB5 | 299 | steve.smith | cifs, SPOONWATCH-DC |

```

Frame 891: 345 bytes on wire (2760 bits), 345 bytes captured (2760 bits)
Ethernet II, Src: Dell_62:ae:26 (20:47:47:62:ae:26), Dst: ASUSTekCOMPU_32:58:f9 (9c:5c:8e:32:58:f9)
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.216
Transmission Control Protocol, Src Port: 88, Dst Port: 49718, Seq: 1461, Ack: 1813, Len: 291
[2 Resassembled TCP Segments (1751 bytes): #890(1460), #891(291)]
  Kerberos
    Record Mark: 1747 bytes
      0... . .... . .... . .... . .... = Reserved: Not set
      .000 0000 0000 0000 0110 1101 0011 = Record Length: 1747
    tgs-rep
      pvno: 5
      msg-type: krb-tgs-rep (13)
      crealm: SPOONWATCH.NET
      cname
        name-type: KRB5-NT-PRINCIPAL (1)
        > cname-string: 1 item
      ticket
        
```

Packets: 5880 · Displayed: 32 (0.5%) · Profile: Default

21)

O hostname é DESKTOP-GXMYNO2\$, obtido via sname do protocolo kerberos, TGS-REP.

| No. | Time | Source | Destination | Protocol | Length | CNameString | SNameString |
|-------|------------|---------------|---------------|----------|--------|-------------------|------------------------------------|
| 1016 | 42.892696 | 192.168.1.216 | 192.168.1.2 | LDAP | 605 | | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 1277 | 103.483774 | 192.168.1.216 | 192.168.1.2 | DCERPC | 772 | | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 1304 | 103.496187 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 1317 | 103.502560 | 192.168.1.216 | 192.168.1.2 | LDAP | 663 | | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 1751 | 7.791968 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMYNO2\$ | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 1771 | 7.792087 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMYNO2\$ | cifs, SPOONWATCH-DC.spoonwatch.net |
| 3181 | 1.929590 | 192.168.1.2 | 192.168.1.216 | KRB5 | 387 | DESKTOP-GXMYNO2\$ | LDAP, SPOONWATCH-DC.spoonwatch.net |
| + 445 | 3.385321 | 192.168.1.2 | 192.168.1.216 | KRB5 | 403 | DESKTOP-GXMYNO2\$ | cifs, SPOONWATCH-DC.spoonwatch.net |
| 891 | 42.553870 | 192.168.1.2 | 192.168.1.216 | KRB5 | 345 | steve.smith | LDAP, SPOONWATCH-DC.spoonwatch.net |
| 976 | 42.781592 | 192.168.1.2 | 192.168.1.216 | KRB5 | 299 | steve.smith | cifs, SPOONWATCH-DC |

Frame 445: 403 bytes on wire (3224 bits), 403 bytes captured (3224 bits)
Ethernet II, Src: Dell_62:ae:26 (20:47:47:62:ae:26), Dst: ASUSTekCOMPU_32:58:f9 (9c:5c:8e:32:58:f9)
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.216
Transmission Control Protocol, Src Port: 88, Dst Port: 49700, Seq: 1461, Ack: 1871, Len: 349
[2 Reassembled TCP Segments (1809 bytes): #444(1460), #445(349)]
Kerberos
Record Mark: 1805 bytes
0... = Reserved: Not set
.000 0000 0000 0000 0111 0000 1101 = Record Length: 1805
tgs-rep
pvno: 5
msg-type: krb-tgs-rep (13)
crealm: SPOONWATCH.NET
cname
name-type: KRB5-NT-PRINCIPAL (1)
> cname-string: 1 item
ticket
desafio4.pcap

22)

Filtrando os requests por http e apenas aqueles que contém zip, podemos encontrar uma série de requests no qual o host de interesse envia jpgs para um host externo.

Ip contaminado: 192.168.1.216 Ip suspeito: 2.56.57.108

o endereço suspeito para o qual o host envia os arquivos é: <http://2.56.57.108/osk//main.php>

Activities Wireshark fev 27 14:19 desafio4.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http contains "zip"

| No. | Time | Source | Destination | Protocol | Length | CName\$SName\$Text | Info |
|------|------------|---------------|-------------|----------|--------|--------------------|------------------------------|
| 4816 | 207.181483 | 192.168.1.216 | 2.56.57.108 | HTTP | 83 | ✓ | POST /osk/ HTTP/1.1 (zip) |
| 4275 | 206.429996 | 192.168.1.216 | 2.56.57.108 | HTTP | 542 | ✓ | POST /osk//main.php HTTP/1.1 |
| 4194 | 205.559859 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//7.jpg HTTP/1.1 |
| 3054 | 205.023863 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//5.jpg HTTP/1.1 |
| 2691 | 204.747216 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//4.jpg HTTP/1.1 |
| 2541 | 204.548240 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//3.jpg HTTP/1.1 |
| 2225 | 204.257253 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//2.jpg HTTP/1.1 |
| 1641 | 203.556638 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//1.jpg HTTP/1.1 |
| 1501 | 203.017250 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//6.jpg HTTP/1.1 |

```

Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-bitmap, *
Accept-Language: ru-RU,ru;q=0.9,en;q=0.8\r\n
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1\r\n
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0\r\n
Content-Type: multipart/form-data; boundary=1BEF0A57BE110FD467A\r\n
Content-Length: 25\r\n
[Content length: 25]
Host: 2.56.57.108\r\n
Connection: Keep-Alive\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://2.56.57.108/osk//main.php]
[HTTP request 8/9]
[Prev request in frame: 4194]
[Response in frame: 4277]
[Next request in frame: 4816]

```

The full requested URI (including host name) (http.request.full_uri)

Packets: 5880 · Displayed: 9 (0.2%) Profile: Default

23)

O request sai da porta 49738 (src port do request) do ip contaminado (192.168.1.216) e vai para a porta 80 do ip suspeito (dst port do request).

Activities Wireshark fev 27 14:23 desafio4.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http contains "zip"

| No. | Time | Source | Destination | Protocol | Length | CName\$SName\$Text | Info |
|------|------------|---------------|-------------|----------|--------|--------------------|------------------------------|
| 4816 | 207.181483 | 192.168.1.216 | 2.56.57.108 | HTTP | 83 | ✓ | POST /osk/ HTTP/1.1 (zip) |
| 4275 | 206.429996 | 192.168.1.216 | 2.56.57.108 | HTTP | 542 | ✓ | POST /osk//main.php HTTP/1.1 |
| 4194 | 205.559859 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//7.jpg HTTP/1.1 |
| 3054 | 205.023863 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//5.jpg HTTP/1.1 |
| 2691 | 204.747216 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//4.jpg HTTP/1.1 |
| 2541 | 204.548240 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//3.jpg HTTP/1.1 |
| 2225 | 204.257253 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//2.jpg HTTP/1.1 |
| 1641 | 203.556638 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//1.jpg HTTP/1.1 |
| 1501 | 203.017250 | 192.168.1.216 | 2.56.57.108 | HTTP | 539 | ✓ | POST /osk//6.jpg HTTP/1.1 |

```

Frame 4275: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits)
Ethernet II, Src: ASUSTekCOMPU_32:58:f9 (9c:5c:8e:32:58:f9), Dst: Cisco_f6:df:0a (1c:17:d3:f6:df:0a)
Internet Protocol Version 4, Src: 192.168.1.216, Dst: 2.56.57.108
Transmission Control Protocol, Src Port: 49738, Dst Port: 80, Seq: 3396, Ack: 3034378, Len: 488
Source Port: 49738
Destination Port: 80
[Stream index: 65]
[Conversation completeness: Complete, WITH_DATA (47)]
[TCP Segment Len: 488]
Sequence Number: 3396 (relative sequence number)
Sequence Number (raw): 1772256039
[Next Sequence Number: 3884 (relative sequence number)]
Acknowledgment Number: 3034378 (relative ack number)
Acknowledgment number (raw): 478411595
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)

```

Destination Port (tcp.dstport), 2 bytes

Packets: 5880 · Displayed: 9 (0.2%) Profile: Default