Design of an IP Flow Record Query Language

Vladislav Marinov

Jacobs University Bremen

Outline

Outline

Example: Blaster Worm Infection

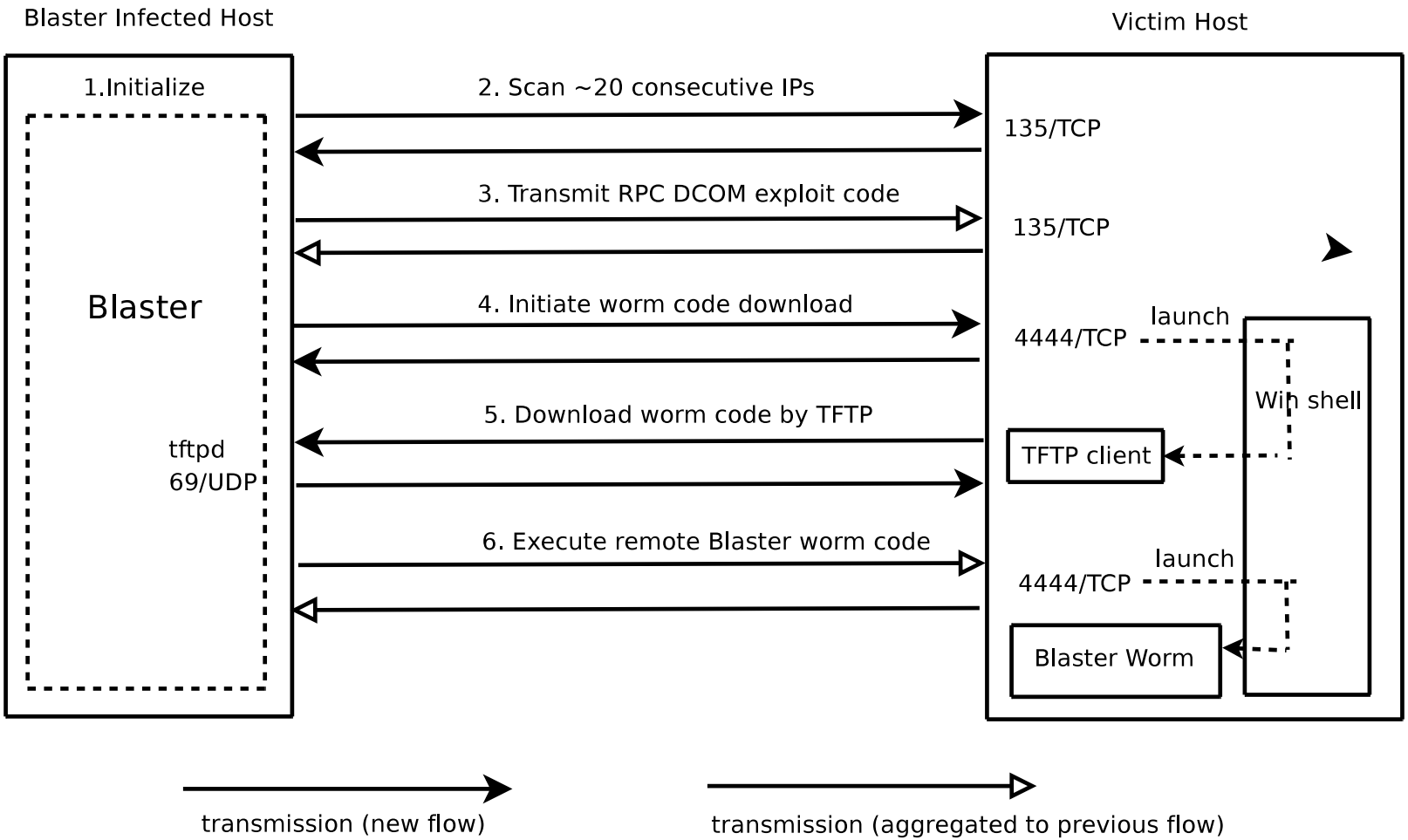


Figure: Packet-level breakdown of blaster infection

Example: Blaster Worm Infection

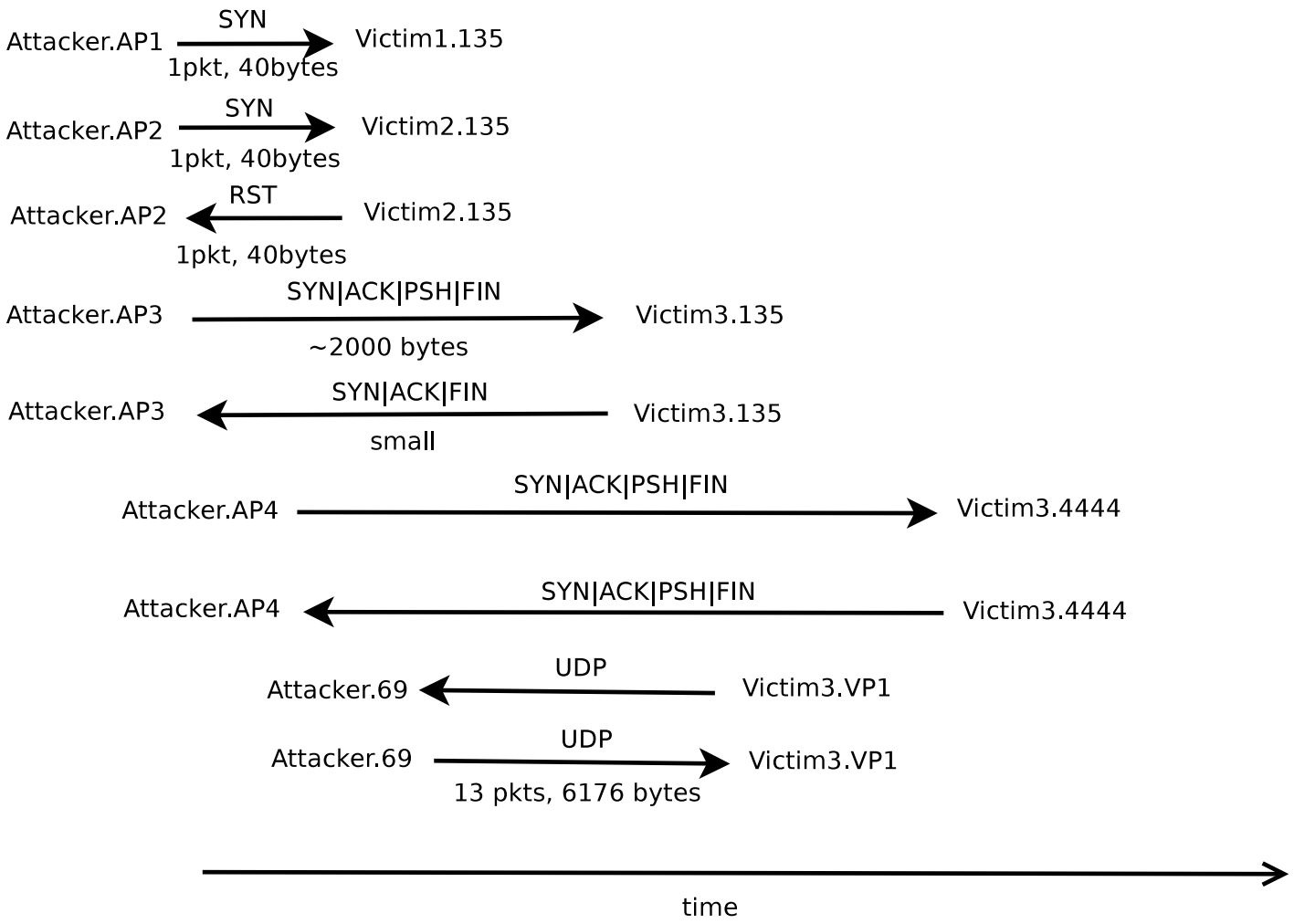


Figure: Flow-level breakdown of blaster infection

Outline

Existent Query Languages

Category	Query Language	Network Tools
SQL / stream languages	SQL, GSQL	B.Nickless et. al, B.Babcock et. al., Gigascope, Tribeca
Filtering languages	Berkeley Packet Filter (BPF), Access Control List (ACL)	`tcpdump`, `nfdump`, CoralReef, Time Machine, Flow-Tools, `ntop`
Procedural languages	Simple Ruleset Language (SRL), perl scripts, tool specific languages	NeTraMet, Flow-Tools FlowScan, Stager AutoFocus, SiLK

- SQL-based query languages lead to poor query performance when storing flow data in a DBMS
- Filtering query languages lack a time and concurrency dimension
- Script-based query languages are powerful but not trivial to understand
- Existent query languages cannot describe traffic patterns composed of a set of flows that have causal dependencies

Problem Statement and Approach

Problem Statement

- Describe and identify the occurrence of network traffic patterns in a collection of flow records
- A flow record query language is needed
- Existent query languages are not suitable for describing complex traffic patterns

Approach

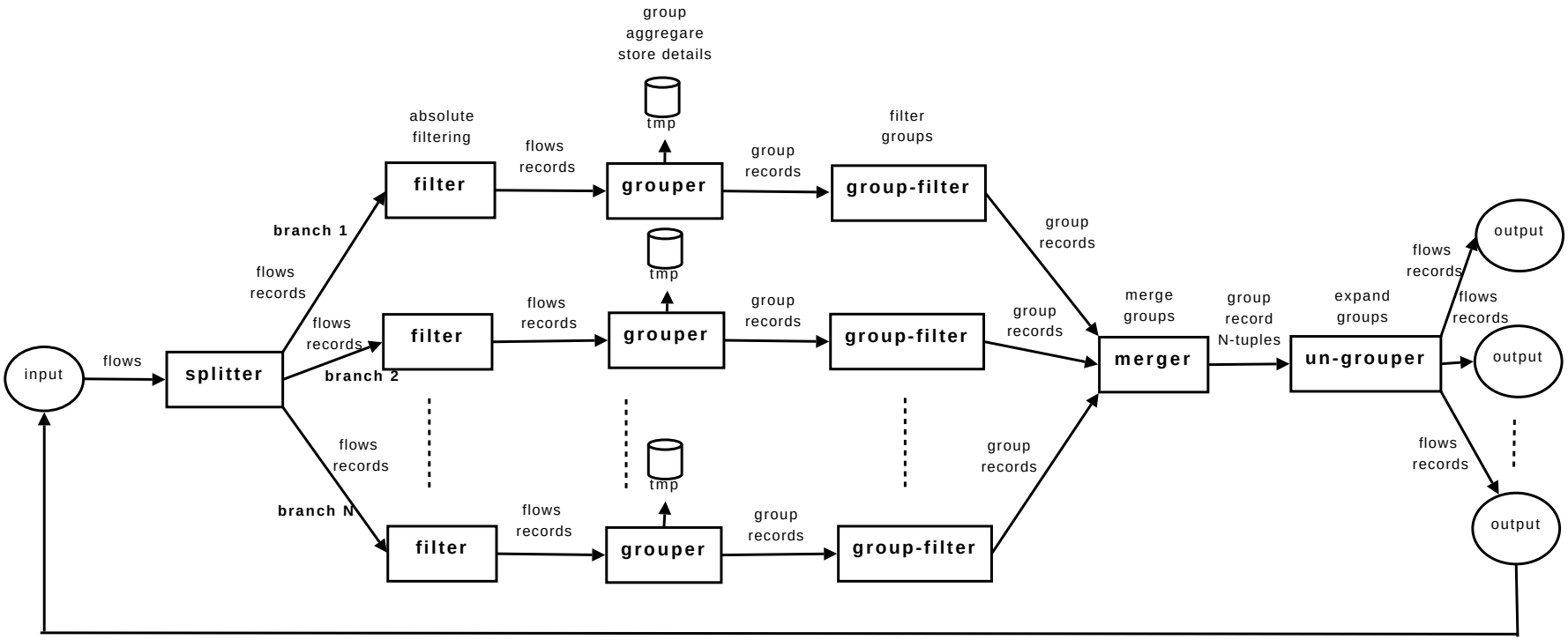
We propose a new IP flow record query language to describe network traffic patterns in a declarative and easy to understand way

- A comprehensible set of language primitives will be built
- The flow patterns of some common network services and application will be derived

Outline

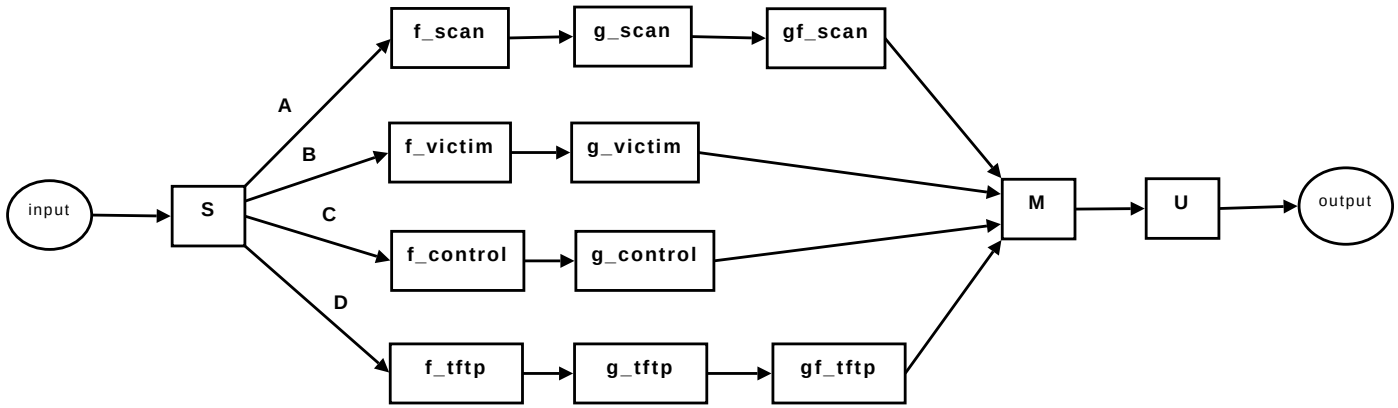
Framework for IP Flow Filtering

- Stream-based approach with several operators
- Primitives to express timing and concurrency relationships
- Primitives to define correlation and dependencies among flow attributes



Outline

Example: Blaster Worm Infection

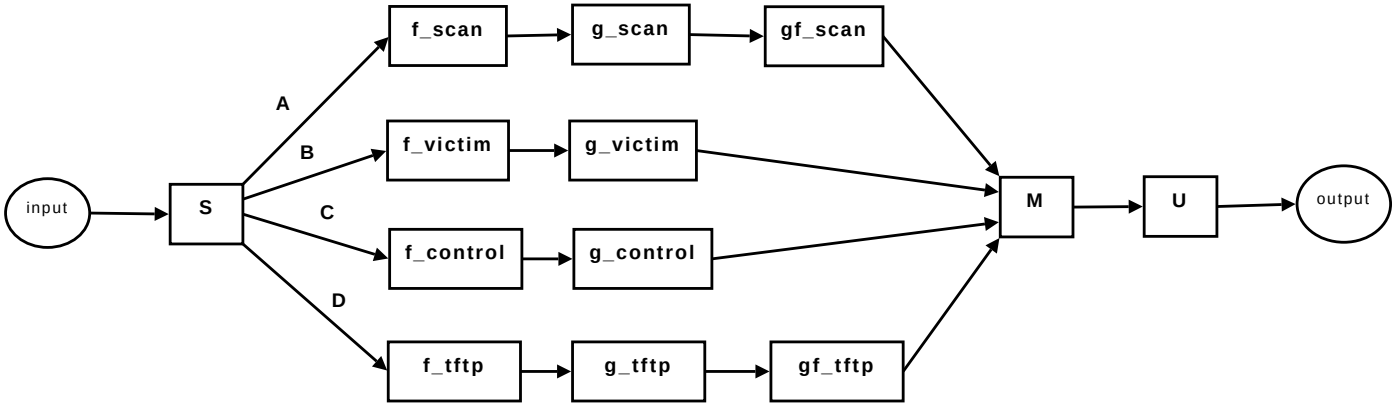



```
filter f_scan {
  dstport = 135
  proto = tcp
  flags = S
}

group-filter gf_scan {
  count > 20
}
```

```
grouper g_scan {
  srcip = srcip
  dstip = dstip relative-delta 1
  stime = stime relative-delta 5ms
  stime = stime absolute-delta 5s
  aggregate srcip, union(dstip),
            min(stime),max(etime),
            count
}
```

Example: Blaster Worm Infection




```
merger M {
    A.srcip = B.srcip
    A.srcip = C.srcip
    A.srcip = D.dstip
    B.dstip = C.dstip
    B.dstip = D.srcip
    B.dstip in union(A.dstip)
    A < B OR A m B OR A o B
    B o C
    D d C
}
```

References

T. Dübendorfer, A. Wagner, T. Hossmann and B. Plattner.

Flow-level Traffic Analysis of the Blaster and Sobig Worm Outbreaks in an Internet Backbone.
In *Proc. of DIMVA'05*. Springer LNCS 3548, July 2005.

V. Marinov and J. Schönwälder.

Design of an IP Flow Record Query Language.
In *Proc. of AIMS'08*. Springer LNCS 5127, July 2008.

Query Language Requirements

- **Filtering** and **Aggregation**- filter and aggregate on all flow record attributes
- **Dependency**
  - *correlation* - match flows where the source IP address of one flow is the same as the destination IP address of another flow.
  - *time dependencies* - time window, delay between start and end times of flows, duration of a flow, flow concurrency etc.
  - *flow order*
  - *existence or non-existence of specific flows*
  - *causal dependencies between flow attributes* - sequence of port numbers or IP addresses.