

Análise forense a redes de computadores

António Pinto
apinto@estg.ipp.pt

Escola Superior de Tecnologia e Gestão

Dezembro 2020 (v2)

Sumário

Visão geral

Análise em redes

Captura de tráfego

Berkeley Packet Filters (BPF)

Utilitários

Conceitos introdutórios

- ▶ Análises **possíveis** em redes de computadores
 - ▶ Identificação de equipamentos (endereços MAC e IP)
 - ▶ Identificação de utilizadores (credenciais de acesso PC/Rede¹)
 - ▶ Captura de pacotes transmitidos em pontos chave da rede
 - ▶ Identificação e análise protocolar (HTTP, SMTP, POP, ...)
 - ▶ Localização de equipamentos em redes locais

¹Muito comum em redes WLAN empresariais com 802.1X

Conceitos introdutórios (2)

- ▶ Momento de **início da captura** é um ponto crítico
 - ▶ Inexistência prévia de mecanismo de captura de tráfego pode inviabilizar a sua utilização
 - ▶ Nestes casos, análise deve focar mais nos equipamentos de rede e *logs*
 - ▶ Quantidade de informação analisar tende a ser menor
 - ▶ Início da análise vs. instante de tempo da atividade ilegal/incorrecta

Sumário

Visão geral

Análise em redes

Captura de tráfego

Berkeley Packet Filters (BPF)

Utilitários

Complexidade

Análise forense a uma rede é de elevado grau de complexidade

- ▶ Número de equipamentos a analisar
- ▶ Heterogeneidade (equipamentos, sistemas operativos, serviços, aplicações)
- ▶ Quantidade de informação a circular na rede
- ▶ Diferentes formas de se obter a informação

Características intrínsecas

1. Segmentação

Dados trocados sob a forma de pacotes IP.

2. Encriptação

Informação trocada pode estar cifrada para impedir a sua leitura por terceiros (ex: HTTPS)

3. Temporalidade

Tráfego de rede só pode ser capturado em tempo-real

4. Localização

Intervenientes podem ser muitos, estar noutro lado do mundo e podem simular uma localização diferente da real

Procedimento de análise

Procedimento de análise em redes de computadores inclui as seguintes fases:

1. Estudo da rede, equipamento e serviços existentes
2. Recolha e processamento de tráfego de rede
3. Recolha e processamento de *logs* e informação de estado

Seguindo-se a análise forense pelo técnico

Procedimento de análise

1. Estudo da rede, equipamento e serviços existentes

- ▶ Identificar os principais equipamentos de rede e suas funcionalidades
- ▶ Identificar gamas de endereçamento IP utilizadas
- ▶ Verificar se existem diagramas (físicos e lógicos) da rede (e se são actuais)
- ▶ Identificar servidores aplicativos (email, web, proxy), de autenticação (AD, LDAP) e de *logs* (syslog, splunk)

Enumeração rápida de equipamentos

Angry IP Scanner

<http://angryip.org>

- ▶ Pesquisa por equipamentos numa rede IP
- ▶ Funciona na LAN e Internet
- ▶ Exporta dados em muitos formatos
- ▶ Multi-plataforma (Win/Linux/macOS)
- ▶ Identifica também serviços
 - ▶ <https://www.iana.org/assignments/port-numbers>
- ▶ Pode ser usado como comando ou GUI

- ▶ Possibilita descoberta de serviços na rede
- ▶ Varrimentos de portas a equipamentos
- ▶ Tenta identificar serviços e sistemas operativos
- ▶ Ferramenta de linha de comandos
- ▶ Também dispõem de GUI (Zenmap)
- ▶ Suporta várias formas de varrimento

Network Mapper

Identificação de sistemas operativos (opção -O)

Uma das opções do nmap é a de tentar descobrir qual é o sistema operativo em execução nos equipamentos analisados.

```
aap@~ $sudo nmap -O 10.0.0.1

Host is up (0.012s latency).
MAC Address: E8:DE:27:11:11:11 (Tp-link Technologies)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.23 - 2.6.38
Network Distance: 1 hop

Nmap done: 1 IP address (1 host up)
scanned in 5.65 seconds
```

Enumeração de equipamentos

Exercício #2 (20 minutos)

Recorra ao **nmap** para tentar identificar os sistemas operativos de todos os equipamentos presentes na sua rede.

Conte, manualmente, o número de equipamentos por sistema operativo.

Submeta a sua resposta pelo **moodle**.

Procedimento de análise

2. Recolha e processamento de tráfego de rede

- ▶ Recolha pode ser intrusiva ou não
- ▶ Técnica depende do meio físico (*passive ethernet taps*)
- ▶ Fácil se equipamento de rede possibilitar técnicas para o efeito (*port mirroring*)
- ▶ Existe *software* (tcpdump, dumpcap, tshark, ...) e *hardware* específico para capturar tráfego de rede
- ▶ Formato comum (PCAP) pode ser processado com *Berkeley Packet Filters* (BPF)

Captura de tráfego com Wireshark

Exercício #3 (15 minutos)

Caso ainda não o tenha feito, instale o **wireshark**. Este programa permite a captura e análise de tráfego de rede.

De seguida, **inicie uma captura** de rede, **execute o comando** abaixo e **pare a captura**.

```
aap@~ $ping www.sapo.pt
PING www.sapo.pt (213.13.146.142): 56 data bytes
64 bytes from 213.13.146.142: time=20.503 ms
64 bytes from 213.13.146.142: time=17.205 ms
```

Submeta a sua **captura** pelo **moodle**.

Análise de captura com Wireshark

Exercício #4 (30 minutos)

Analise a captura efetuada no exercício anterior. Foque-se em particular no tráfego de rede gerado pelo comando efetuado.

Responda às seguintes perguntas:

1. Qual é a porta de destino do pedido de resolução de nomes (DNS)?
2. Qual é a porta de origem da resposta ao pedido de resolução de nomes (DNS)?
3. Para que endereço IP foi enviado o pedido de resolução de nomes? Este IP é o igual ao definido como servidor de DNS do seu computador?
4. Qual é o tipo de query DNS do pedido?
5. A mensagem de resposta contém alguma resolução?
6. Quantas resoluções aparecem na mensagem de resposta?

Submeta a sua resposta pelo **moodle**.

Procedimento de análise

3. Recolha e processamento de *logs* e informação de estado

- ▶ Evitar desligar equipamentos
- ▶ Ligação aos equipamentos de rede por cabos de consola (de preferência) → *out of band!*
- ▶ Registrar a data/hora dos equipamentos e calcular a diferença para o real
- ▶ Recolher informação de acordo com a sua volatilidade

Sumário

Visão geral

Análise em redes

Captura de tráfego

Berkeley Packet Filters (BPF)

Utilitários

Portas de monitorização

- ▶ *Switchs* têm técnicas para captura de tráfego de rede
- ▶ *Port mirror* é um nome usual, mas depende do fabricante
- ▶ Consiste em replicar para uma segunda porta, todo o tráfego que passa numa porta
- ▶ Utilizado em conjunto com IDS, NIDS, SIEMs, ...

Portas de monitorização

- ▶ Não implica alterar a estrutura da rede
- ▶ Requer credenciais de configuração dos equipamentos de rede
- ▶ Pode impor limitações ao tráfego analisado
- ▶ Obriga conhecer comandos de configuração em equipamentos de diferentes fabricantes

Portas de monitorização em Cisco

Como configurar a porta Fa0/10 como *mirror* da porta Fa0/1 (ambas as direções)

```
monitor session 1 source interface Fa0/1 both
monitor session 1 destination interface Fa0/10
```

Requer modo de acesso privilegiado e de configuração no equipamento Cisco

Portas de monitorização em Cisco

Como consultar os *port mirroring* existentes

```
show monitor

Session 1
-----
Type : Local Session
Source Ports:
RX Only: None
TX Only: None
Both: Fa0/1
Source VLANs:
RX Only: None
TX Only: None
Both: None
Source RSPAN VLAN: None
Destination Ports: Fa0/10
...
```

Portas de monitorização em Cisco

Como enviar o tráfego de uma porta (Fa0/2) para duas portas (Fa0/11 e Fa0/12)

```
monitor session 2 source interface Fa0/2 both  
monitor session 2 destination interface Fa0/11  
monitor session 2 destination interface Fa0/12
```

Requer modo de acesso privilegiado e de configuração no equipamento Cisco

Portas de monitorização em Cisco

Como replicar pela porta Gi1/2, todo o tráfego enviado para a VLAN 10

```
monitor session 1 source vlan 10 rx  
monitor session 1 destination interface Gi1/2
```

Devem usar-se portas de saída *gigabit* e nunca o *both*

Requer modo de acesso privilegiado e de configuração no equipamento Cisco

Berkeley Packet Filters (BPF)

O que são?

BPF é uma forma de descrever filtros a aplicar a recolhas de tráfego de rede

- ▶ Incluído no *kernel* de alguns sistemas operativos (ex.: FreeBSD)
- ▶ Suportado pela generalidade de aplicações baseadas em *libpcap* (ex.: tcpdump, Wireshark)
- ▶ Reconhece, e permite a utilização, das estruturas dos principais protocolos de rede (Ethernet, MPLS, IP, UDP, TCP, ICMP, ...)
- ▶ Elevado nível de granularidade (análise *bit-a-bit*)

Berkeley Packet Filters (BPF)

Como funciona?

Aplicações de recolha de tráfego de rede capturam todos os pacotes, excepto se for especificada uma expressão BPF

- ▶ Quando presente, apenas são capturados os pacotes que confirmem a expressão BPF (i.e. quando a expressão é verdadeira)
- ▶ Expressão é composta por *qualifiers*
- ▶ Ex.: `src host 10.10.1.1`
 - ▶ **src:** *qualifier* de direcção
 - ▶ **host:** *qualifier* de tipo
 - ▶ **10.10.1.1:** identificador (ou *id*)

Qualifiers BPF

Tipos de *qualifiers* em BPF

Que representam o tipo da coisa a procurar

- ▶ **host** para equipamentos
- ▶ **net** para redes
- ▶ **port** para portos aplicativos

Que representam a direção do tráfego

- ▶ **src** para origem
- ▶ **dst** para destino
- ▶ **src or dst**, **src and dst** como combinações

Que representam o protocolo

- ▶ **ether**, **fddi**, **ip**, **ip6**, **arp**, **rarp**, **tcp**, **udp**

Casos particulares

- ▶ **gateway**, **broadcast**, **less**, **greater**

Primitivas BPF

- ▶ Primitiva é a forma básica de representação de padrões de pesquisa
- ▶ Conseguído por combinação de *qualifiers* e identificadores
- ▶ Exemplos
 - ▶ **dst host 10.1.1.1**
 - ▶ **gateway 192.168.1.1**
 - ▶ **ip broadcast**

Exemplo de utilização

(com tcpdump)

Capturar todos os pacotes com porta 80 e guardar em ficheiro.

```
[aap@eb-aap ~] $ sudo tcpdump -w http.pcap port http
tcpdump: listening on enp0s25, link-type EN10MB,
capture size 262144 bytes

^C
49 packets captured
49 packets received by filter
0 packets dropped by kernel
```

Filtragem de capturas

Exercício #5 (15 minutos)

Utilizando a captura do exercício anterior e o comando **tcpdump**, grave num segundo ficheiro PCAP os pacotes que cumpram com pelo menos um dos seguintes:

- ▶ Porto destino 53
- ▶ Protocolo UDP
- ▶ Aplicação HTTPS

Submeta a sua **captura** pelo **moodle**.

Operação ao nível do bit

Relembrar alguns cabeçalhos

Cabeçalho IP

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Tamanho (Bytes)	
16-bit Identification			3-bit Flags	13-bit Fragment Offset
8-bit Time to Live (TTL)		8-bit Protocolo	16-bit Header Checksum	
32-bit Endereço IP Origem				
32-bit Endereço IP Destino				
Dados (payload)				

Cabeçalho (20 bytes)

Operação ao nível do bit (2)

Relembrar alguns cabeçalhos

Cabeçalho TCP

0	4	10	16	31
Porto origem			Porto destino	
N.º Sequência				
N.º Ack				
Hlen	Reserved	Flags	Window	
Checksum			Urgent Pointer	
Options (if any)			Padding	
Data				
...				

Operação ao nível do bit (3)

- ▶ Expressões como 'ip[9] = 0x06' são também possíveis
 - ▶ **Verdadeiro** se o 10º *byte* (1º *byte* é 0) do cabeçalho IP é igual a 6 (ou 0x06 em hexadecimal)
 - ▶ Equivale a **proto tcp**

Operação ao nível do bit (4)

Bitwise AND

- ▶ IP têm duas versões (IPv4 e IPv6)
- ▶ Número (4 ou 6) da versão é incluída nos 4 *bits* mais significativos (à direita) do 1º *byte*
- ▶ BPF permitem a pesquisa por determinados *bits* usando-se a operação *Bitwise AND* (operação E lógico, *bit-a-bit*)
 - ▶ Primitiva para IPv6: 'ip[0] & 0xF0 = 0x06'
 - ▶ Primitiva para IPv4: 'ip[0] & 0xF0 = 0x04'

Operação ao nível do bit (5)

Bitwise AND

Descompondo a primitiva: 'ip[0] & 0xF0 = 0x06'

- ▶ ip[0] → 1^o *byte* do cabeçalho IP (ex.: 0110 0101)
- ▶ ip[0] & 0xF0 → AND Lógico entre ip[0] e 0xF0

$$\begin{array}{r} \text{ip}[0] \quad 0110\ 0101 \\ 0xF0 \quad 1111\ 0000 \\ \hline 0110\ 0000 \end{array}$$

- ▶ Obtendo-se assim o valor a comparar ($0110_{(2)} = 6_{(10)}$)

Operação ao nível do bit (6)

Bitwise AND

Mais exemplos:

- ▶ UDP: 'ip[9] = 0x11'
- ▶ ICMP: 'ip[9] = 0x01'
- ▶ Porta TCP destino inferior a 20: 'tcp[2:2] < 0x14'
- ▶ ICMP Echo request: 'icmp[0] = 0x08'

Combinação de primitivas

Visando a redução do tráfego em análise, é por vezes é necessário combinar-se primitivas.

- ▶ Negação: 'not' ou '!'
- ▶ Conjunção: 'and' ou '&&'
- ▶ Disjunção: 'or' ou '||'

Programas úteis

A quantidade de informação que circula na rede é muita. É comum que o técnico recorra a ferramentas para o auxiliar na análise, destas destacam-se as seguintes:

- ▶ **wireshark**
A ferramenta mais comum e completa
- ▶ **tshark**
Versão em modo de texto do *wireshark*, útil para *scripts* ou quando não se dispõem de GUI
- ▶ **tcpextract**
Extração de ficheiros em trocas HTTP de capturas
+info:<https://pypi.python.org/pypi/tcpextract>
- ▶ **tcpxtract**
Extração de ficheiros de capturas usando assinaturas hexadecimais
+info:<http://tcpxtract.sourceforge.net/>

+ Programas úteis

▶ **tcpflow**

Extração de sessões TCP em capturas de rede

+info:<http://digitalcorpora.org/downloads/tcpflow/>

▶ **ntop**

Versão para redes do comando *top* (*web-based*)

+info:<http://www.ntop.org/>

▶ **ngrep**

Versão para redes do comando *grep*

+info:<http://ngrep.sourceforge.net/>

▶ **tcpdump**

Utilitário mais comum para captura de tráfego de rede

+info:<http://www.tcpdump.org/>

Utilitários

43/47

Exemplos de utilização

ngrep

Capturar pedidos de DNS em qualquer interface e guardando o resultado da captura para ficheiro.

```
[aap@eb-aap ~]$ ngrep -O dns.cap -d any -T port domain
interface: any
filter: ip and ( port domain )
output: dns.cap
#
U +0.000000 203.115.225.24:53 -> 64.90.164.74:53
.....m.razor2.cloudmark.com.....).....
#
U +0.000281 64.90.164.74:53 -> 203.115.225.24:53
.....m.razor2.cloudmark.com.....' . ns1 ...
hostmaster..ws....p.... :.....).....
#
U +0.078184 195.113.155.7:2949 -> 64.90.164.74:53
.....a.razor2.cloudmark.com.....
^Cexit
6 received , 0 dropped
```

Utilitários

44/47

Exemplos de utilização

tshark

Capturar pacotes com destino à porta 80, guardando o resultado da captura para ficheiro.

```
[aap@eb-aap ~] $ sudo tshark -f "port 80" -w file.cap
Capturing on 'enp0s25'
1159 ^C
```

O filtro especificado (com -f) é um BPF e não um filtro utilizável no *wireshark*².

²O filtro *wireshark* equivalente seria "tcp.port == 80"

Exemplos de utilização

tshark

Mostrar a porta de origem dos pacotes TCP de uma captura.

```
[aap@eb-aap ~] $ sudo tshark -z "proto,colinfo,tcp.srcport" -r
/tmp/capture.cap
1 0.000000000 172.20.100.154 -> 69.89.31.120 TCP 74 58537->80 [
  SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval
  =10379680 TSecr=0 WS=128 tcp.srcport == 58537
2 0.218503000 69.89.31.120 -> 172.20.100.154 TCP 74 80->58537 [
  SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=576 SACK_PERM=1
  TSval=123744260 TSecr=10379680 WS=128 tcp.srcport == 80
3 0.218613000 172.20.100.154 -> 69.89.31.120 TCP 66 58537->80 [
  ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=10379898 TSecr
  =123744260 tcp.srcport == 58537
```

Algumas opções do *tshark* só podem ser utilizadas sobre capturas em ficheiro.