Self-Modulating Endpoint Observability

—

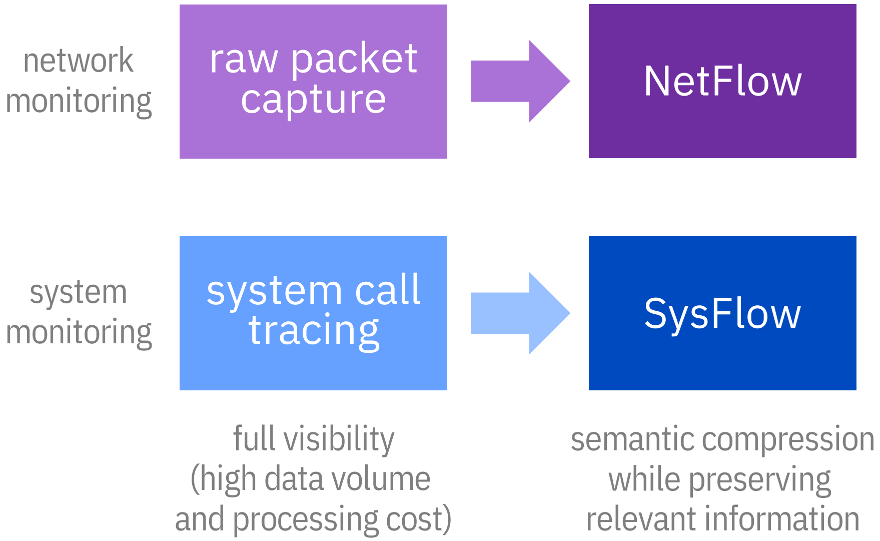
Fred Araujo and **Teryl** Taylor

IBM **Research**

FloCon 2020

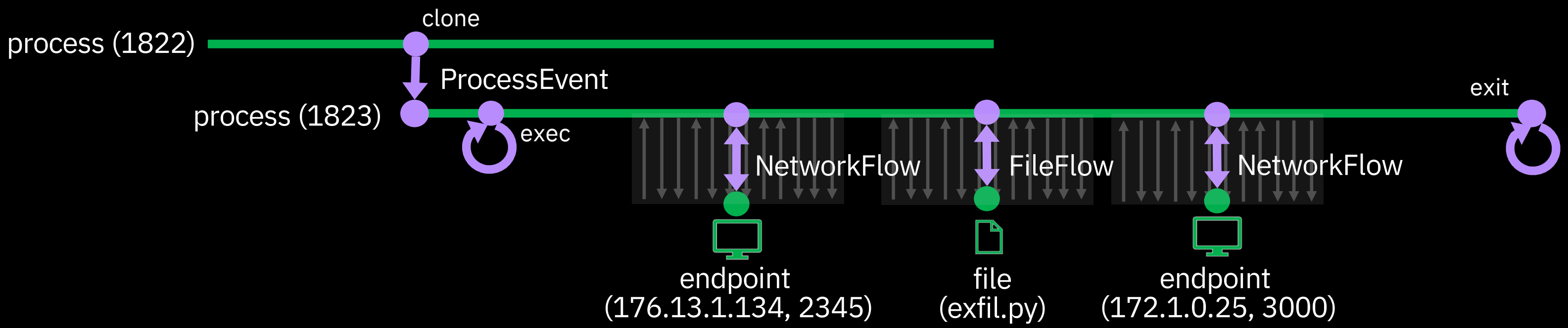
SysFlow is open sourced!

- “NetFlow” for system events
- Captures **process** control flows, **file** interactions, and **network** communications
- Container-aware, flow-centric semantics for system analytics

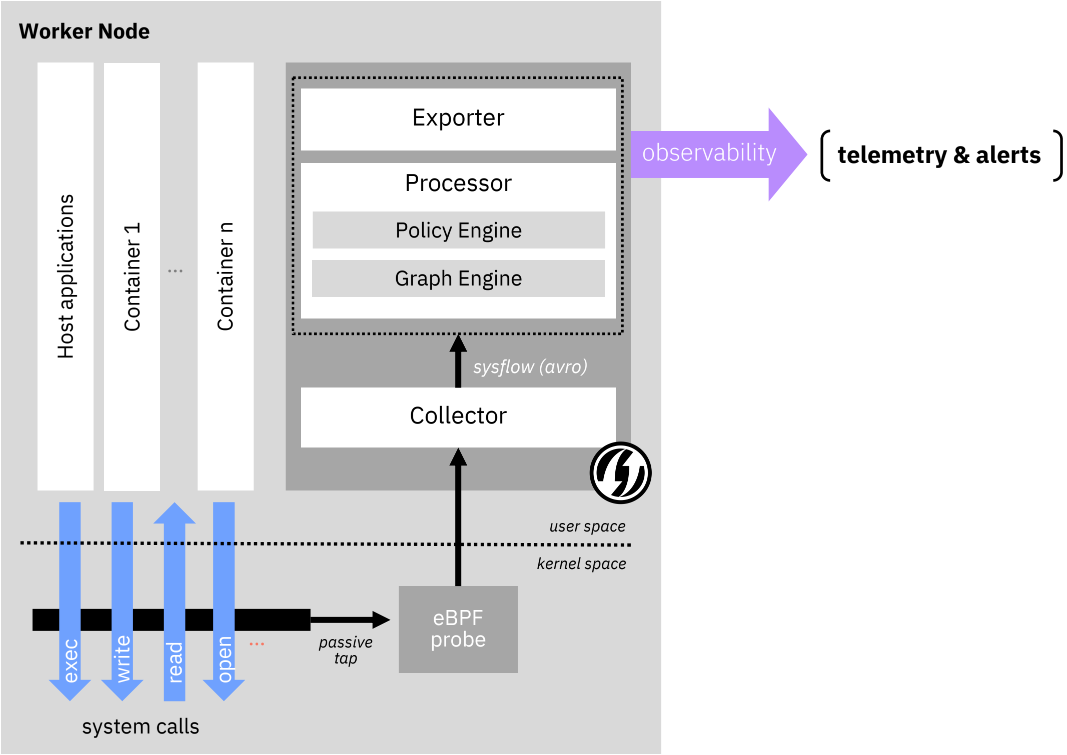

network monitoring
raw packet capture
NetFlow
system monitoring
system call tracing
SysFlow
full visibility (high data volume and processing cost)
semantic compression while preserving relevant information

Data science on system telemetry made easier!

“Semantically compressed system events for scalable security, compliance, and performance analytics.”

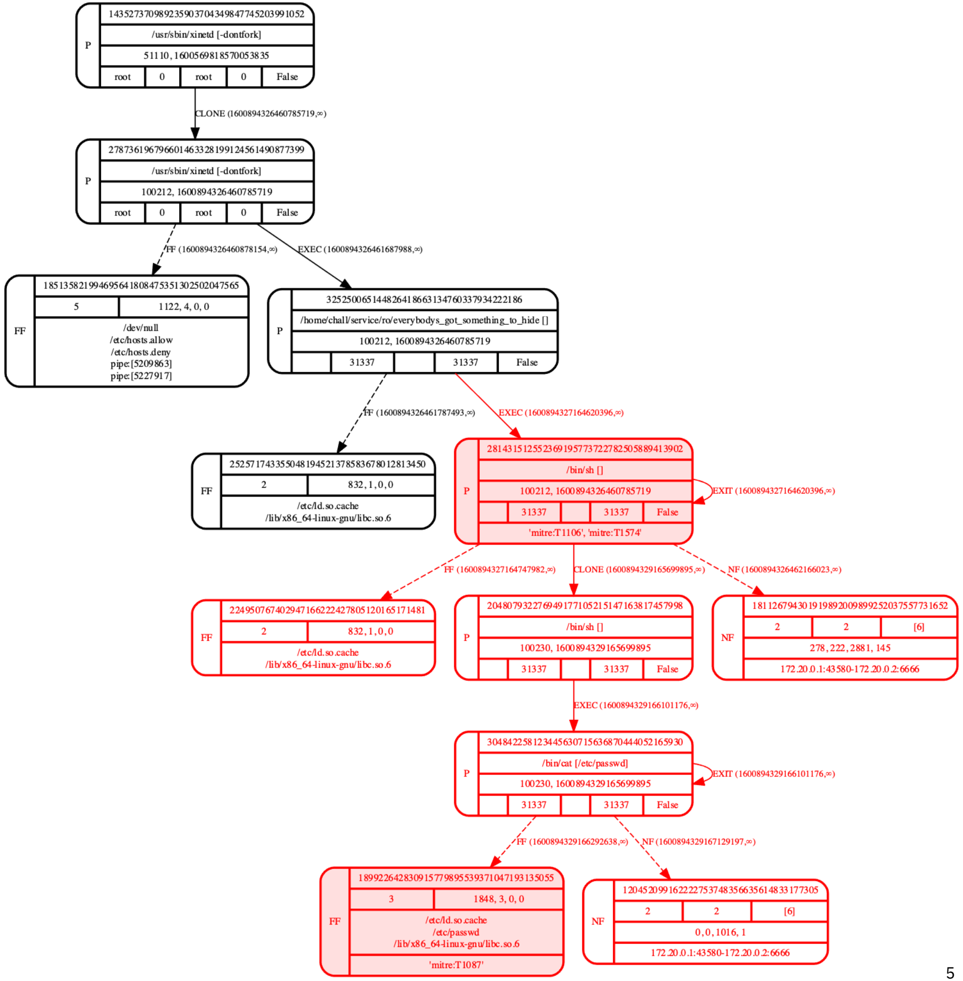


Last year, we brought you the processor, enabling edge analytics



…and introduced graphlets with TTP tagging

- MITRE ATT&CK TTP tagging
- Behavior coalescing
- Attack kill chain interpretation



Today, we'll discuss how to combine these technologies to help reduce event fatigue.

Rate limiting system events – **WHY?**

– SIEMs charge by ingestion rates (eps)

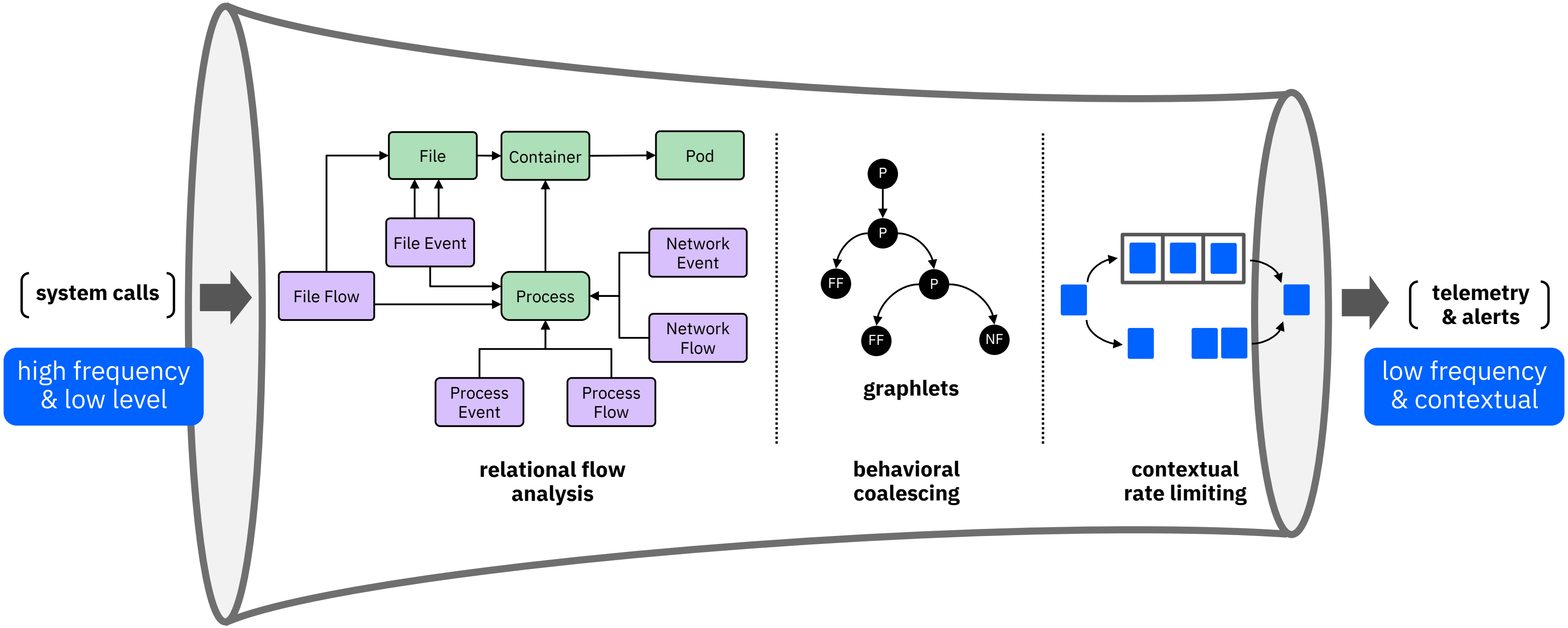
– Excessive/redundant alerts

- Event tuning is an expensive manual process

– Reduced resource usage for alert/policy engines

- Lack of event context is an issue

System telemetry stream modulation



system calls

high frequency & low level

relational flow analysis

graphlets

behavioral coalescing

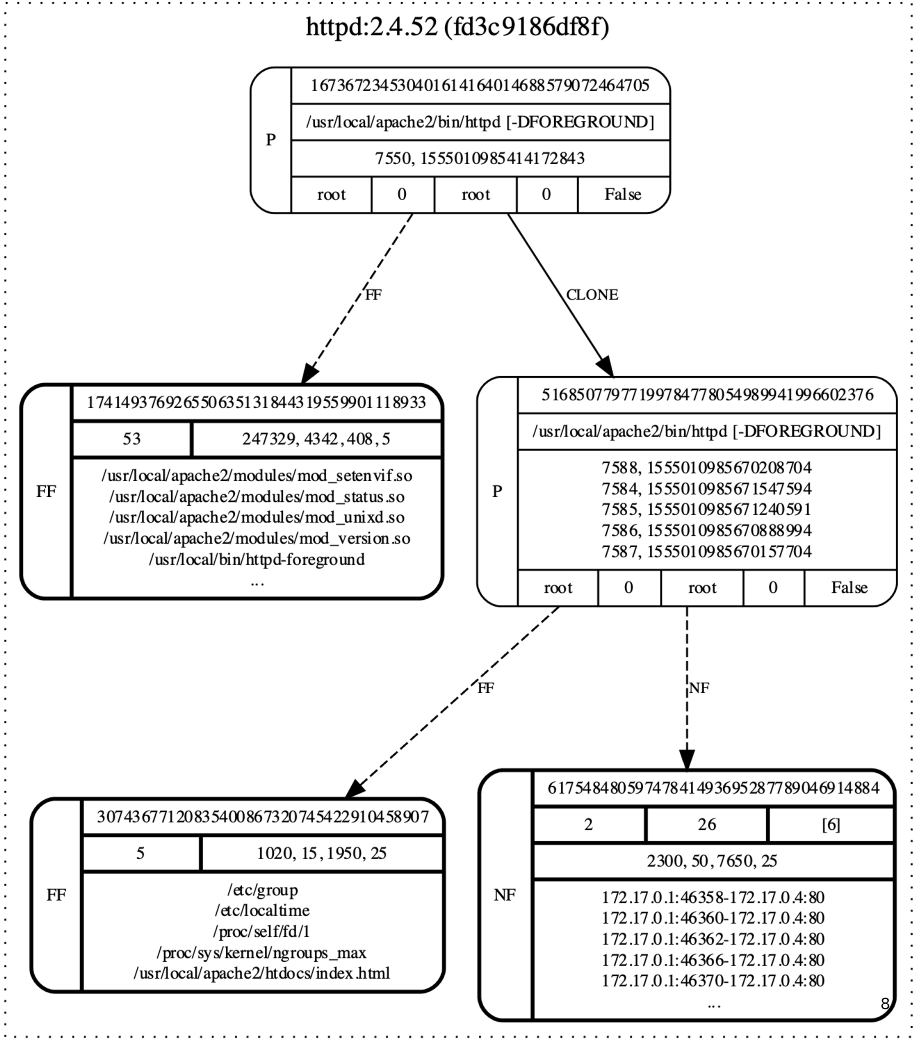
contextual rate limiting

telemetry & alerts

low frequency & contextual

Graphlets

- Provide context for deciding when to forward records downstream
- Coalesce process events, file flows, and network flows based on control flow path
- Labeled direct graph
    - Events are labeled edges that associate two process nodes inhabited by their corresponding process instances
    - Flows associate a process node to file and network flow nodes that summarize filesystem and network activity



httpd:2.4.52 (fd3c9186df8f)

P
167367234530401614164014688579072464705
/usr/local/apache2/bin/httpd [-DFOREGROUND]
7550, 1555010985414172843
root | 0 | root | 0 | False

FF
174149376926550635131844319559901118933
53 | 247329, 4342, 408, 5
/usr/local/apache2/modules/mod_setenvif.so
/usr/local/apache2/modules/mod_status.so
/usr/local/apache2/modules/mod_unixd.so
/usr/local/apache2/modules/mod_version.so
/usr/local/bin/httpd-foreground
...

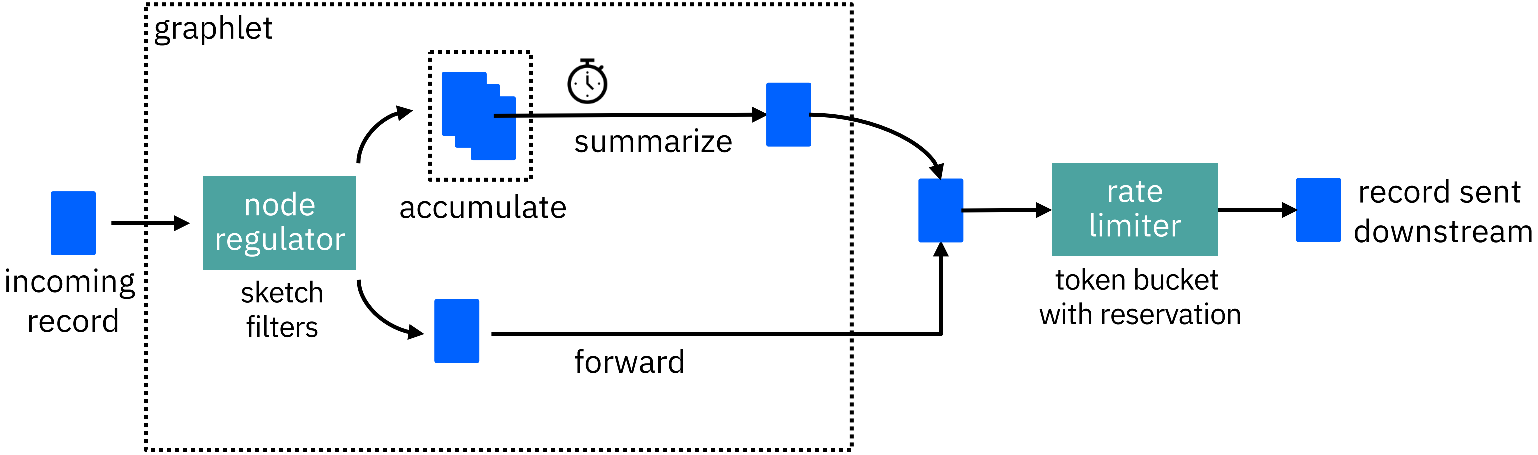
P
51685077977199784778054989941996602376
/usr/local/apache2/bin/httpd [-DFOREGROUND]
7588, 1555010985670208704
7584, 1555010985671547594
7585, 1555010985671240591
7586, 1555010985670888994
7587, 1555010985670157704
root | 0 | root | 0 | False

FF
30743677120835400867320745422910458907
5 | 1020, 15, 1950, 25
/etc/group
/etc/localtime
/proc/self/fd/1
/proc/sys/kernel/ngroups_max
/usr/local/apache2/htdocs/index.html

NF
61754848059747841493695287789046914884
2 | 26 | [6]
2300, 50, 7650, 25
172.17.0.1:46358-172.17.0.4:80
172.17.0.1:46360-172.17.0.4:80
172.17.0.1:46362-172.17.0.4:80
172.17.0.1:46366-172.17.0.4:80
172.17.0.1:46370-172.17.0.4:80
...

Rate modulation

- **Regulators** semantically reduce the telemetry stream by minimizing **heavy hitters** and **scanners** (spatial dispersion)
- **Rate limiter** modulates the output stream to minimize event **bursts** and enforce a maximum output rate



Node-level regulators

– Nodes use sketch data structures for deciding when to immediately forward records

- Flow nodes use tries for summarization

HyperLogLog sketch

– Approximates the number of distinct items in a multiset

– Intuition: Cardinality of uniform distributed numbers can be approximated by the maximum number of leading 0’s in the binary representation of each number in the set

– Represent set of 10^9 in 1.5KB

Count-min sketch

– Probabilistic frequency table of events

– 2-D Array M[w cols x d rows]

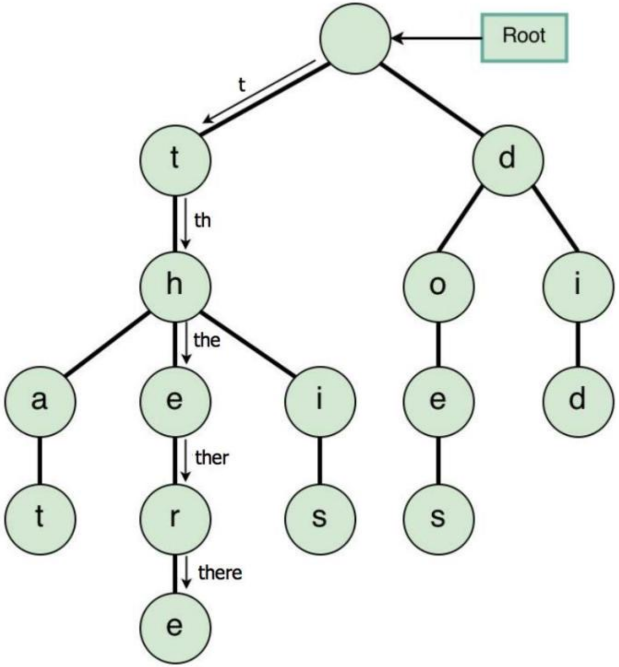
– Event type: i, d hashes

- index j,k = hj(i) M[j,k]++
- aj = min count [j, hj(i)]

Curbing file access explosion

Tries

- Search tree where keys are embedded as nodes
- Search time: O(m) where m is length of search key
- File flow nodes use path tries (filesystem paths separated by “/”) to aggregate file flow instances at each node of the trie matching accessed files

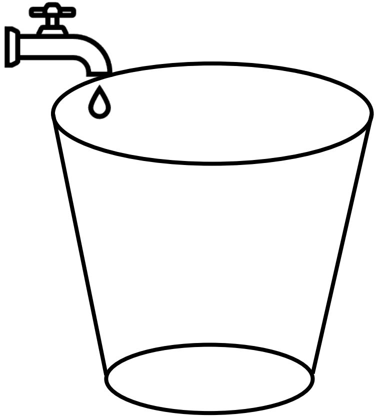


https://theoryofprogramming.wordpress.com/2015/01/16/trie-tree-implementation/

Rate Limiting

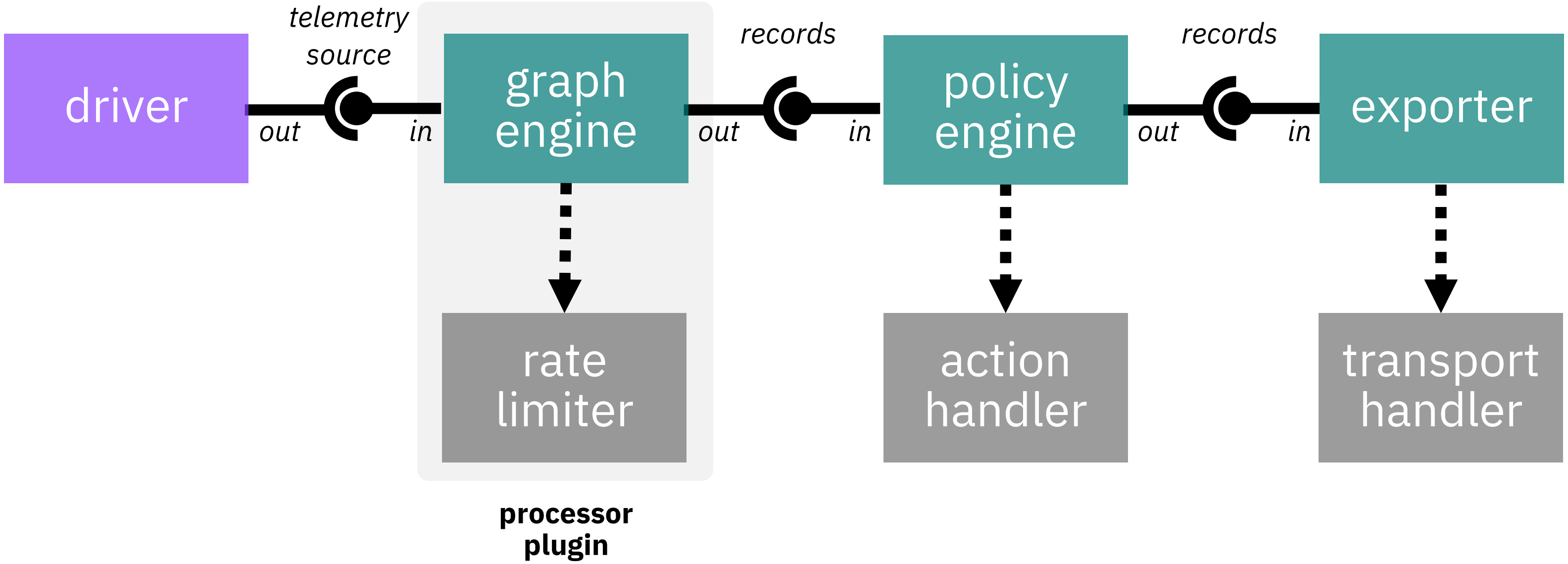
Token-bucket limiter

- Tokens added to bucket at rate 1/r
- Bucket holds maximum of b tokens
- Deciding whether to forward event *i*:
  - If token available, it’s removed and *i* is forwarded
  - If no token, event buffered in queue of size *n*
    - If queue full, event *i* dropped
  - Queue emptied by reserving tokens



Implementation

- Uses the SysFlow plugin system
- Custom edge processing pipeline



K8s Benchmark

K8s cluster

- 12 worker nodes
- monitoring host and container pods across all namespaces during regression and pentesting
- Duration: 100 min
- Contains infrastructure and user pods

Metrics

- Forwarded: # of events immediately forwarded
- Accumulated: # of events aggregated
- Reduced: # of reduced events forwarded
- Alerts: # of alerts exported based on MITRE ATT&CK TTP policy

Metric	**SysFlow** **without rate modulation**	**SysFlow** **with rate modulation**
Forwarded	4,124,549	27,814
Accumulated	-	3,568,889
Reduced	-	44,905
Events (F+R)	4,124,549 (687 eps)	72,719 (12 eps)
Alerts (TTPs)	108,546 (18 aps)	2,853 (0.5 aps)

Observations

- No event drops; rate limiting handles event bursts
- Stream modulation drastically curbs the number of duplicate alerts while preserving unique behaviors
- Reduced resource usage for alert/policy engine

SysFlow Project

- Open source
  github.com/sysflow-telemetry
- Growing set of APIs
  Python, C/C++, Go, ...
- Non disruptive and easily deployable
  helm, oc, docker, and ansible deployments

Thank you

github.com/sysflow-telemetry

—

sysflow-telemetry.slack.com
sysflow@us.ibm.com

IBM