ISOVALENT

Application Network Security

In an Encrypted Future

John Fastabend, Isovalent

Agenda

- Tetragon - eBPF-based Security Observability & Runtime Enforcement
- L7 Observability and Security Today
    - Security Use Cases
- Encryption and Zero Trust broke my tooling
- BPF to the Rescue
- KTLS and SK_MSG Demo
- Whats Next

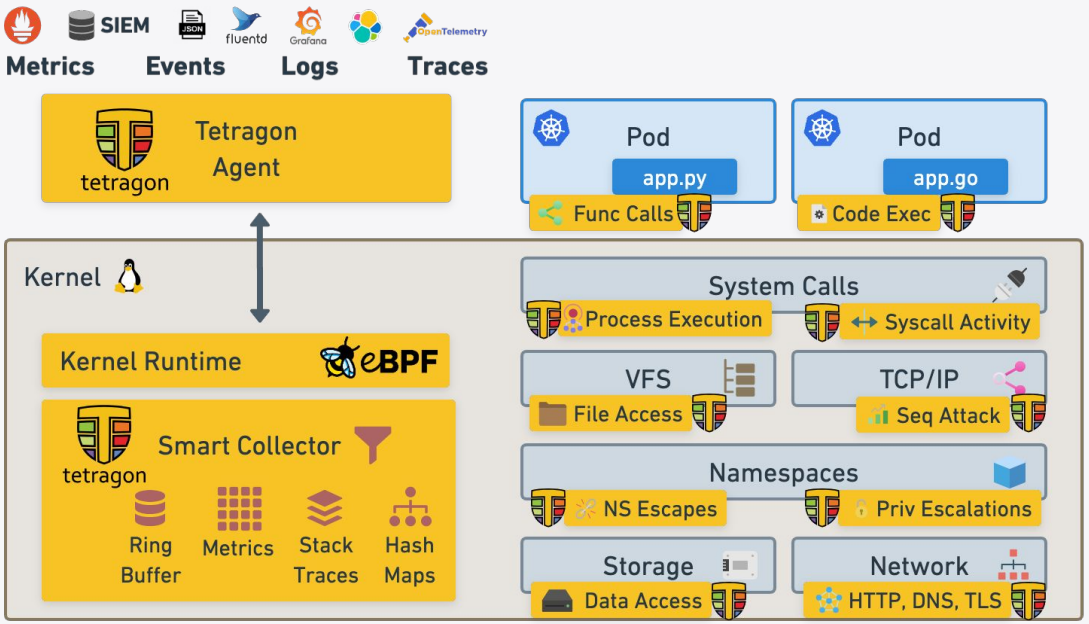
ISOVALENT

What is Tetragon?

Tetragon - Security Observability & Runtime Enforcement

Why is it so powerful?

- **Synchronous BPF monitoring**, **filtering** & **enforcement** completely with eBPF.
- **Overall efficiency focus**, team and users CPU/memory obsessed.
- **Kubernetes Identity Aware in BPF** teaches kernel what K8s pods, namespaces and labels.
- **Hook arbitrary kernel functions** through yaml and not just syscalls.
- **eBPF-based inline Enforcement** allows blocking actions and killing process via SIGKILL.

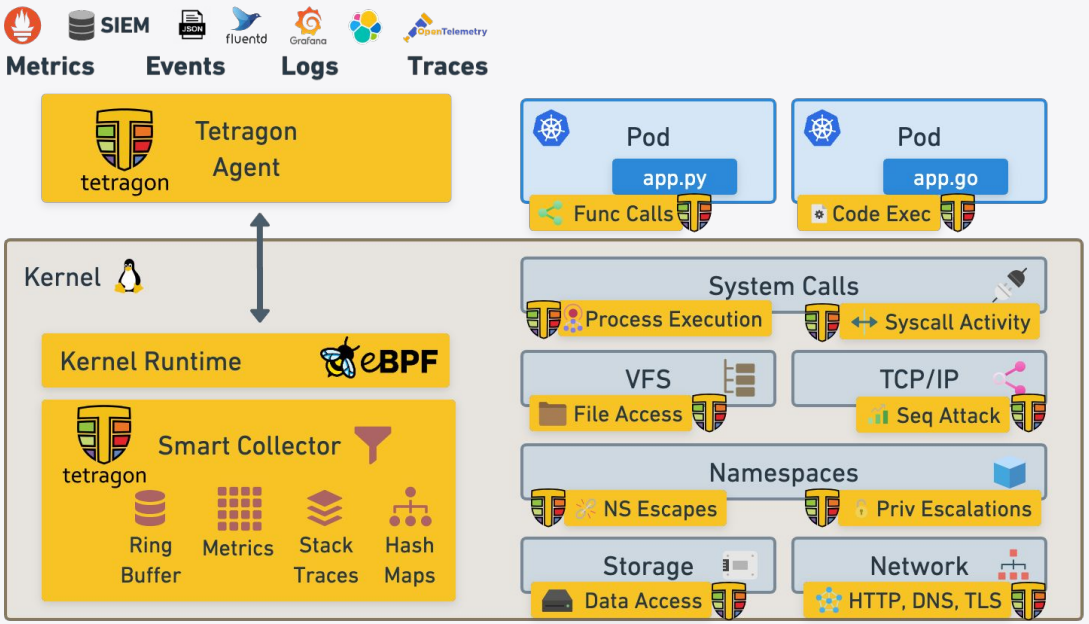


ISOVALENT

Tetragon - Security Observability & Runtime Enforcement

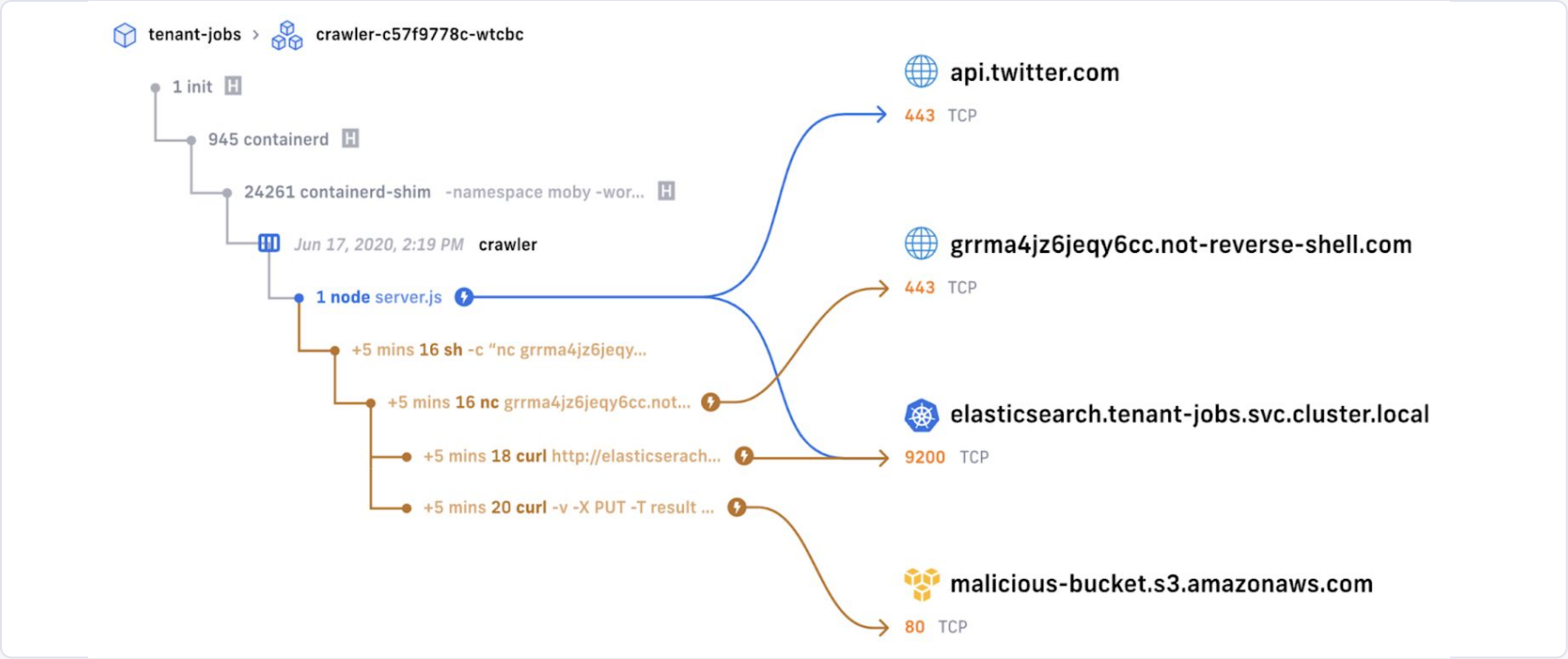
Deep Visibility

- Process executions & System call activity
- L3/L4 network connections
- Data & File Access
- Linux namespace changes
- Linux capabilities & privilege changes
- CVE mitigations
- SSH/bash capture and replay
- **L7 parsing with sockmap**
- DNS, HTTP, TLS, ...



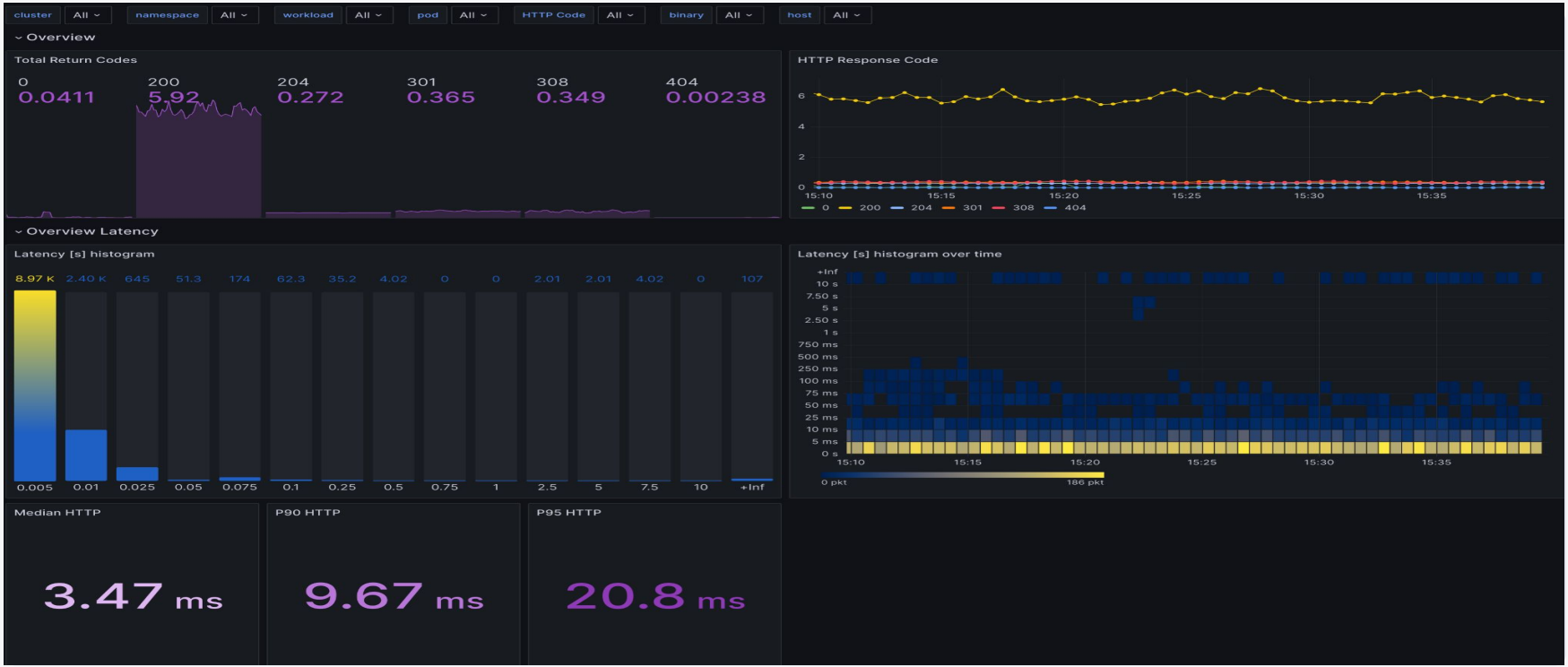
ISOVALENT

Core Tetragon Use Case



ISOVALENT

HTTP Observability



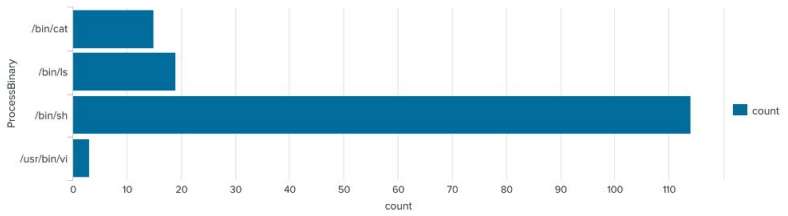
ISOVALENT

Monitoring File Access

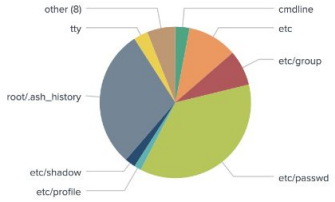
Sensitive File Open

Edit | Export | ...

Binaries (coreapi)



File Names (coreapi)

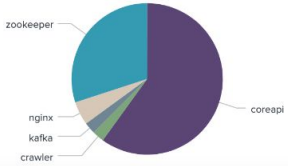


Sensitive File Open (coreapi)

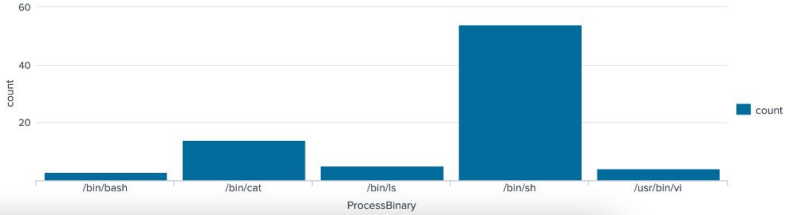
StartTime	SourceNamespace	SourcePod	SourceImage	ProcessBinary	FileName	count
2021-11-22T18:37:33.639Z	tenant-jobs	coreapi	quay.io/isovalent/jobs-app-coreapi:latest	/bin/sh	etc/passwd	6
2021-11-22T18:37:33.639Z	tenant-jobs	coreapi	quay.io/isovalent/jobs-app-coreapi:latest	/bin/sh	opt/app	1
2021-11-22T18:37:33.639Z	tenant-jobs	coreapi	quay.io/isovalent/jobs-app-coreapi:latest	/bin/sh	root/.ash_history	3
2021-11-22T18:46:41.992Z	tenant-jobs	coreapi	quay.io/isovalent/jobs-app-coreapi:latest	/usr/bin/vi	etc/shadow	1
2021-11-22T18:52:04.182Z	tenant-jobs	coreapi	quay.io/isovalent/jobs-app-coreapi:latest	/bin/cat	etc/shadow	1

« Prev 1 2 3 4 5 6 7 8 9 Next »

/etc/passwd (by SourcePod)



/etc/passwd (by SourceBinary)

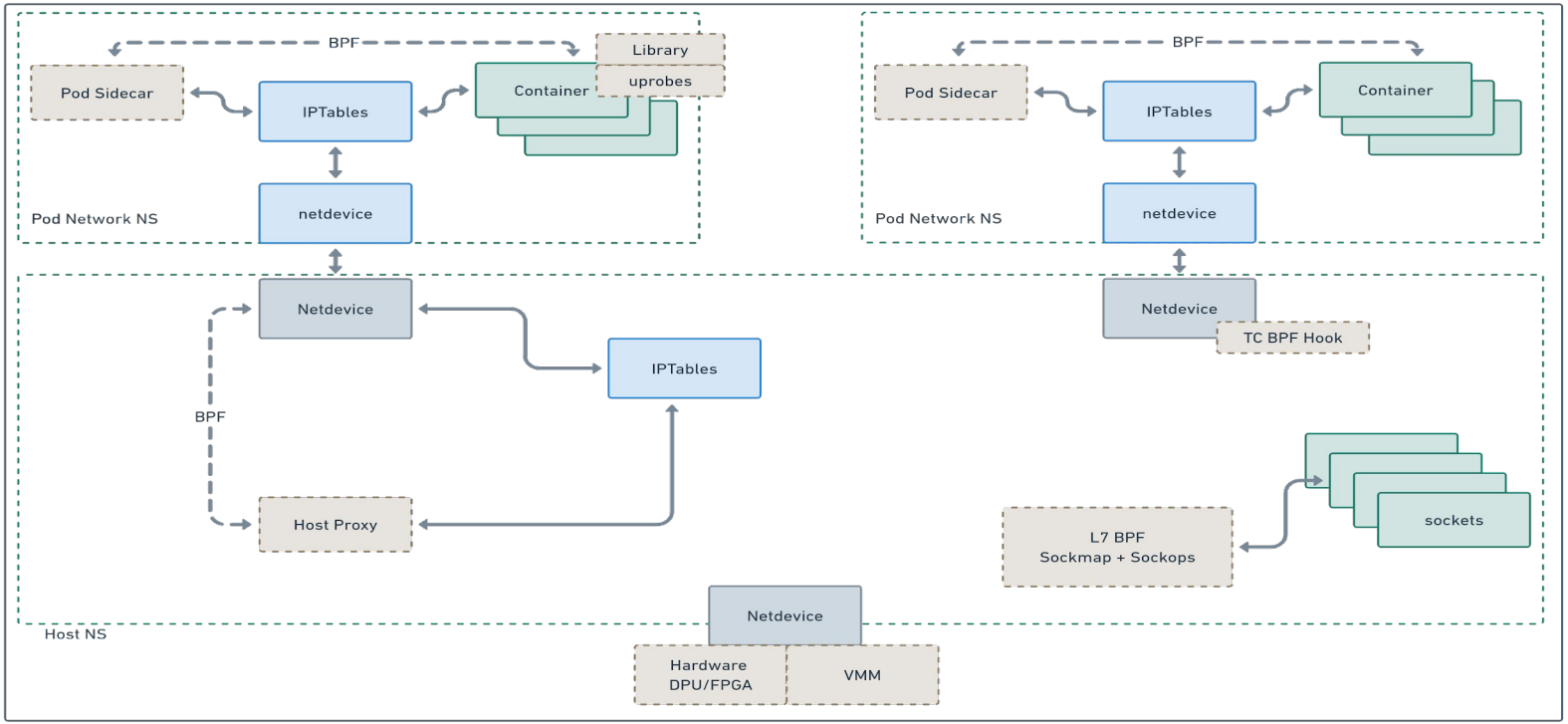


ISOVALENT

ISOVALENT

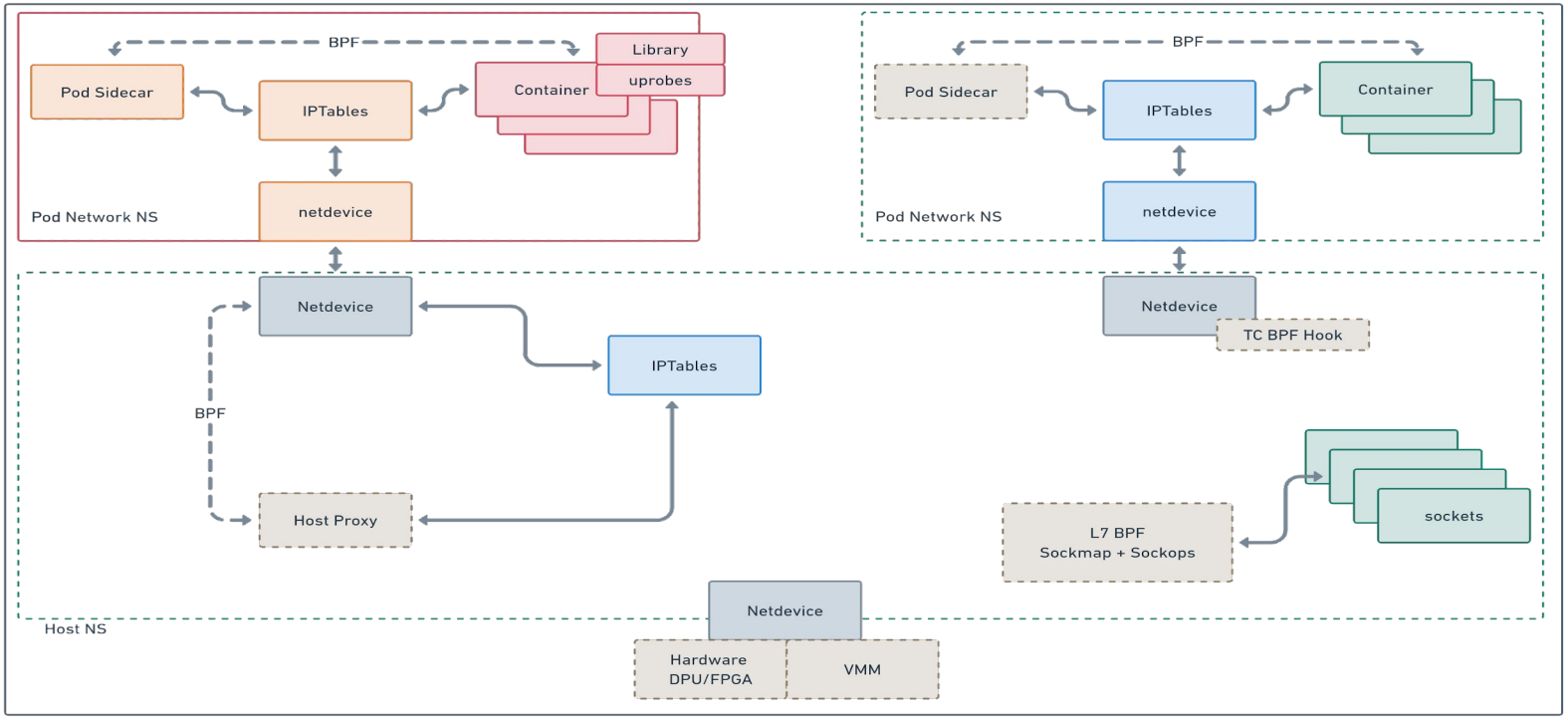
Tetragon: Trust Model

L7 Observability Hook Points



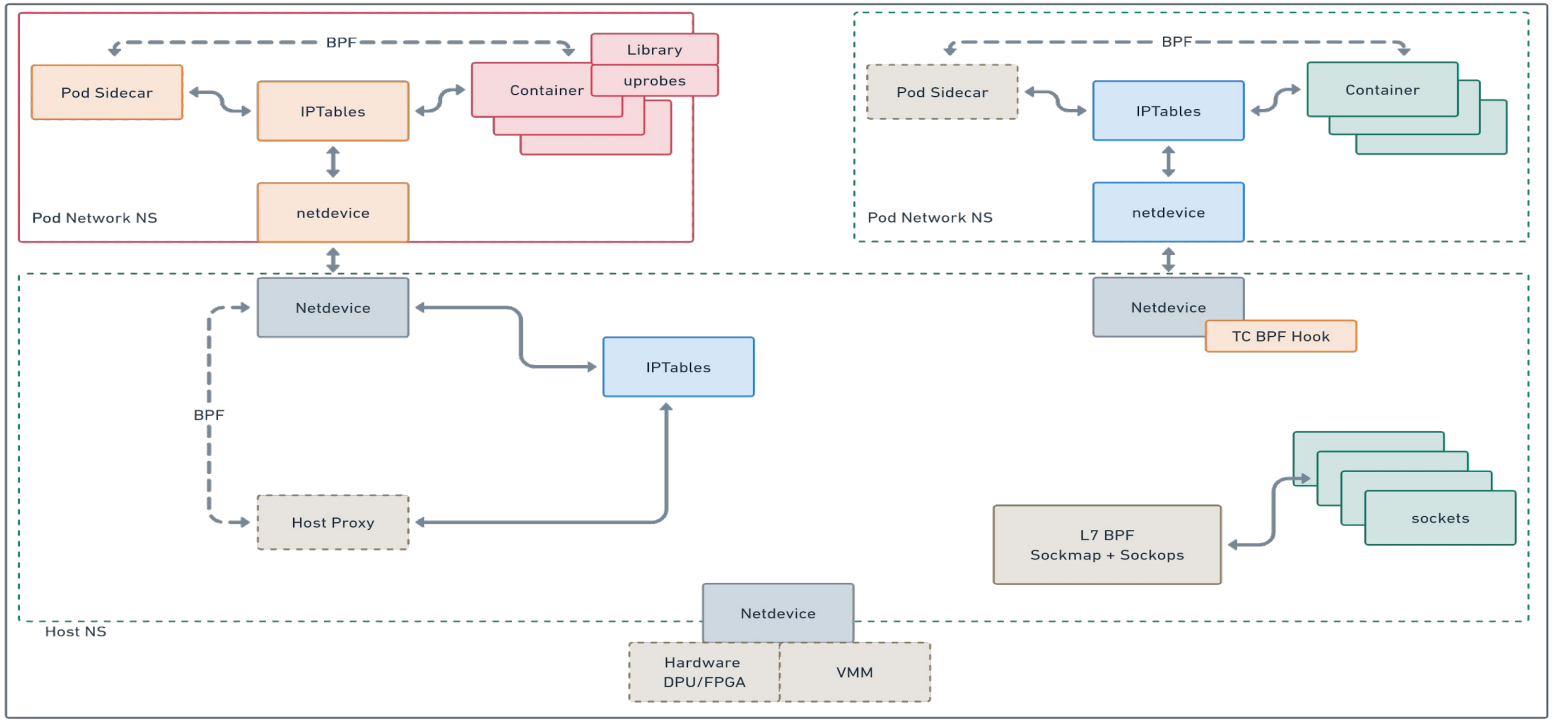
ISOVALENT

L7 Observability Trust Model



ISOVALENT

L7 Observability Viable Hook Points

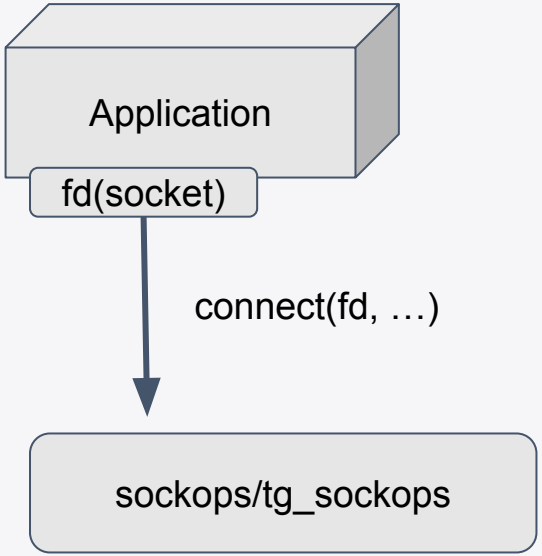


ISOVALENT

ISOVALENT

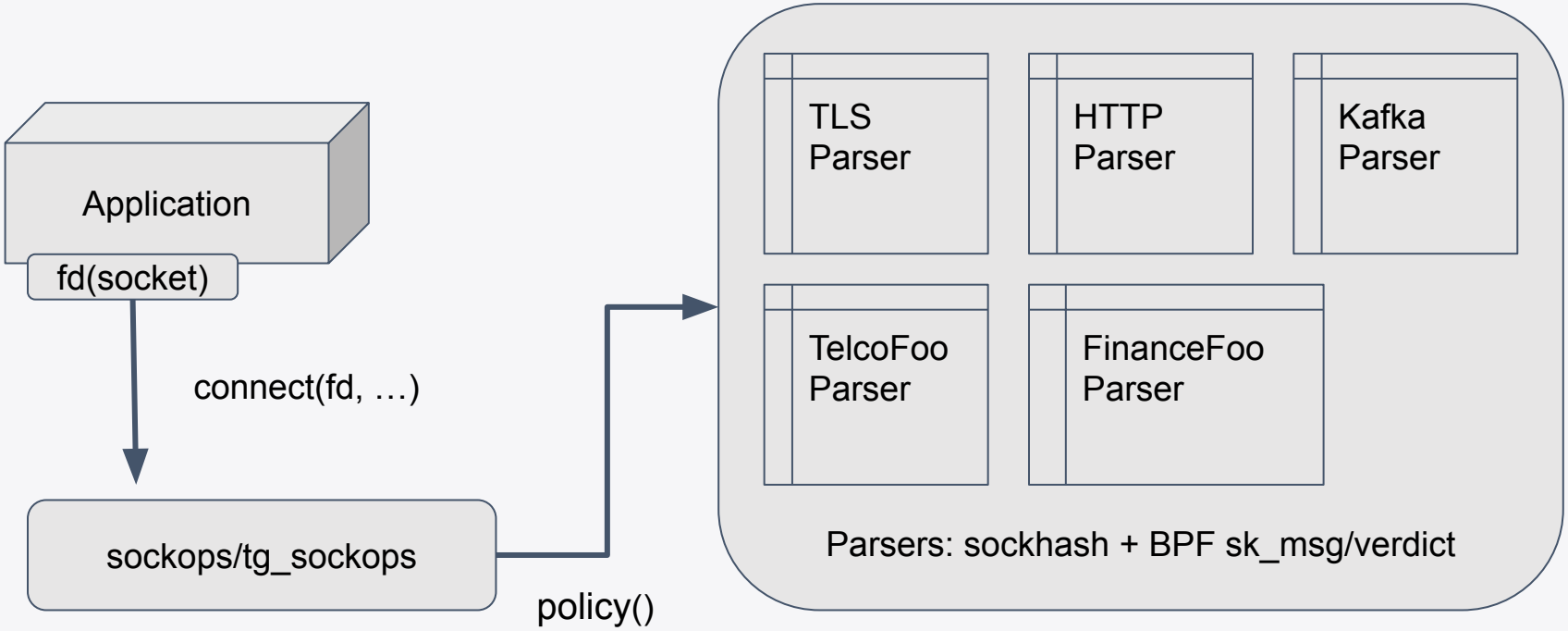
L7 Observability
SK_MSG, SockOps, Sockhash

L7 Observability: sockops

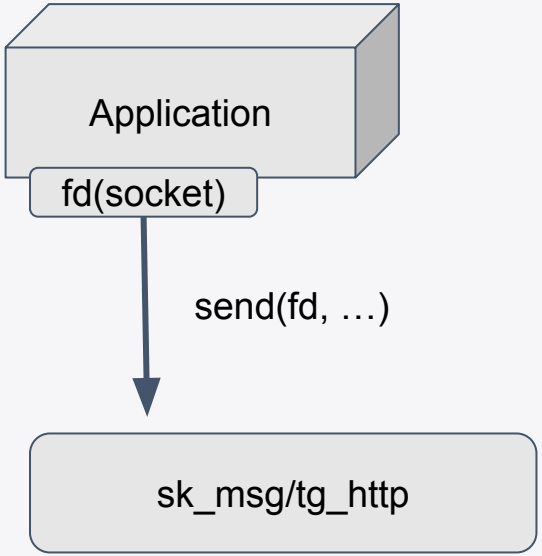



```
switch (op) {
  case BPF_SOCK_OPS_PASSIVE_ESTABLISHED_CB:
  case BPF_SOCK_OPS_ACTIVE_ESTABLISHED_CB:
        if (family == AF_INET || family == AF_INET6)
              enable_parser(skops);
        break;
  default:
        break;
  }
```

L7 Observability: sockops

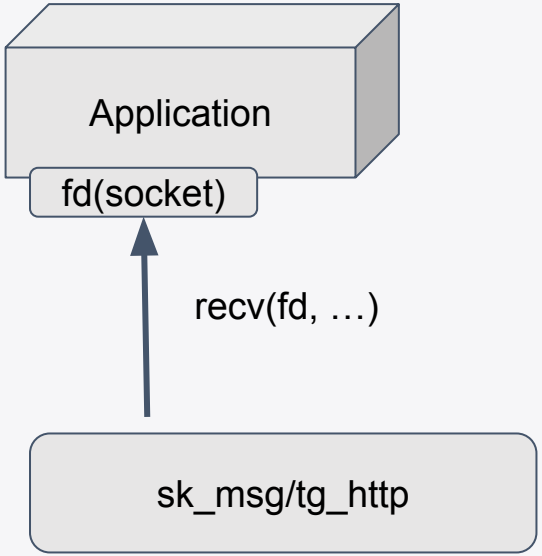


L7 Observability: sk_msg



- SK_MSG:
  - BPF socket hook operates after copy user to kernel space to avoid TOCTTOU.
  - Before TCP/UDP/… stack so program “sees” data sg list not a packet or skb.
  - Helpers to drop, redirect, cork, skip data.
  - Supports sendpage maintaining TOCTTOU results in overhead.

L7 Observability: verdict

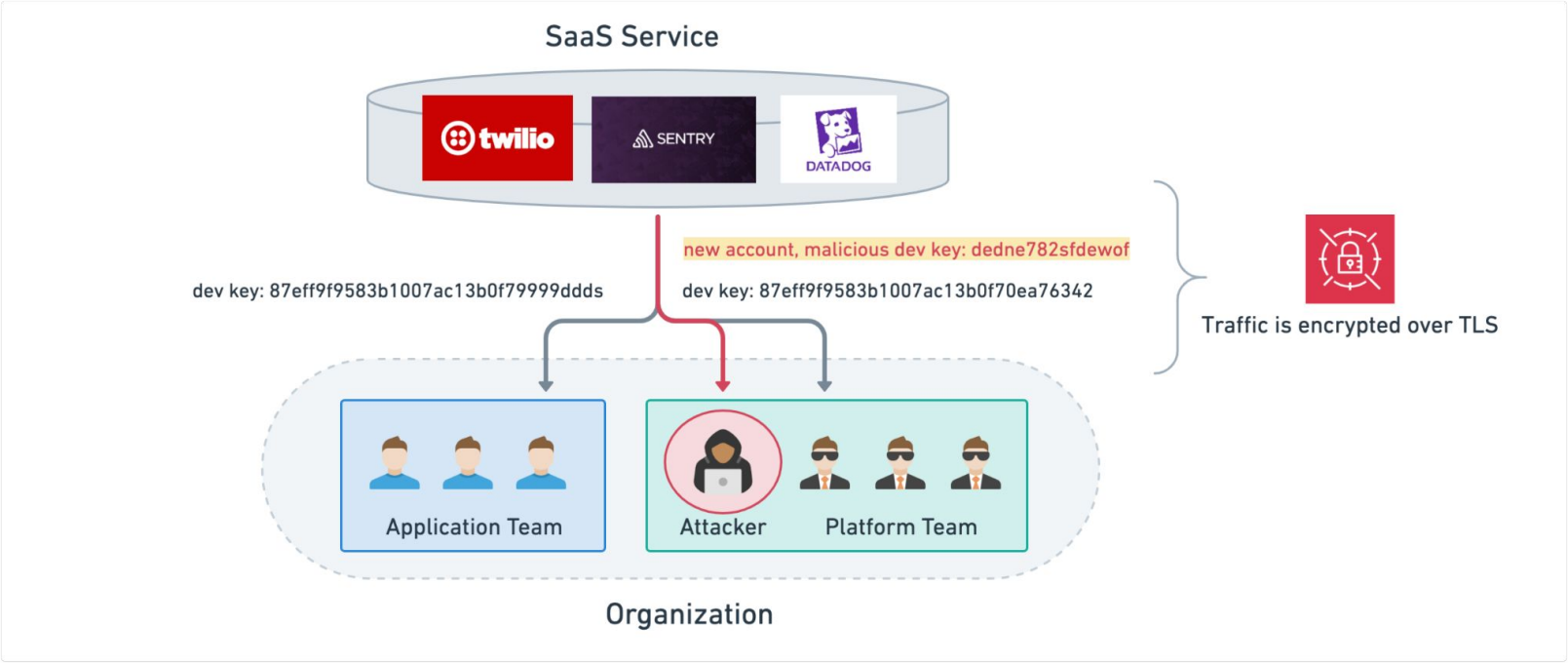


- Verdict:
  - BPF socket hook operates after TCP stack and before copy to user space.
  - BPF program “sees” data in order and retransmits are consumed by TCP stack.
  - Helpers to drop, redirect
  - Cork data can be done with parser program.

ISOVALENT

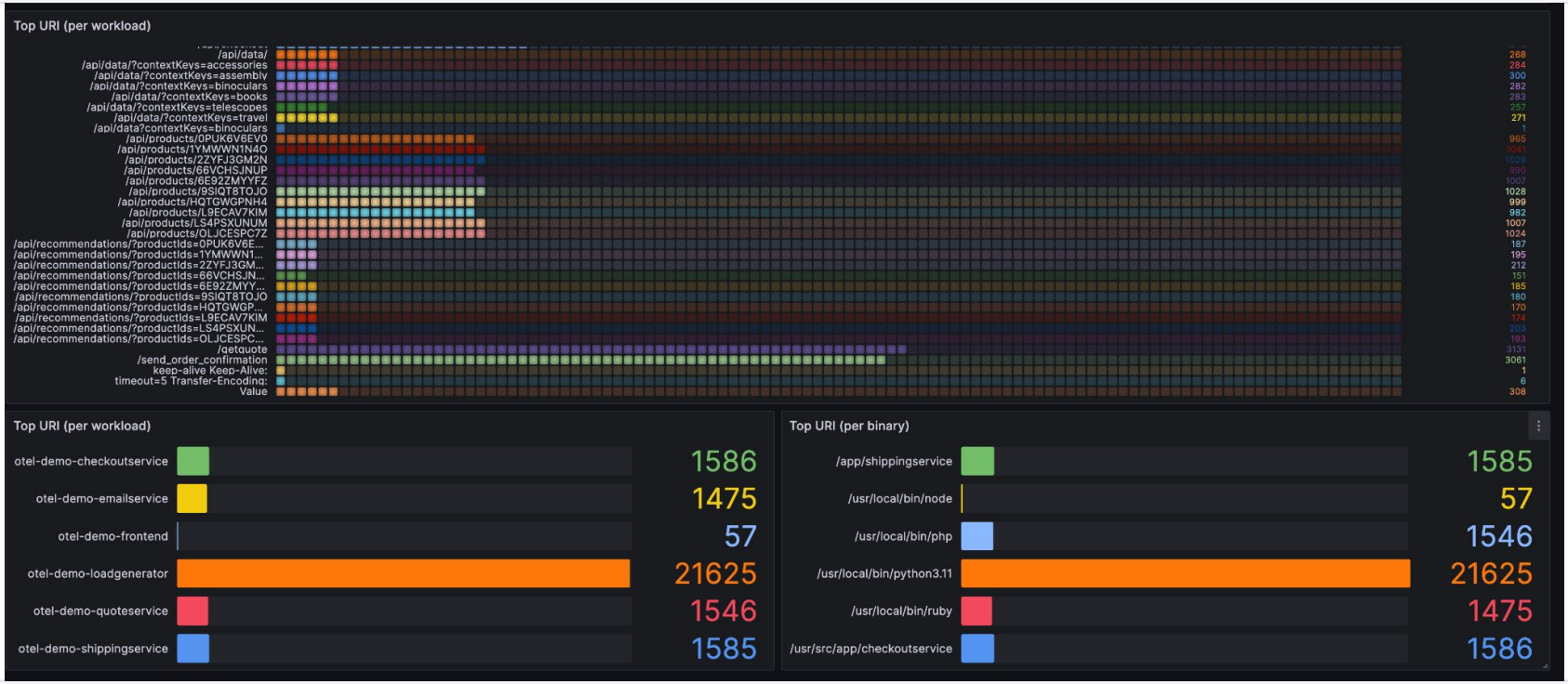
L7 HTTPS Observability Use Cases

Kubernetes Data Exfiltration via token



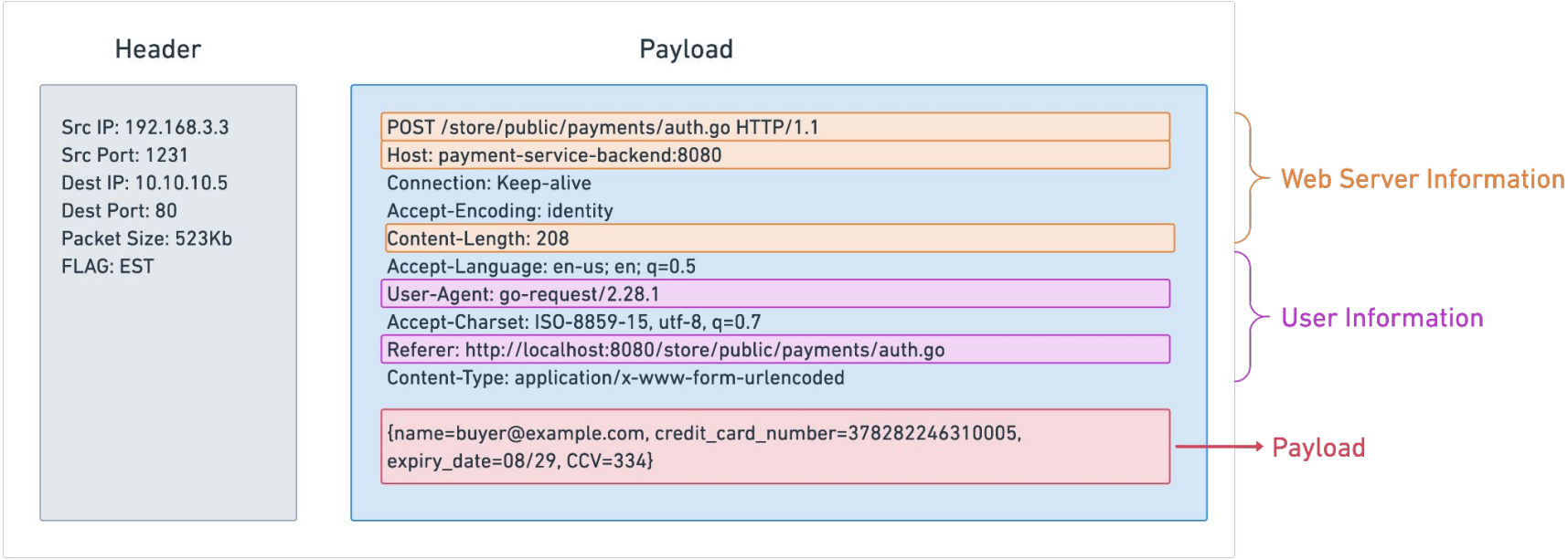
ISOVALENT

Kubernetes Data Exfiltration via token



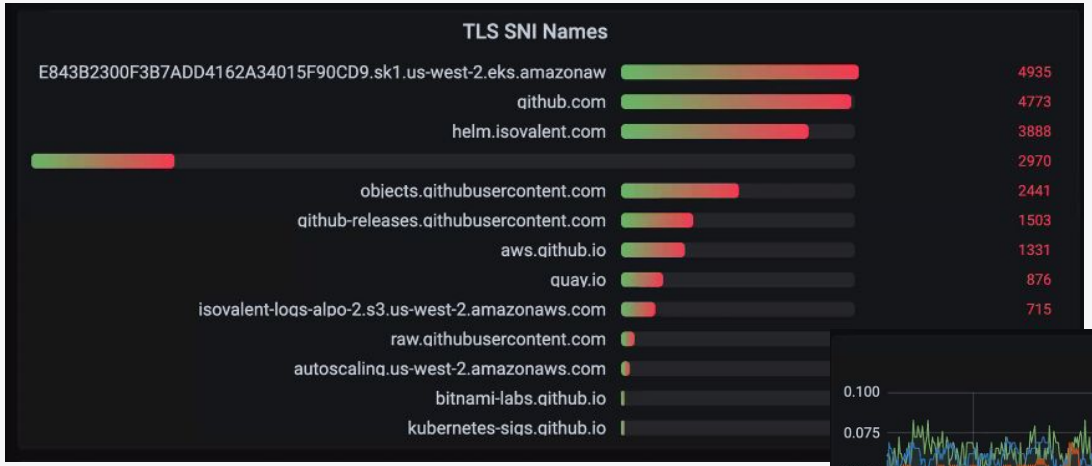
ISOVALENT

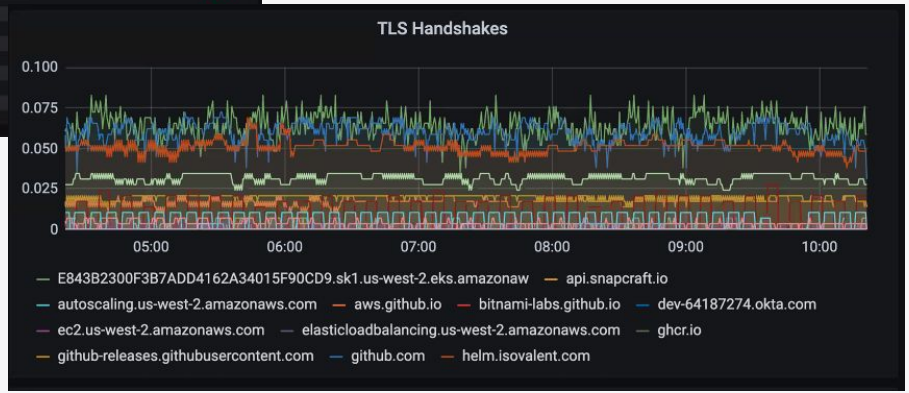
Detecting Sensitive Data Patterns from HTTP body



ISOVALENT

TLS/SSL Visibility

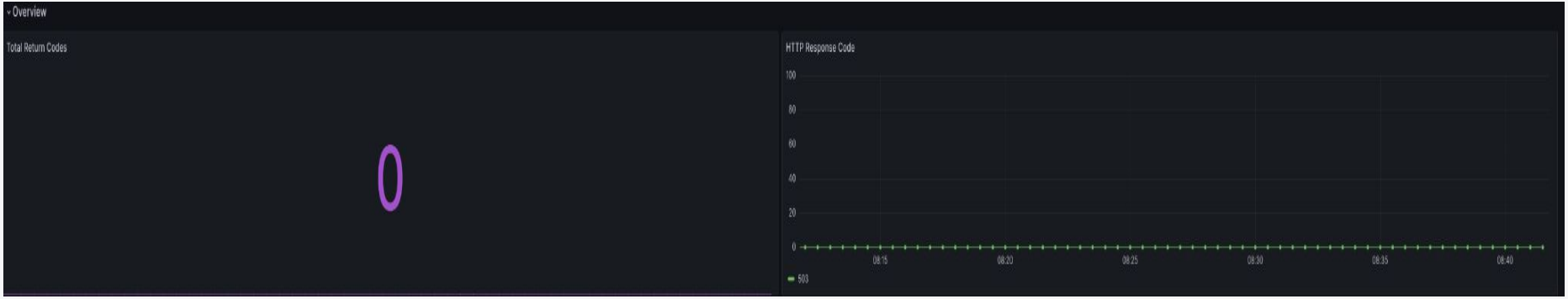




ISOVALENT

L7 Observability
How TLS breaks Observability?

L7 Observability: Encryption Enabled



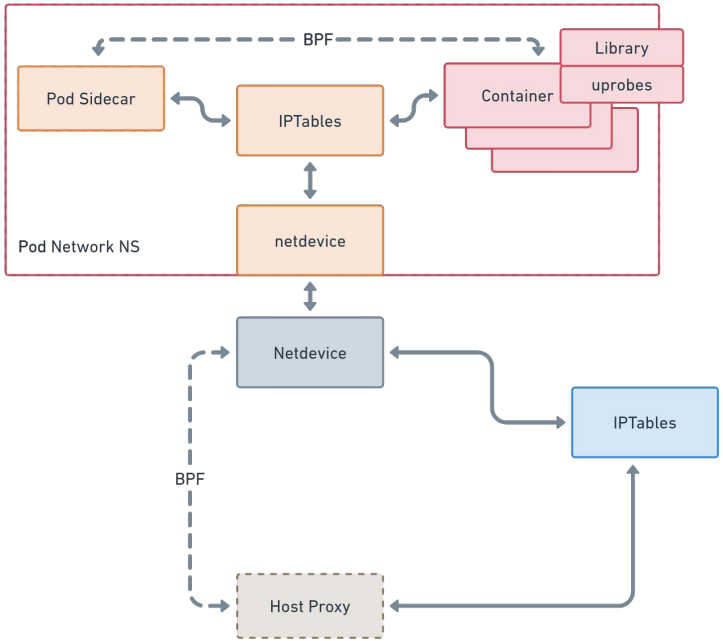
L7 Observability With Encryption

1. Don’t encrypt [not a viable Security model]
2. Uprobes [not a viable Security model]
3. TLS termination [certificate orchestration & consolidation]
4. kTLS + BPF [requires openssl3, currently in Beta]

L7 Observability: TLS Termination

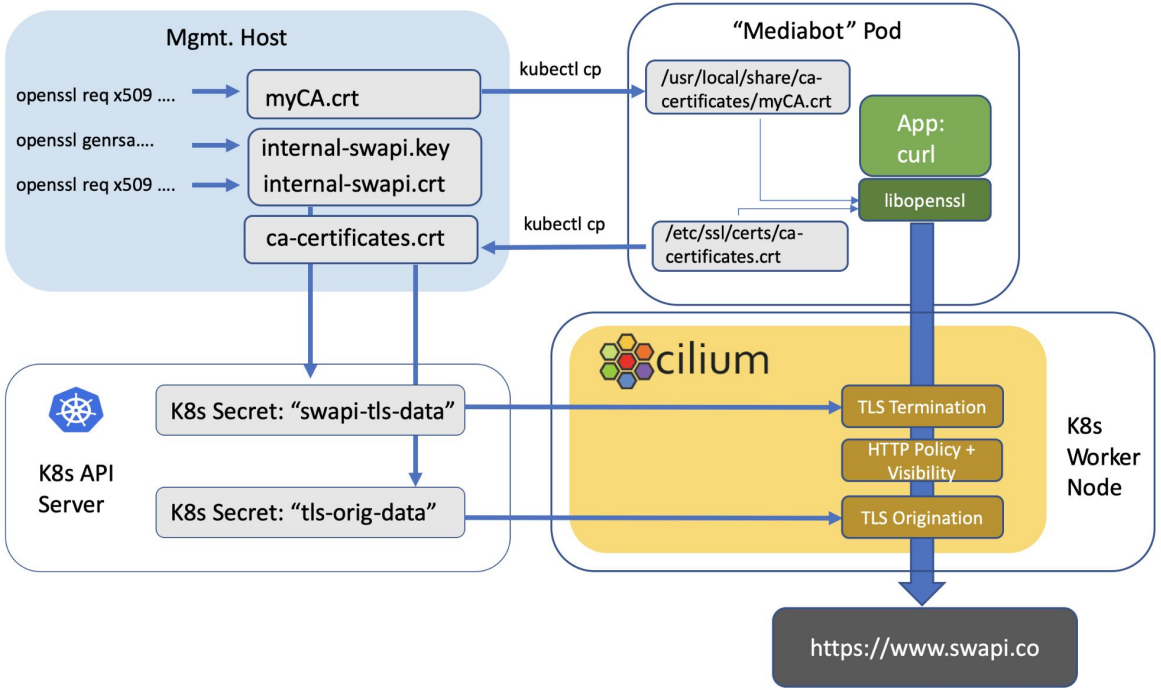
curl https://ebpf.io:

1. **curl: openssl encrypt HTTP Request**
   a. **Initiate TLS normally**
2. curl: sendmsg HTTPS Request
3. Pod Networking Sends PKT(s)
4. Host Networking Redirects to Host Proxy
5. **Host Proxy observes HTTPS Request**
   a. **TLS Terminate**
   b. **Observe**
   c. **TLS Origination**



ISOVALENT

L7 Observability: TLS Termination



ISOVALENT

L7 Observability: TLS Termination

curl http://ebpf.io:

1. **curl: openssl encrypt HTTP Request**
   a. **Initiate TLS normally**
2. curl: sendmsg HTTPS Request
3. Pod Networking Sends PKT(s)
4. Host Networking Redirects to Host Proxy
5. **Host Proxy observes HTTPS Request**
   a. **TLS Terminate**
   b. **Observe**
   c. **TLS Origination**

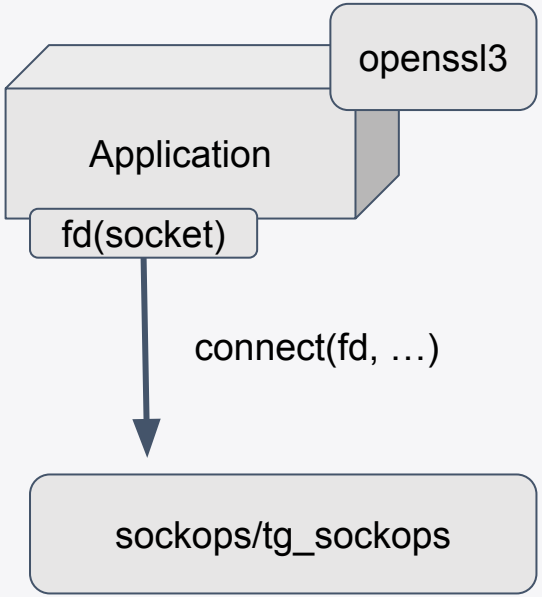
Drawbacks

- Multiple encrypts/decrypt
- CNI routing integration
- Certificate injection
- DNS parser/integration

ISOVALENT

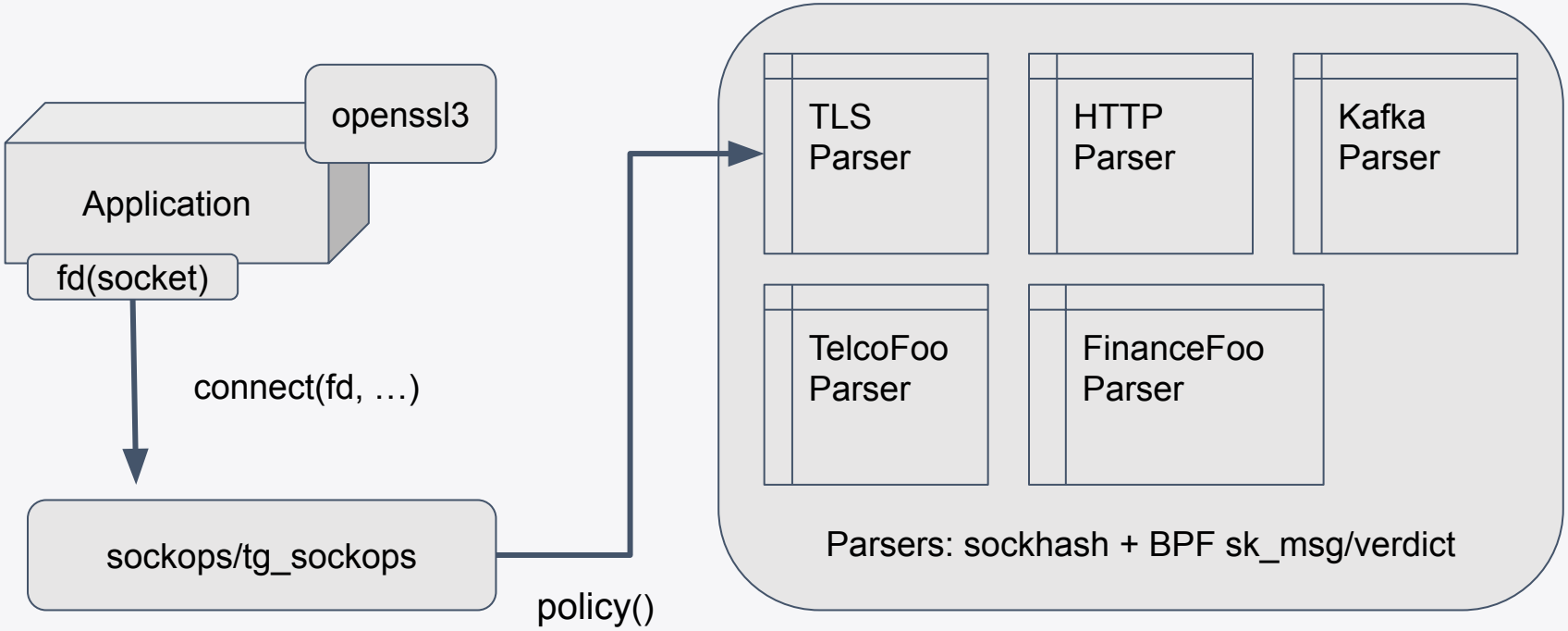
L7 Observability
KTLS

L7 Observability: sockops

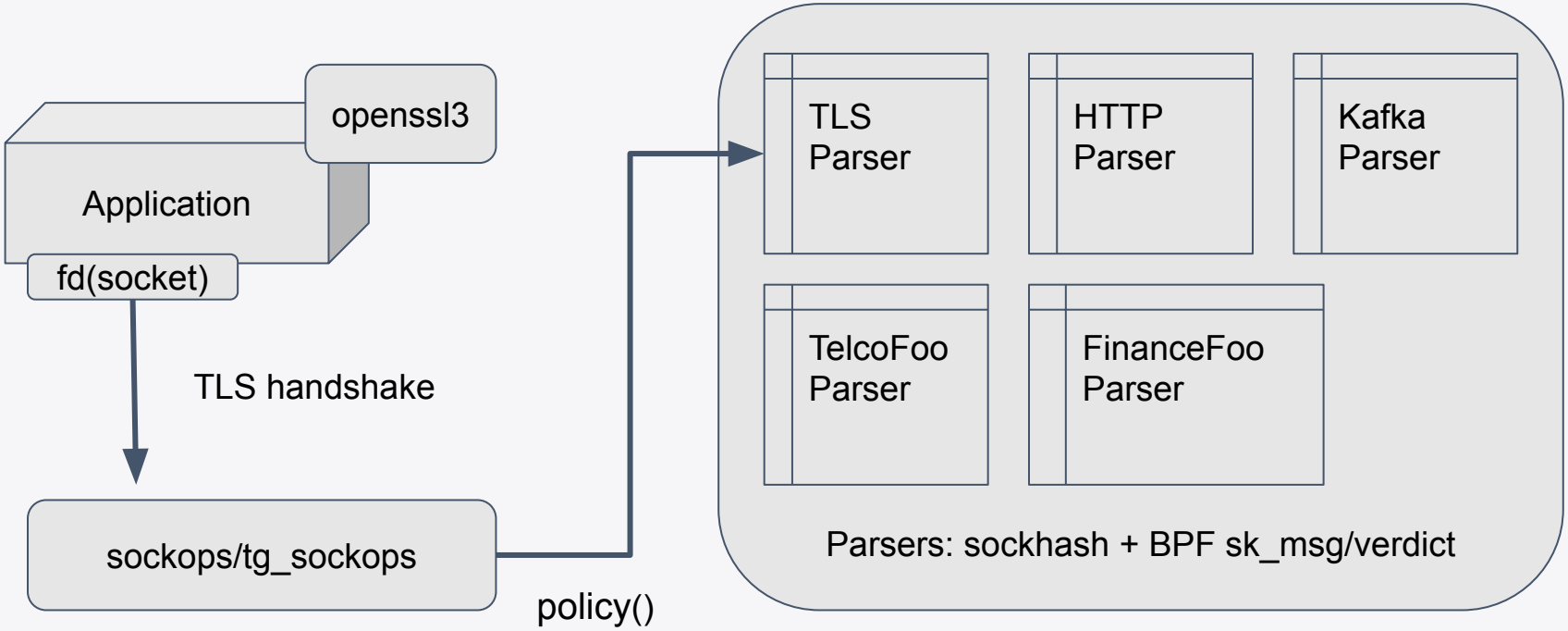



```
switch (op) {
  case BPF_SOCK_OPS_PASSIVE_ESTABLISHED_CB:
  case BPF_SOCK_OPS_ACTIVE_ESTABLISHED_CB:
        if (family == AF_INET || family == AF_INET6)
              enable_parser(skops);
        break;
  default:
        break;
  }
```

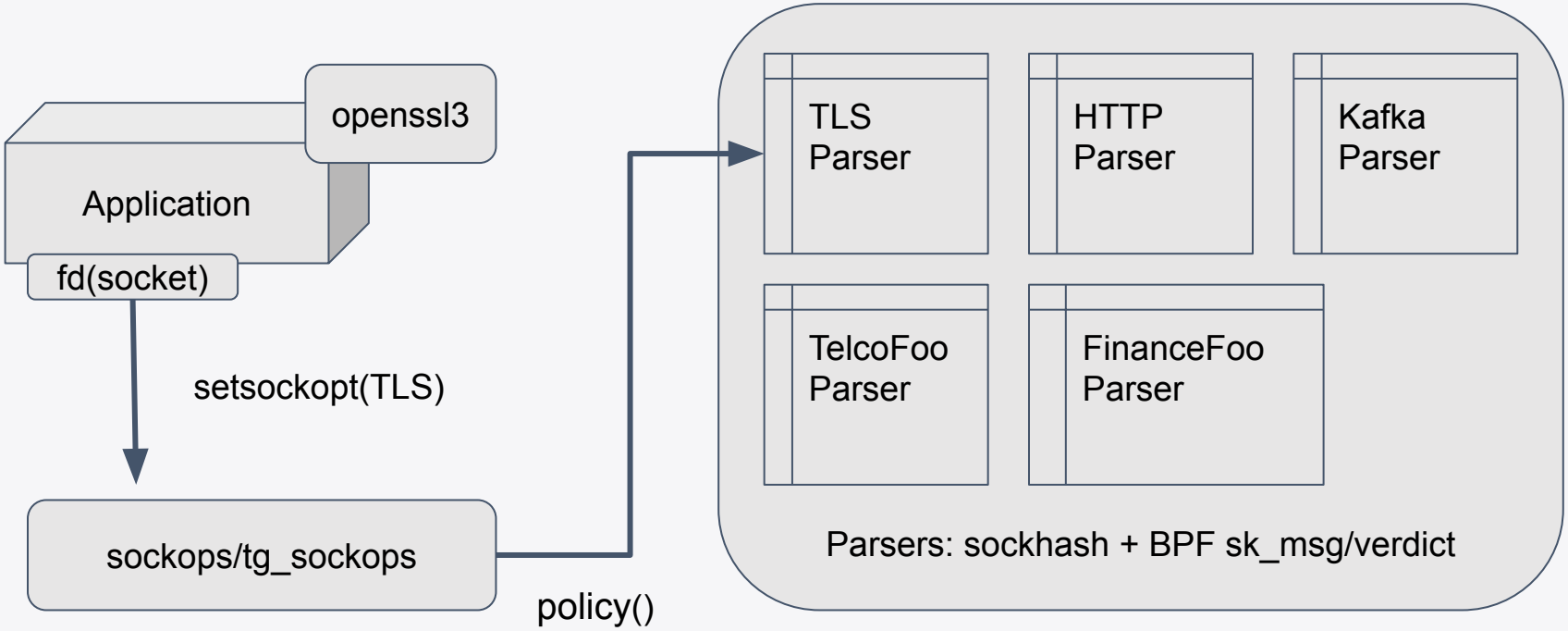
L7 Observability: sockops



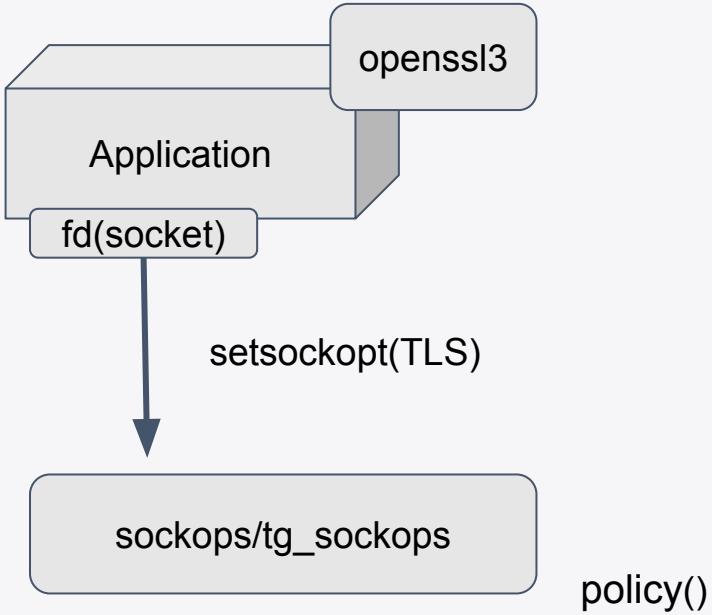
L7 Observability: sockops



L7 Observability: sockops

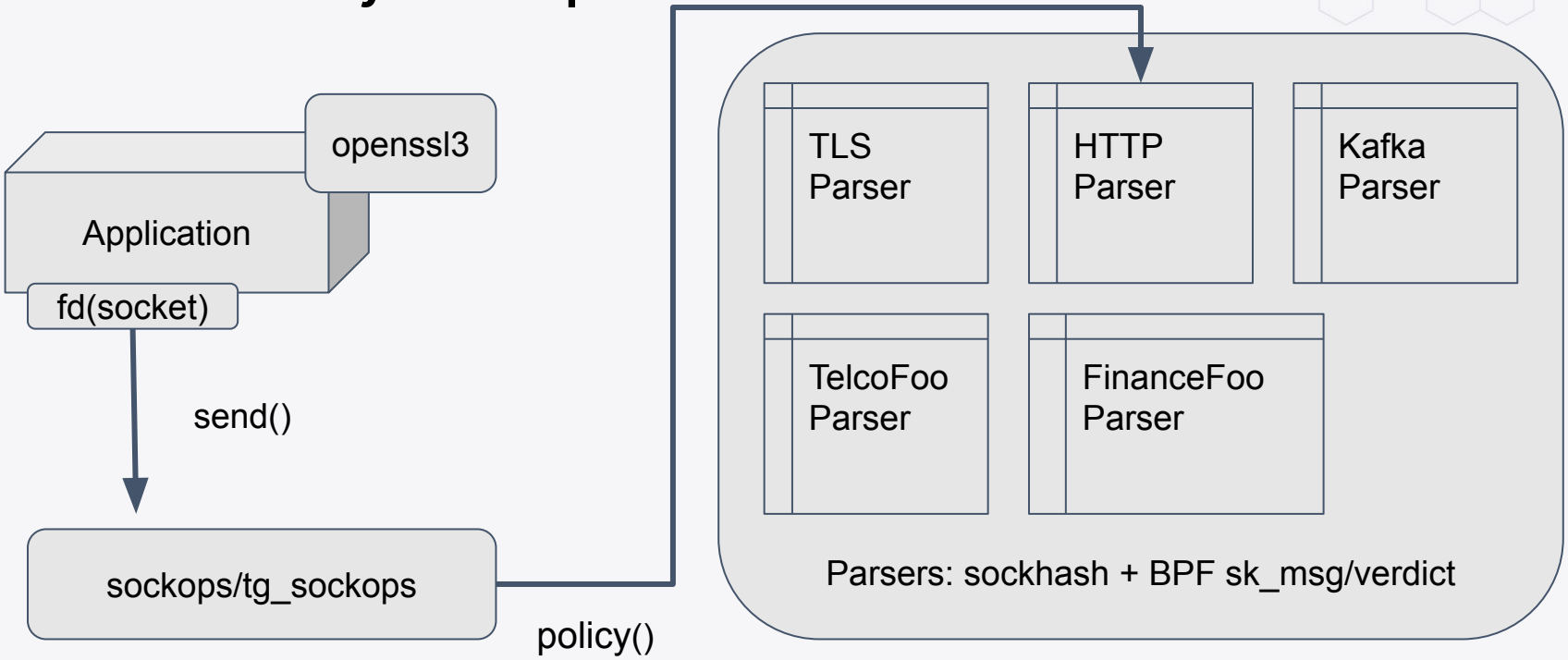


L7 Observability: sockops



Note: As a policy engine we have an interesting decision to make if the application does not use kTLS. We decide to alert if no visibility is available, but could drop traffic, sigkill application, freeze pod and so forth.

L7 Observability: sockops



ISOVALENT

L7 Observability With TLS Demo

kTLS: Demo

🚀 process /usr/local/bin/curl https://ebpf.io/tetragon
🔌 connect /usr/local/bin/curl UDP 172.17.0.2:42938 => 8.8.8.8:53
📖 dns /usr/local/bin/curl [ebpf.io.] [A]
📖 dns /usr/local/bin/curl [ebpf.io.] [AAAA]
📖 dns /usr/local/bin/curl NOERROR [ebpf.io.] [A] [104.198.14.52]
📖 dns /usr/local/bin/curl NOERROR [ebpf.io.] [] []
🧮 socket /usr/local/bin/curl UDP 172.17.0.2:42938 => 8.8.8.8:53 tx 50 B rx 131 B
🧹 close /usr/local/bin/curl UDP 172.17.0.2:42938 => 8.8.8.8:53 tx 50 B rx 131 B
🔌 connect /usr/local/bin/curl TCP 172.17.0.2:60066 => 104.198.14.52:443 [ebpf.io.]
🔐 tls /usr/local/bin/curl 104.198.14.52:443 ebpf.io TLS1.3 TLS_AES_128_GCM_SHA256
🌐 http /usr/local/bin/curl ebpf.io GET /tetragon
🧹 close /usr/local/bin/curl TCP 172.17.0.2:60066 => 104.198.14.52:443 [ebpf.io.] tx 706 B rx 226 kB
💥 exit /usr/local/bin/curl https://ebpf.io/tetragon 0

ISOVALENT

L7 Observability Limitations

Limitations and Critics

- BPF programs only have *bounded* loops!
  BPF must terminate!
  - This is a feature. Have you ever wanted your parser to run forever?
- The header limits must be so small. Not good enough for a real parser.
  - Apache: Max headers: 100 Max header size 8kB (defaults)
  - BPF: Max headers: 120 max Headers size 1k (defaults) [120kB!?!]

    Problematic error when set to 8k:

    “The sequence of 8193 jumps is too complex”

    Could use bpf_loop and others, haven’t tried yet.

ISOVALENT

Whats Next

Sockmap, SK_MSG, Verdict

- Parser, open bug incorrect wakeup logic.
- Sockmap management and sizing is annoying
  - Separate *psock* from map its not necessary
- We only test TCP at the moment
  - nginx and apache test suites (bpf-next, LTS)
  - set of tcpreplay like http request/reply (bpf-next, LTS)
  - Multiple tetragon test suites (bpf-next, LTS)
  - veristat run on LTS kernels (missing bpf-next at the moment)
- Make Testing available as much as possible
- Benchmark, likely low hanging fruit.

KTLS

- Openssl 3.0 Required
  - TBD: Golang, Java, Python, Boring TLS(?)
  - TLS 1.3 Testing needed

- Rolling out to internal cluster now
- Support in Cilium for KTLS (including DTLS)
- Benchmarking needed to confirm results (teaser for next talk)
- Move KTLS + HTTP testing on par with just HTTP

Parsers

- Regex Payload parser
  - Do we need a kfunc or open coded?
  - We don’t need a full regex basic foo*.[io,com,org]/*/[1..10]/*/bar/*
- Increase Header Size to 8kB. (bpf_loop?)
- Http2 parser, early prototype exists.
- How to handle existing sessions?

Thank you! Q&A

cilium/tetragon

@ciliumproject

cilium.io

Email:
john.fastabend@gmail.com
john@isovalent.com
Slack:
@jrfastab