Banco de dados de fluxos para análises de segurança.

Grupo de Trabalho em Segurança – GTS
São Paulo, 27 de Outubro de 2007.

André Proto

Jorge Luiz Corrêa

UNESP – Universidade Estadual Paulista – Instituto de Biociências, Letras e Ciências Exatas (IBILCE) - Campus de São José do Rio Preto, SP

Organização da apresentação

- Motivação e objetivos
- Fluxo de dados (Netflow)
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- Aferências e detecção de eventos (consultas SQL)
- Comparações com outras ferramentas
- Conclusão
  - Trabalhos em desenvolvimento

Organização da apresentação

- **Motivação e objetivos**
- Fluxo de dados (Netflow)
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- Aferências e detecção de eventos (consultas SQL)
- Comparações com outras ferramentas
- Conclusão
  - Trabalhos em desenvolvimento

Motivação e objetivos

- **Motivações:**
  - Dificuldade em analisar o tráfego de uma rede de grande porte.
  - Dependência de ferramentas que manipulam fluxos.
  - Obter resultados mais precisos na análise dos fluxos.
  - Necessidade de um modelo de armazenamento dos fluxos de dados que seja utilizado por qualquer software de análise.

Motivação e objetivos

- **Objetivos:**
  - Criar um modelo de manipulação de fluxos que seja independente de qualquer ferramenta.
  - Prover infra-estrutura de acesso a fluxos para o desenvolvimento de metodologias de análise de redes.
  - Resultados precisos utilizando os recursos do Sistema Gerenciador de Banco de Dados (SGBD).

Organização da apresentação

- Motivação e objetivos
- **Fluxo de dados (Netflow)**
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- Aferências e detecção de eventos (consultas SQL)
- Comparações com outras ferramentas
- Conclusão
  - Trabalhos em desenvolvimento

Fluxo de dados (*Netflow*)

- **Histórico:**
  - Desenvolvimento de aplicações necessitavam de medições de tráfego dentro de uma rede de computadores.
  - Diversos padrões foram propostos dentre eles o *Netflow* pela *Cisco Systems* RFC-3954.
  - IETF propôs a criação do padrão IPFIX (*IP Flow Information* Export) RFC-3917.

Fluxo de dados (*Netflow*)

- **O que é um fluxo?**
  - Uma seqüência unidirecional de pacotes entre dois *hosts*.
  - Uma tupla na qual as seguintes informações aparecem com o mesmo valor:
    - Endereço IP de origem e destino;
    - Porta de origem e destino (camada de transporte);
    - Campo *Protocol* do datagrama IP;
    - Campo *Type of Service* do datagrama IP;
    - Interface lógica do datagrama no roteador ou switch.

Fluxo de dados (*Netflow*)

Flow Header Format (Netflow v.5)

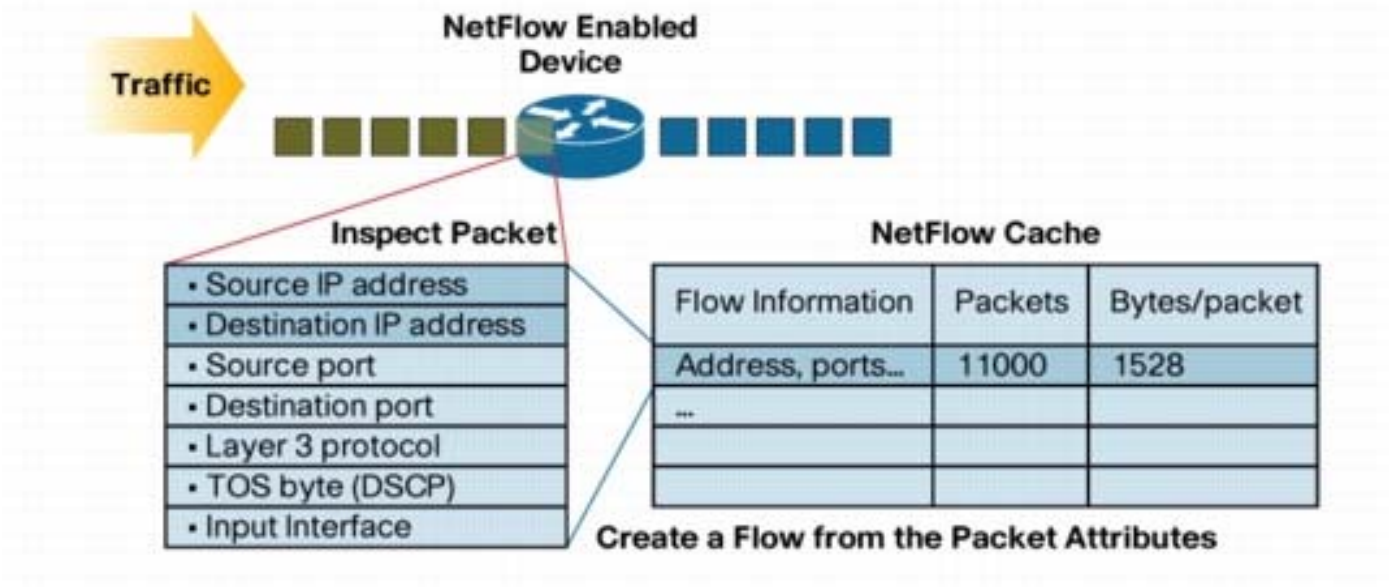
0	1	2	3	4	5	6	7
version		**count**		**sys_uptime**			
unix_secs				**unix_nsecs**			
flow_sequence				**engine_ type**	**engine_ id**	**sampling_interval**	

Flow Record Format (Netflow v.5)

0	1	2	3	4	5	6	7
srcaddr				**dstaddr**			
nexthop				**input**		**output**	
dPkts				**dOctets**			
fist				**last**			
srcport		**dstport**		**pad1**	**tcp_flags**	**prot**	**tos**
src_as		**dst_as**		**src_mask**	**dst_mask**	**pad2**	

Fluxo de dados (*Netflow*)

- **Os fluxos são exportados para o coletor quando:**
  - permanece ocioso por mais de 15 segundos;
  - uma conexão TCP é encerrada com a *flag* FIN ou RST;
  - a tabela de fluxos está cheia ou o usuário redefine as configurações de fluxo;
  - sua duração excede 30 minutos;



Organização da apresentação

- Motivação e objetivos
- Fluxo de dados (Netflow)
- **Apresentação do modelo**
  - **Coletor**
  - **Banco de dados de fluxos**
  - **Ambiente**
- Aferências e detecção de eventos (consultas SQL)
- Comparações com outras ferramentas
- Conclusão
  - Trabalhos em desenvolvimento

Coletor

- Desenvolvido em linguagem Java;
- Identifica os campos do *Netflow* (v5);
- Realiza transformações necessárias nos dados;
- Grava as informações coletadas no Banco de Dados.

Banco de Dados de Fluxos

- Modelo para banco de dados relacional;
- Sistema Gerenciador de Banco de Dados escolhido: MySQL;
- Campos da tabela no BD:
  - Baseado na proposta de *John-Paul Navarro, Bill Nickless e Linda Winkler* (http://www.usenix.org/event/lisa2000/navarro.html).
  - Alguns campos foram otimizados tornando o tamanho de cada tupla menor;

Pela proposta inicial:

IP
192.168.20.145

BIGINT
192168020145

Nossa proposta:

IP
192.168.20.145

INT
3232240785

Apresentação do modelo

Proposta incial (Tamanho da tupla: 57 bytes):

Field	Type
srcaddr	bigint(20) unsigned
dstaddr	bigint(20) unsigned
nexthop	bigint(20) unsigned
input	smallint(5) unsigned
output	smallint(5) unsigned
dPkts	int(10) unsigned
dOctets	int(10) unsigned
first	timestamp
last	timestamp
srcport	smallint(5) unsigned
dstport	smallint(5) unsigned
tcp_flags	tinyint(3) unsigned
prot	tinyint(3) unsigned
tos	tinyint(3) unsigned
src_as	smallint(5) unsigned
dst_as	smallint(5) unsigned
src_mask	tinyint(3) unsigned
dst_mask	tinyint(3) unsigned

Nova proposta (Tamanho da tupla: 45 bytes):

Field	Type
srcaddr	int(10) unsigned
dstaddr	int(10) unsigned
nexthop	int(10) unsigned
input	smallint(5) unsigned
output	smallint(5) unsigned
dPkts	int(10) unsigned
dOctets	int(10) unsigned
first	timestamp
last	timestamp
srcport	smallint(5) unsigned
dstport	smallint(5) unsigned
tcp_flags	tinyint(3) unsigned
prot	tinyint(3) unsigned
tos	tinyint(3) unsigned
src_as	smallint(5) unsigned
dst_as	smallint(5) unsigned
src_mask	tinyint(3) unsigned
dst_mask	tinyint(3) unsigned

Apresentação do modelo

- **Como converter um inteiro para o formato IPv4?**
  - Através das funções **INET_NTOA(expr)** e **INET_ATON(expr)**. Exemplo:

mysql> SELECT INET_NTOA(3232240785); → **192.168.20.145**

mysql> SELECT INET_ATON('192.168.20.145'); → **3232240785**

- Proposta inicial:
  7.000.000 x 57 bytes = 399.000.000 bytes ~ 380MB
- Nova proposta:
  7.000.000 x 45 bytes = 315.000.000 bytes ~ 300MB

Economia de ~ 21%!

Características da base de dados

- Cada tabela do banco armazena fluxos referentes a 1 dia;
  - Restrição com base na perda de desempenho para tabelas maiores.
- Campo base: marcação do início do fluxo (*first*);
- Nome da tabela: afm"%date", ou seja, afm20071027.
- Tabela especial ***last30minutes***:
  - Contém os últimos 30 minutos de fluxos (restrição dos fluxos);
  - A consulta nessa tabela é mais rápida (consulta em memória);
  - Um procedimento SQL é responsável pela sincronia *last30minutes*→disco;
  - Facilita a seleção de *Top Talkers*.

Características da base de dados

- Duas outras tabelas têm papéis importantes: *input* e *output*.
  - Input (sessões e conexões de fora do ambiente para dentro);
  - Output (sessões e conexões de dentro do ambiente para fora);
  - São tabelas mantidas na memória com os fluxos de um intervalo de tempo limitado;
  - Um procedimento (produto cartesiano) é responsável por cruzar informações destas tabelas e **reconstruir todas as conexões** do ambiente.

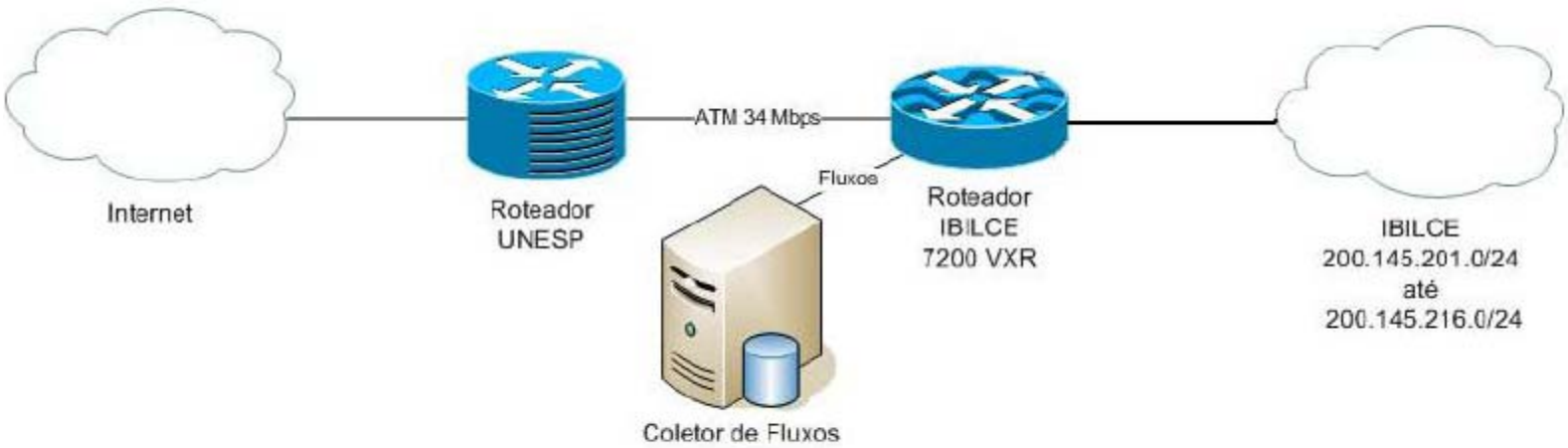
Características da base de dados

- Conexões de um ambiente: procedimento que utiliza tabelas *input* e *output*.

InsideHost	OutsideHost	BytesIn	BytesOut	dPktsIn	dPktsOut
192.168.20.5	192.168.81.247	69	85	1	1
192.168.20.5	192.168.129.129	286	418	4	4
192.168.20.5	192.168.85.78	126	95	2	1
192.168.20.5	192.168.123.182	238	334	3	3
192.168.20.5	192.168.124.72	158	179	2	1
192.168.20.5	192.168.61.36	66	230	1	1
192.168.20.5	192.168.2.44	156	452	2	2
192.168.20.5	192.168.219.3	156	476	2	2

Ambiente

- Esquema de coleta dos fluxos de dados:



- Máquina coletora:
  - Pentium IV 1.8GHz, 256KB Cache (Socket 478);
  - 768MB RAM;
  - HD IDE 80GB ATA-100;

Organização da apresentação

- Motivação e objetivos
- Fluxo de dados (Netflow)
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- **Aferências e detecção de eventos (consultas SQL)**
- Comparações com outras ferramentas
- Conclusão
  - Trabalhos em desenvolvimento

Aferências e detecção de eventos

- Sintaxe do comando de consultas no MySQL:
  SELECT select_expr
  FROM table_references
  [WHERE where_condition];

Contagem de fluxos

```
mysql> select count(*) from afm20071010;
+-----------+
| count(*)  |
+-----------+
| 9213879|
+-----------+
1 row in set (0.00 sec)
```

```
mysql> select count(*) as input
    -> from afm20071010
    -> where input=28;
+------------+
| input      |
+------------+
| 4128368 |
+------------+
1 row in set (24.19 sec)
```

```
mysql> select count(*) as output
    -> from afm20071010
    -> where output=28;
+-----------+
| output    |
+-----------+
| 4772825 |
+-----------+
1 row in set (3.59 sec)
```

Analisando o tráfego da rede: um conjunto de hosts e um único host.

```
mysql> select count(*) as input_net
    -> from afm20071010
    -> where inet_ntoa(dstaddr) like
    -> '200.145.202.%%%'
    -> and input=28;
+-------------+
| input_net   |
+-------------+
|   448273    |
+-------------+
1 row in set (9.02 sec)
```

```
mysql> select count(*) as host_in
    -> from afm20071010
    -> where dstaddr=inet_aton(
    -> '200.145.202.5') and
    -> input=28;
+-----------+
| host_in   |
+-----------+
|   13722   |
+-----------+
1 row in set (8.57 sec)
```

Contagem de fluxos por minuto.

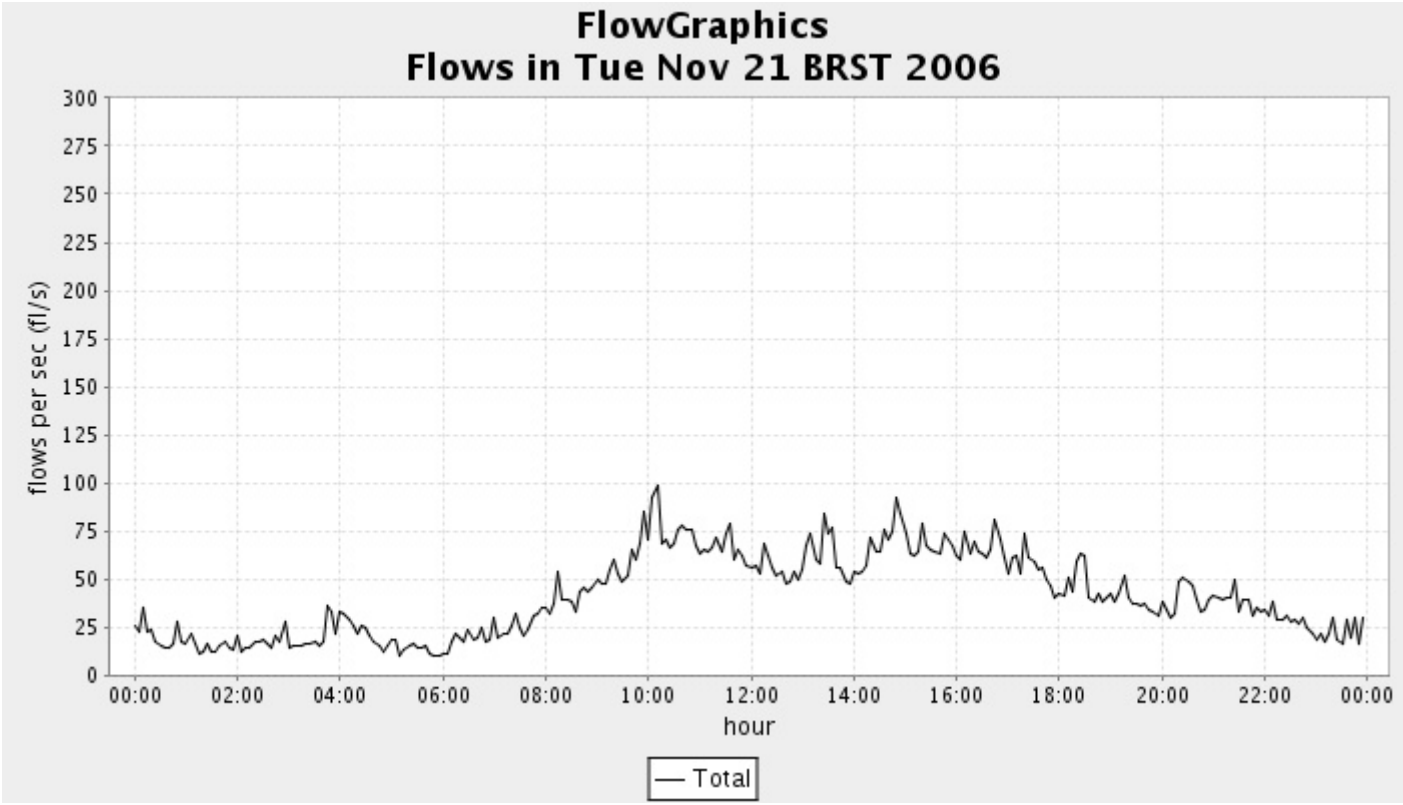
```
mysql> select date_sub(first,interval second(first) second) as Date,count(*) as
flows, count(*)/60 as flows_per_sec from afm20071010  where inet_ntoa(srcaddr)
like '200.145.202.%%%' group by hour(first),minute(first) order by first;
```

Date	flows	flows_per_sec
2007-10-10 00:00:00	234	3.9000
2007-10-10 00:01:00	266	4.4333
2007-10-10 00:02:00	217	3.6167
2007-10-10 00:03:00	285	4.7500
2007-10-10 00:04:00	296	4.9333
2007-10-10 00:05:00	268	4.4667
2007-10-10 00:06:00	267	4.4500
2007-10-10 00:07:00	237	3.9500
2007-10-10 00:08:00	223	3.7167
2007-10-10 00:09:00	269	4.4833
...	...	...

1440 rows in set (13.18 sec)

Aferências e detecção de eventos

- **Gráfico gerado por aplicação JAVA utilizando dados obtidos no banco.**



Os 10 hosts com maior número de octetos recebidos.

```
mysql> select inet_ntoa(dstaddr) as dst_address,sum(dPkts)/300 as pkts_sec_in,
sum(dOctets)/300 as bytes_sec_in from last30minutes where input=28 and
first > date_sub(now(),interval 5 minute) group by dstaddr order by
sum(dOctets) desc limit 10;
```

dst_address	pkts_sec_in	bytes_sec_in
192.168.20.9	270.2733	335454.8567
192.168.10.91	208.0900	265581.1367
192.168.30.247	560.3167	62886.5700
192.168.30.74	29.5600	44336.2667
192.168.90.111	29.4100	39109.0867
192.168.10.41	28.6633	35481.3333
192.168.40.99	20.2567	30335.3100
192.168.10.46	18.8000	27250.6167
192.168.90.222	19.4033	24144.2300
192.168.90.96	12.3900	14373.5400

10 rows in set (0.65 sec)

Os 10 hosts com maior número de fluxos de saída.

```
mysql> select inet_ntoa(srcaddr) as src_address, count(srcaddr) as flows,
count(srcaddr)/300 as flows_per_sec from last30minutes where first >
date_sub(now(), interval 5 minute) and output=28 group by srcaddr order by flows
desc limit 10;
```

src_address	flows	flows_per_sec
192.168.20.9	3471	11.5700
192.168.10.1	1245	4.1500
192.168.10.91	1141	3.8033
192.168.20.178	574	1.9133
192.168.130.68	558	1.8600
192.168.20.5	392	1.3067
192.168.40.80	383	1.2767
192.168.30.154	351	1.1700
192.168.70.168	332	1.1067
192.168.30.42	314	1.0467

10 rows in set (0.24 sec)

Detecção de ataques: hosts realizando prospecções na rede.

```
mysql> select inet_ntoa(srcaddr) as src_address, count(distinct dstaddr) as
hosts_scanned, count(distinct dstport) as ports_scanned from last30minutes
where first > date_sub(now(),interval 30 minute) and input="28" and tcp_flags & 2 = "2"
and timediff(last,first) < "00:00:15" group by srcaddr having ports_scanned > 300 or
hosts_scanned > 300 order by hosts_scanned;
```

src_address	hosts_scanned	ports_scanned
10.139.136.159	1	3466
193.49.208.251	2	15968
192.152.161.101	8	980
196.109.112.136	8	741
192.188.2.109	29	652
192.96.239.18	510	1
192.188.154.181	573	4
192.188.249.167	581	4
192.188.226.145	1402	5

9 rows in set (0.78 sec)

Detecção de ataques de dicionário no SSH.

```
mysql> select inet_ntoa(srcaddr) as src_address,inet_ntoa(dstaddr) as dst_address,
count(*) as attempts from afm20071010 where dstport="22" and tcp_flags & 2 ="2"
and inet_ntoa(dstaddr) like “192.168.%%%.%%%" and  timediff(last,first) < "00:00:15“
group by srcaddr,dstaddr having attempts > 100 order by attempts;
```

src_address	dst_address	attempts
192.194.17.103	192.168.30.204	113
192.17.109.114	192.168.30.21	177
194.77.105.210	192.168.30.73	319
192.17.109.114	192.168.211.65	447
192.17.109.114	192.168.30.11	686
192.17.109.114	192.168.30.230	795
192.17.109.114	192.168.30.17	940
192.17.109.114	192.168.30.62	1102
192.17.109.114	192.168.30.246	1176
192.17.109.114	192.168.30.58	1336

10 rows in set (5.90 sec)

Possíveis hosts utilizando serviços peer-to-peer.

```
mysql> select inet_ntoa(dstaddr) as Destination,count(distinct srcaddr) as
Num_Sources_Host,sum(dOctets) as Bytes,sum(dOctets)/time_to_sec(timediff(now(),
date_sub(now(), interval 30 minute))) as Bytes_per_second from last30minutes
where dstport>1024 and srcport>1024  and input=28 group by dstaddr having
Num_Sources_Host > 100 and Bytes > 1000000 order by Bytes;
+-----------------+-------------------+---------+-----------------+
| Destination     | Num_Sources_Host  | Bytes   | Bytes_per_second |
+-----------------+-------------------+---------+-----------------+
| 192.168.30.107  |               131 | 1754815 |          974.8972 |
| 192.168.130.236 |              2494 | 2523820 |          1402.1222 |
| 192.168.20.175  |               141 | 7859254 |          4366.2522 |
| 192.168.30.42   |              2922 | 11820309 |          6566.8383 |
| 192.168.100.207 |              1911 | 11835903 |          6575.5017 |
| 192.168.30.25   |              1408 | 50975412 |          28319.6733 |
| 192.168.130.200 |               125 | 54694986 |          30386.1033 |
+-----------------+-------------------+---------+-----------------+
7 rows in set (0.32 sec)
```

Organização da apresentação

- Motivação e objetivos
- Fluxo de dados (Netflow)
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- Aferências e detecção de eventos (consultas SQL)
- **Comparações com outras ferramentas**
- Conclusão
  - Trabalhos em desenvolvimento

Comparações com outras ferramentas

Flow-tools

- Análise do tempo de consulta aos fluxos:

Tempo em segundos para realizar consulta aos fluxos		
Consulta	**Banco de Dados**	**Flow-tools**
Quantidade de fluxos em um dia.	0.00s	31.78s
Contagem de fluxos por minuto.	13.18s	31.78s
Tráfego de uma rede específica.	13.76s	33.52s
10 hosts com maior número de octetos recebidos (24 horas).	24.73s	53.43s

- Análise sobre o armazenamento dos fluxos:

Espaço utilizado em disco		
Consulta	**$7x10^6$ fluxos**	**1 fluxo**
Banco de dados	~300 MB	45 bytes
Flow-tools	~134 MB	~20 bytes

Organização da apresentação

- Motivação e objetivos
- Fluxo de dados (Netflow)
- Apresentação do modelo
  - Coletor
  - Banco de dados de fluxos
  - Ambiente
- Aferências e detecção de eventos (consultas SQL)
- Comparações com outras ferramentas
- **Conclusão**
  - **Trabalhos em desenvolvimento**

Conclusão

- A análise de fluxos de dados consolidou-se como uma boa opção para análise e monitoramento do tráfego de uma rede;
- A utilização de bancos relacionais insere um **maior nível de abstração, grande versatilidade** e certa escalabilidade na manipulação de fluxos;
- Diversos eventos puderam ser detectados:
  - Tentativas de prospecção;
  - Ataques de dicionário;
  - Possíveis atividades com *file sharing;*
  - Possíveis ataques de DoS;
  - Estatísticas em geral.
- Conforme um dos objetivos, a infra-estrutura de armazenamento pode ser utilizada por quaisquer metodologias e sistemas novos, de maneira simples.

Trabalhos em desenvolvimento

- “Prover infra-estrutura de acesso a fluxos para o desenvolvimento de metodologias de análise de redes”.
  - Desenvolvimento de IDS baseados na análise de fluxos de dados;
  - Detecção de *DoS* e *DDoS*, com informações precisas;
  - Detecção de intrusão baseada em anomalias (análise comportamental);
  - Identificação de eventos na rede.

- Como conseqüência desta pesquisa:
  - Aplicação web para manipulação dos recursos do SGBD, geração de gráficos e estatísticas da rede, semelhante ao Flow-Scan.

Contatos

- **André Proto**
  andreproto.gts@acmesecurity.org
  PGP KeyID: 0xA6FC761A
- **Jorge Luiz Corrêa**
  jorge.gts@acmesecurity.org
  PGP KeyID: 0x1BCB7255

Obrigado!