AMRES
Serbian Academic Network

BELGRADE SECURITY WORKSHOP 2015

Identifying users behind NAT devices

Matěj Grégr

GÉANT

Network Address Translation

- Several NAT variants are described in RFC 3489
  - One-to-One = Full Cone = Static
  - Restricted cone, Port Restricted cone, Symetric
- Real implementations use own algorithms
  - Dynamic NAT is not standardized
- CGNAT
  - NAPT in ISP networks (NAT444)

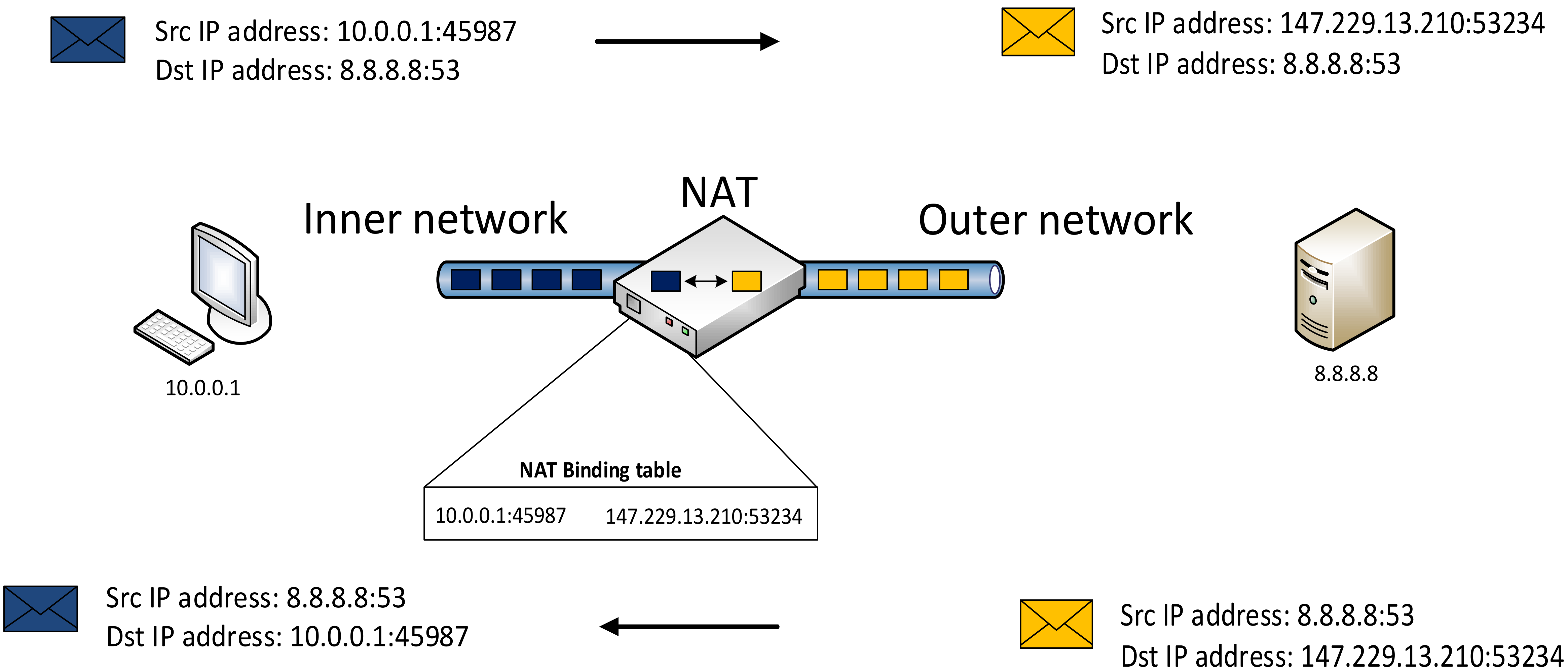
NAT and user identification

- User identification is lost if address translation is performed
- End server must log enough information – time, IP address, port number
- Several ways for keeping user identification:
  - User obtains fixed set of ports
  - Translation table is exported to another device:
    - NetFlow NEL
    - iptables ulogd
    - Export directly to a database
    - SNMP
    - …

NAT and NetFlow?

- NetFlow is popular accounting tool for keeping metadata about network communication
- Cisco NEL
  - NetFlow extension – Cisco only
  - Only creation/deletion of mapping is exported – transferred bytes are missing
- NetFlow and research:
  - Detection number of devices behind NAT
  - Detection NAT traffic from NetFlow data

Network Address Translation



Src IP address: 10.0.0.1:45987
Dst IP address: 8.8.8.8:53

Src IP address: 147.229.13.210:53234
Dst IP address: 8.8.8.8:53

Inner network

NAT

Outer network

10.0.0.1

8.8.8.8

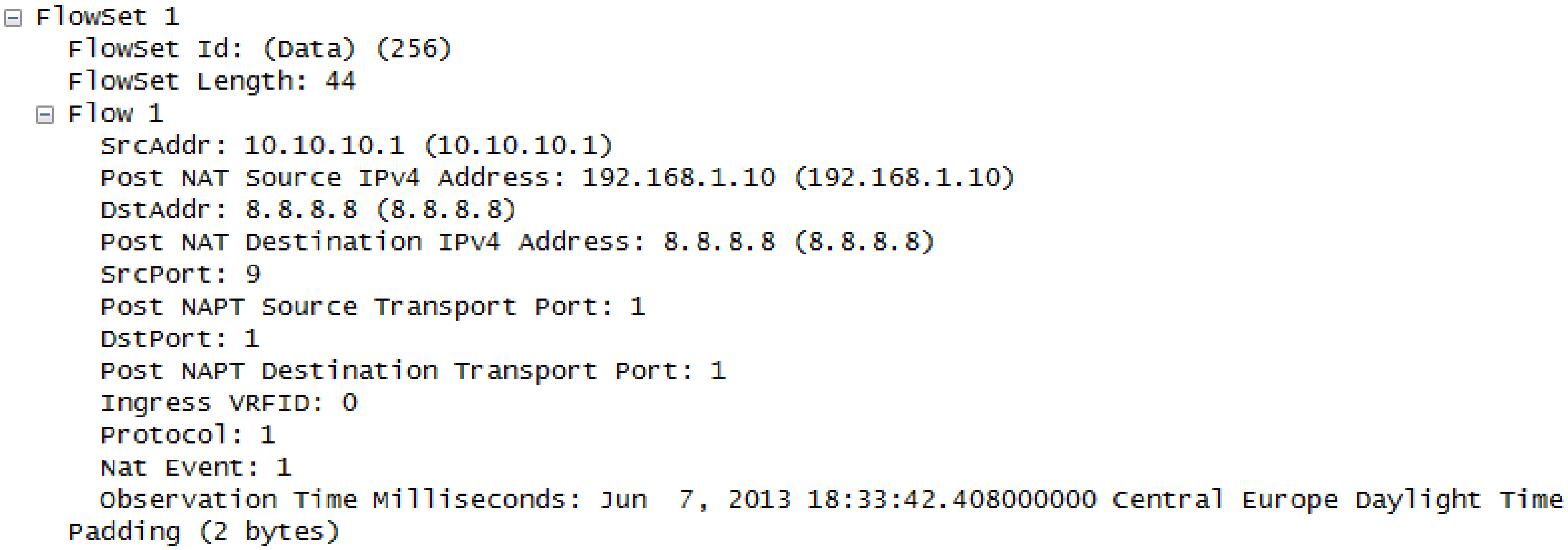
NAT Binding table

10.0.0.1:45987	147.229.13.210:53234

Src IP address: 8.8.8.8:53
Dst IP address: 10.0.0.1:45987

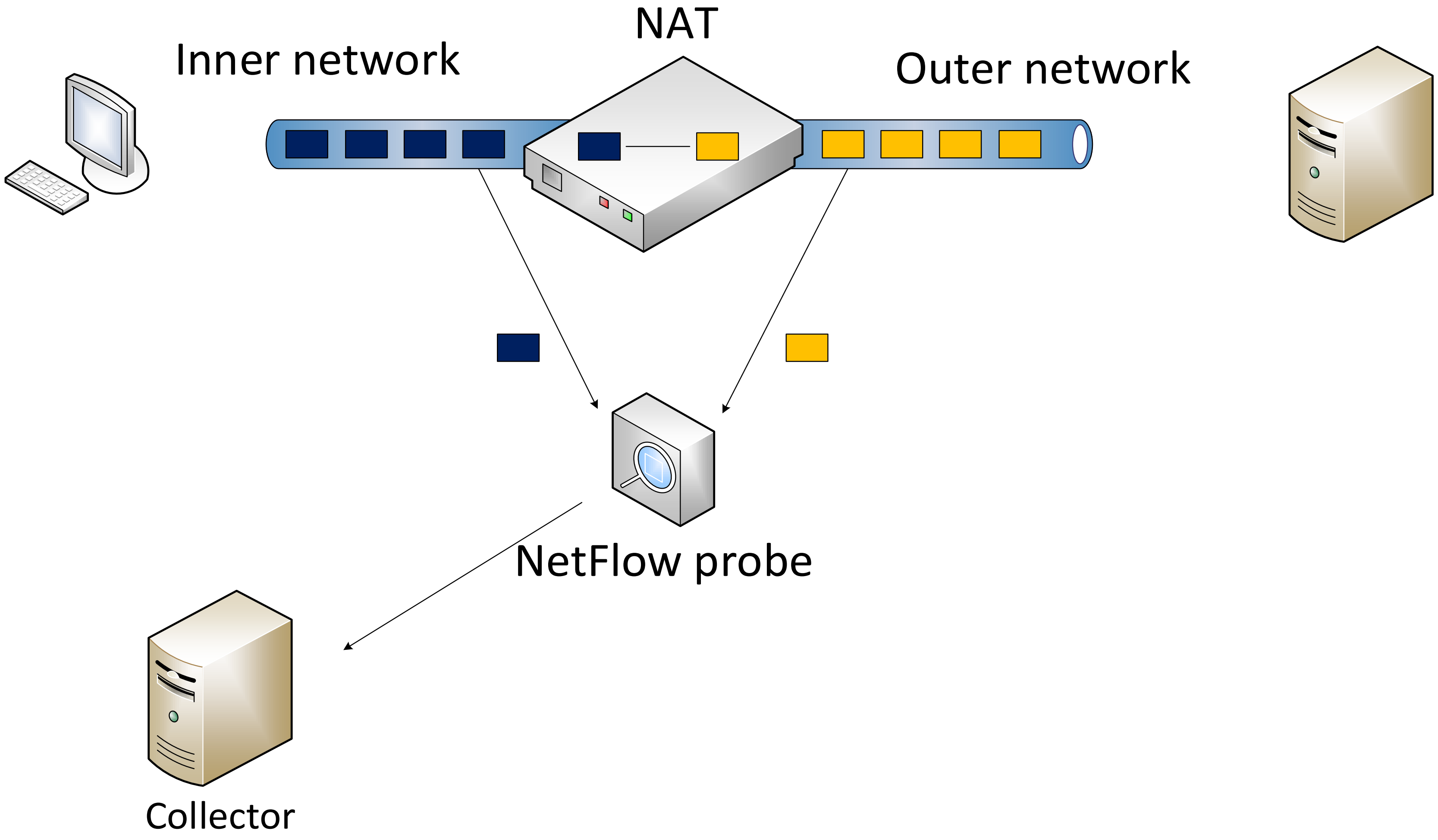
Src IP address: 8.8.8.8:53
Dst IP address: 147.229.13.210:53234

NAT NetFlow export

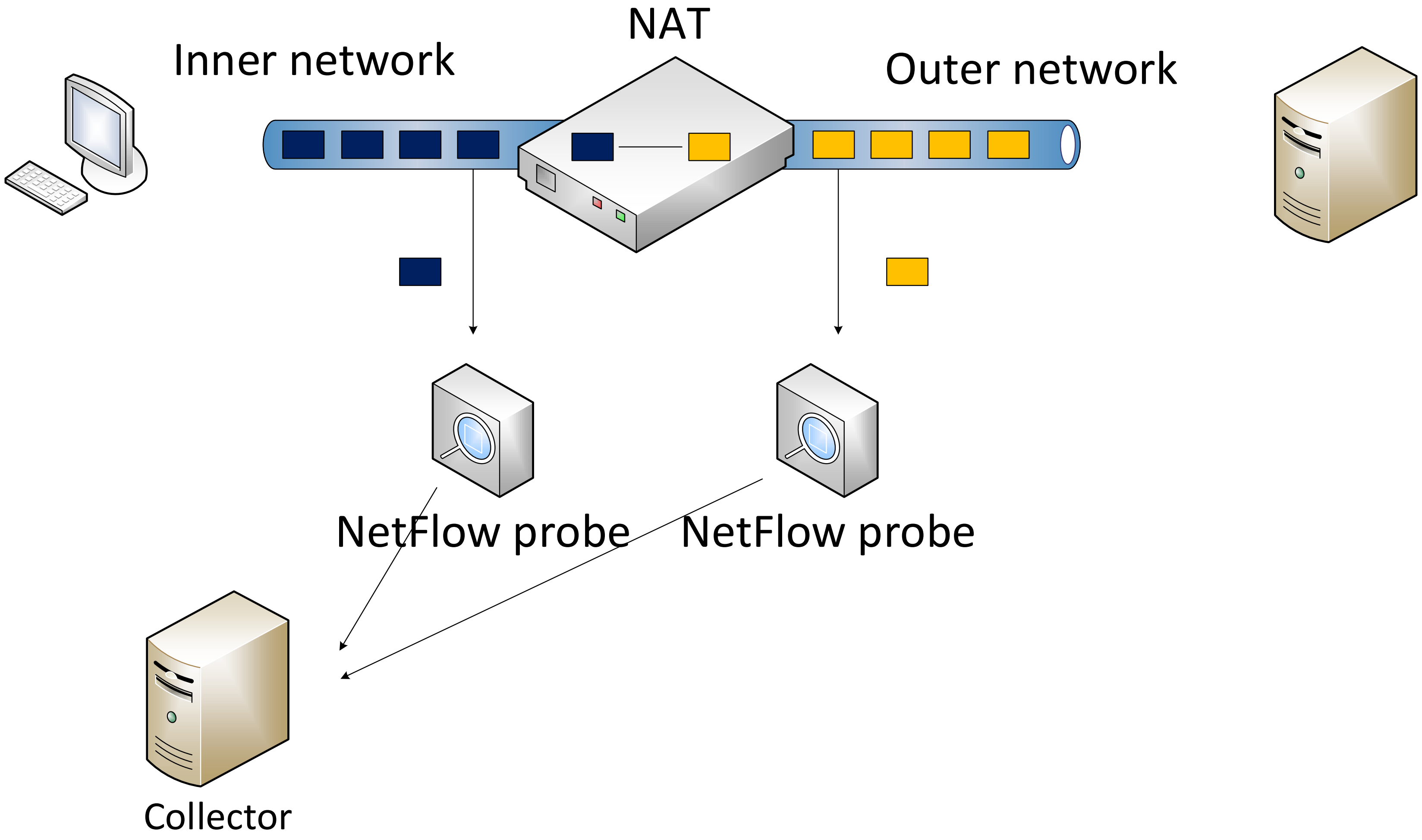


```
⊟ FlowSet 1
    FlowSet Id: (Data) (256)
    FlowSet Length: 44
  ⊟ Flow 1
      SrcAddr: 10.10.10.1 (10.10.10.1)
      Post NAT Source IPv4 Address: 192.168.1.10 (192.168.1.10)
      DstAddr: 8.8.8.8 (8.8.8.8)
      Post NAT Destination IPv4 Address: 8.8.8.8 (8.8.8.8)
      SrcPort: 9
      Post NAPT Source Transport Port: 1
      DstPort: 1
      Post NAPT Destination Transport Port: 1
      Ingress VRFID: 0
      Protocol: 1
      Nat Event: 1
      Observation Time Milliseconds: Jun  7, 2013 18:33:42.408000000 Central Europe Daylight Time
    Padding (2 bytes)
```

NAT – possible way to monitor traffic



NAT – possible way to monitor traffic



NAT – available information

- NetFlow probes don’t have access to translation table
- We have to create statefull information in stateless monitoring
- TCP – sequence number can be used
- UDP/ICMP – hash of the packet payload can be used

- Computation is done only with first packet of the flow

- The general idea: If a probe capture packet on a inner link, it should capture the packet on the outer link as well

Stateless monitoring, Statefull information

- Abstract idea: Interprocess/probes communication
  - Syncrhonization between monitoring processes or probes
- Possible solutions:
  - Interprocess communication – cannot be used for two separate probes
  - Shared memory – same as above
  - In memory database: Memcached, Redis
- Redis as in-memory key-value cache
  - Active development, library for most languages
  - Simple, fast, stable time complexity for queries
  - Records can have TTL – they are purged from database after period of time

NAT accounting – implementation

- Probe – concept tested using Flowmon probe from Invea-Tech
- Probes are extended with plugins:
  - Packet processing and parsing
  - Computation of statefull information
  - Saving to a cache
  - Export NetFlow data to a collector

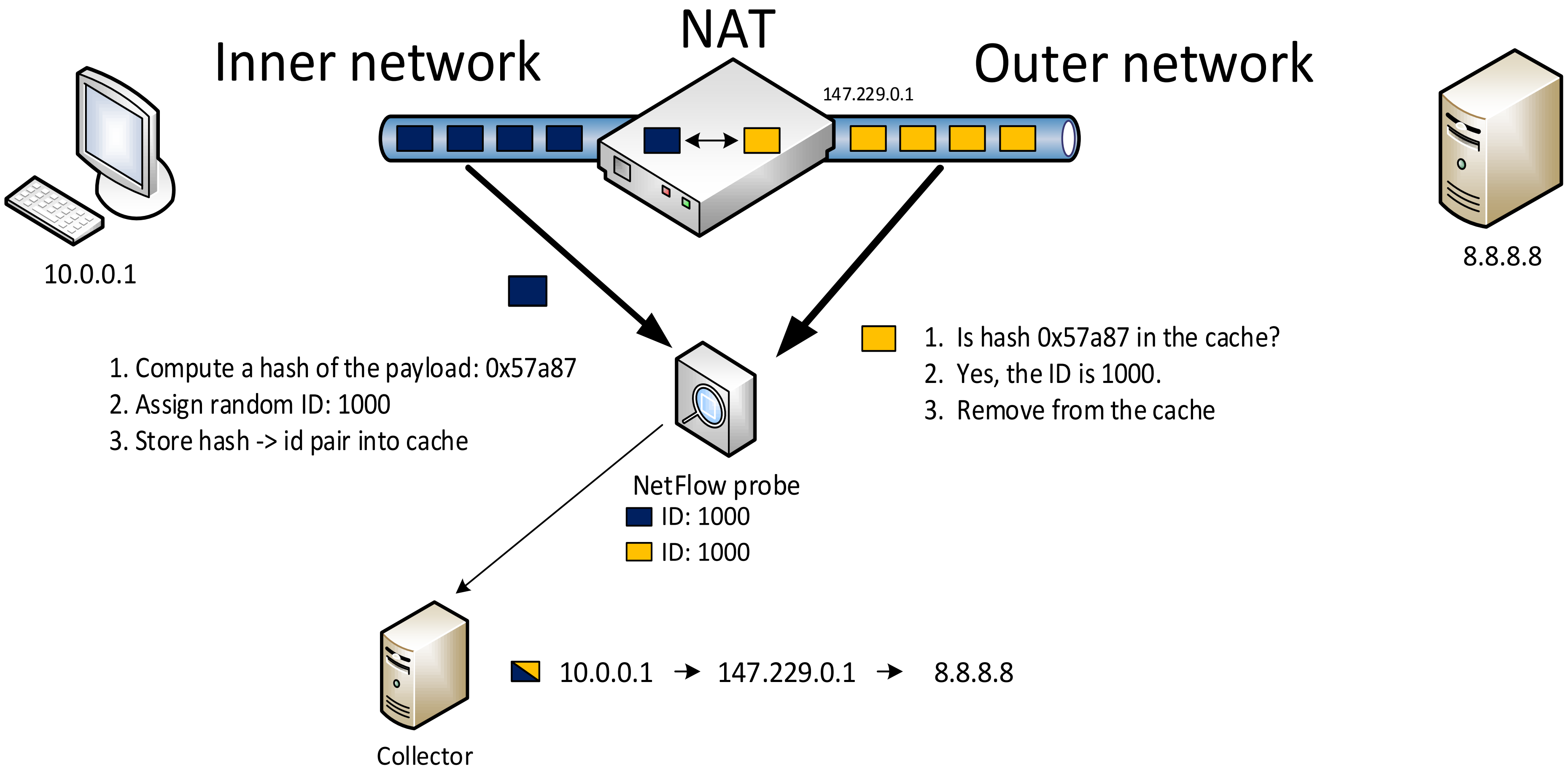
- ID is exported inside NetFlow data
  - IPFIX
  - A field in current NetFlow (ASN)

NAT – possible way to monitor traffic

🟦 Src IP address: 10.0.0.1:45987
Dst IP address: 8.8.8.8:53

→

🟨 Src IP address: 147.229.13.210:53234
Dst IP address: 8.8.8.8:53



🟦
1. Compute a hash of the payload: 0x57a87
2. Assign random ID: 1000
3. Store hash -> id pair into cache

🟨
1. Is hash 0x57a87 in the cache?
2. Yes, the ID is 1000.
3. Remove from the cache

NetFlow probe
🟦 ID: 1000
🟨 ID: 1000

10.0.0.1 → 147.229.0.1 → 8.8.8.8

Postprocessing on a collector

- NetFlow collector contains data before and after translation
- Flows before and after are combined based on ID value
- `Libnf` library is used for NetFlow data manipulation
- A flow with all information is created

Benchmark and testing

- Memory cache – basic benchmark shows 100 000 insert/delete operations/s
  - Without any optimization

- Without any sampling, all records are available
  - Can be used for fulfill data retention laws

Conclusion

- Users behind NAT can be monitored using NetFlow
- Performance is good enough even for large networks
- Collected NetFlow data contains all necessary information

BELGRADE SECURITY WORKSHOP 2015