Hunting for Attack Chains in Event Streams

Ray Ruvinskiy and Jonathan Walsh

April 2, 2019

ARCTIC WOLF

About the Presenters

Ray Ruvinskiy

- Lead of the Detection Automation Team. Our team's job is to help our Security Analysts avoid the noise and zero in on the actionable.

Jonathan Walsh

- Member of the Threat Operations and Analysis team. We utilize the tools provided by the Detection Automation Team to improve our threat detection.

About Arctic Wolf Networks

What is a SOC?

Security Operations Centres (SOCs) deal with security issues on an organizational level.

About Arctic Wolf Networks

SOC-as-a-Service

AWN CyberSOC manages the security detection for our customer, gathering data from deployed network sensors and logs from other security tools.

These logs are processed by our system and the output is provided to human experts to make decisions on.

Alert on actionable data

Rising above the noise

Balance between being informative and being noisy

Focus on what the client needs to deal with immediately

“

Crying wolf may have been the boy’s undoing, but the true irony was that the wolves were always lurking nearby.

Wes Fesler
American football player

Lifecycle of an attack

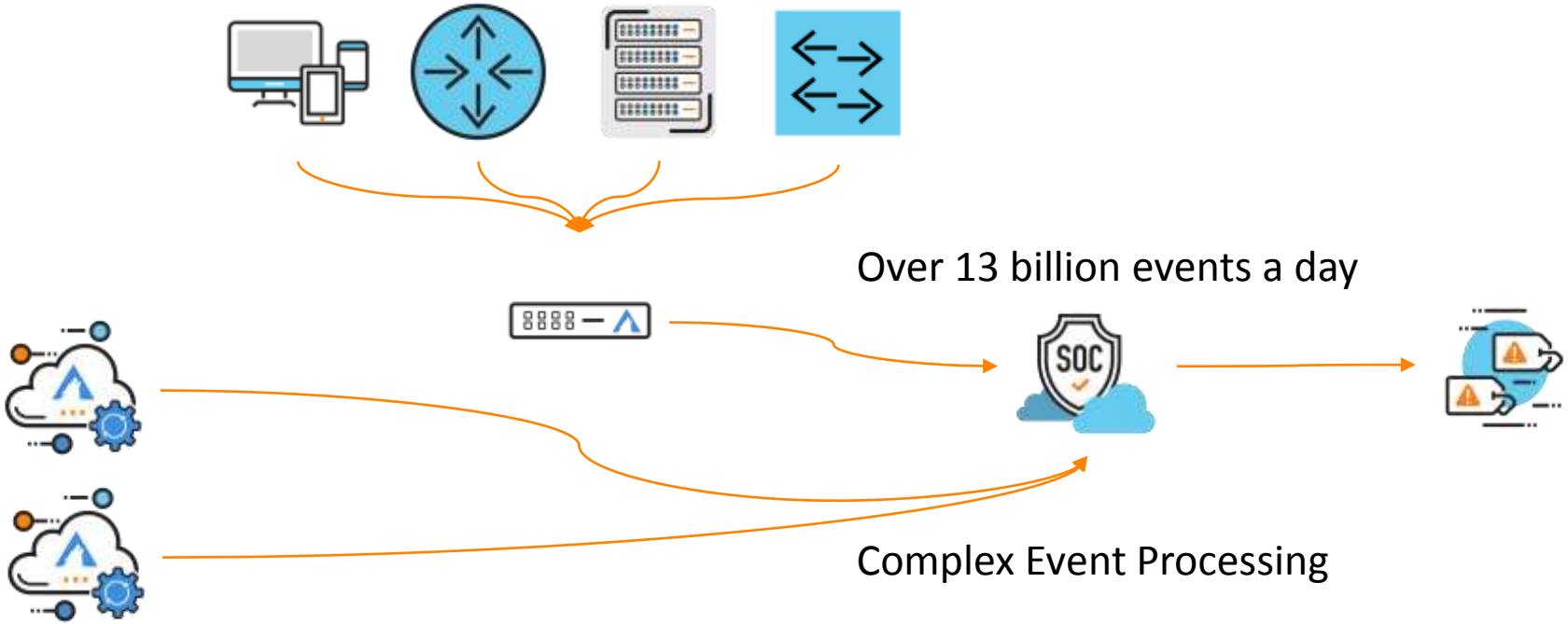
Attributes shared by network breaches

Source: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Applying Attack Chain to WannaCry Outbreaks

- Reconnaissance – social networking
- Weaponization – build ransomware binary
- Deliver/exploit – spearphishing
- Command and control – encrypted backchannel
- Actions on Objectives – encryption and lateral movement

Indicators at each of these steps can be noisy and prone to false positives. Correlation of events focuses us on what is relevant.



Why Flink?

Built for streaming

Out-of-order events

Scalability

FlinkCEP

Our experience dates to 1.2.1

Difficulties with expressivity and scalability

- At the time, no counts

Esper

Mature and expressive streaming CEP framework

EPL – SQL-like Event Processing Language

EPL

PsExec services created on at least 10 devices in 5 minutes

```
select … from PsExecEvent#time(5 minutes)
                #unique(target_ip)
                #length_batch(10);
```

Typed event, with properties

Stream composition

Property in `PsExecEvent`

EPL

Endpoint virus detection followed by network IDS exploit detection

```
select … from pattern [
  AntiVirusEvent -> IDSExploitEvent where timer:within(15 minutes)
];
```

”followed by” operator

EPL

A minimum number of login failures with no successes interspersed, then a success

```
create window LoginFailureWindow#time(5 minutes) as LoginEvent;

on LoginEvent(not login_success) as login_failure
    merge LoginFailureWindow as failures
    where login_failure.id = failures.id
    when not matched then insert select *;

on LoginEvent(login_success) as login_success
    select and delete … from LoginFailureWindow as failures
        having count(*) >= 10;
```

Complex Event Processing with Flink and Esper

Putting it all together

A single, continuously-running Flink job

Complex Event Processing with Flink and Esper

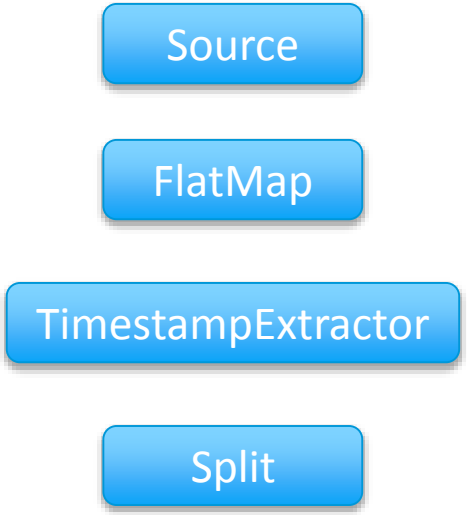
Putting it all together

Events are ingested

Events are parsed and subjected to sanity filtering

Timestamp extraction, watermark assignment

`SplitStream` used to create a stream per rule



Complex Event Processing with Flink and Esper

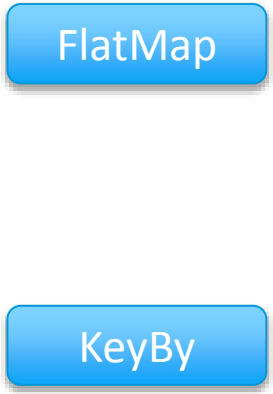
Putting it all together

Each of the per-rule streams aggressively filters events of interest and converts to `Tuples`

- Reduce per-event memory footprint as much as possible

`keyBy` rule-specific partition key to ensure unrelated events are processed in isolation

- e.g., for a login failure rule we key by customer ID and username

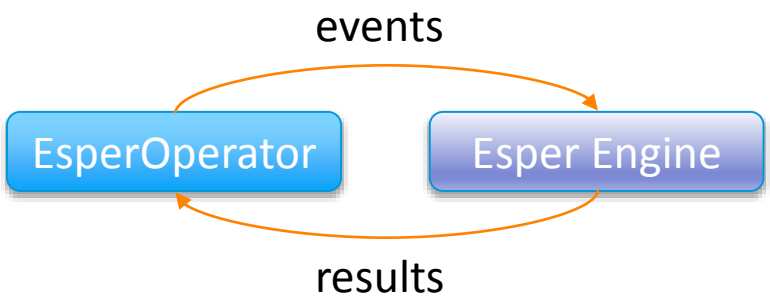


Complex Event Processing with Flink and Esper

Putting it all together

Esper operator

- Per-rule (potentially multi-statement) EPL that defines the logic
- Priority queue to order events by timestamp
  - Same as FlinkCEP
- Send tuples to Esper engine
- Register subscriber for results



Complex Event Processing with Flink and Esper

Putting it all together

Esper operator output formatted as AWN events with a special type

These AWN events then undergo another filter pass where some are whitelisted depending on specific rules defined by Security Analysts

Events that survive this pass then brought to Security Analyst attention for investigation

Flink/Esper at AWN

AWS EMR (Elastic MapReduce)

- Managed Hadoop environment
- Easiest way to get started two years ago (Flink 1.2)

Flink 1.4.2

m5.2xlarge core nodes

- 8 virtual cores, 32 GB RAM
- 8 task slots per node

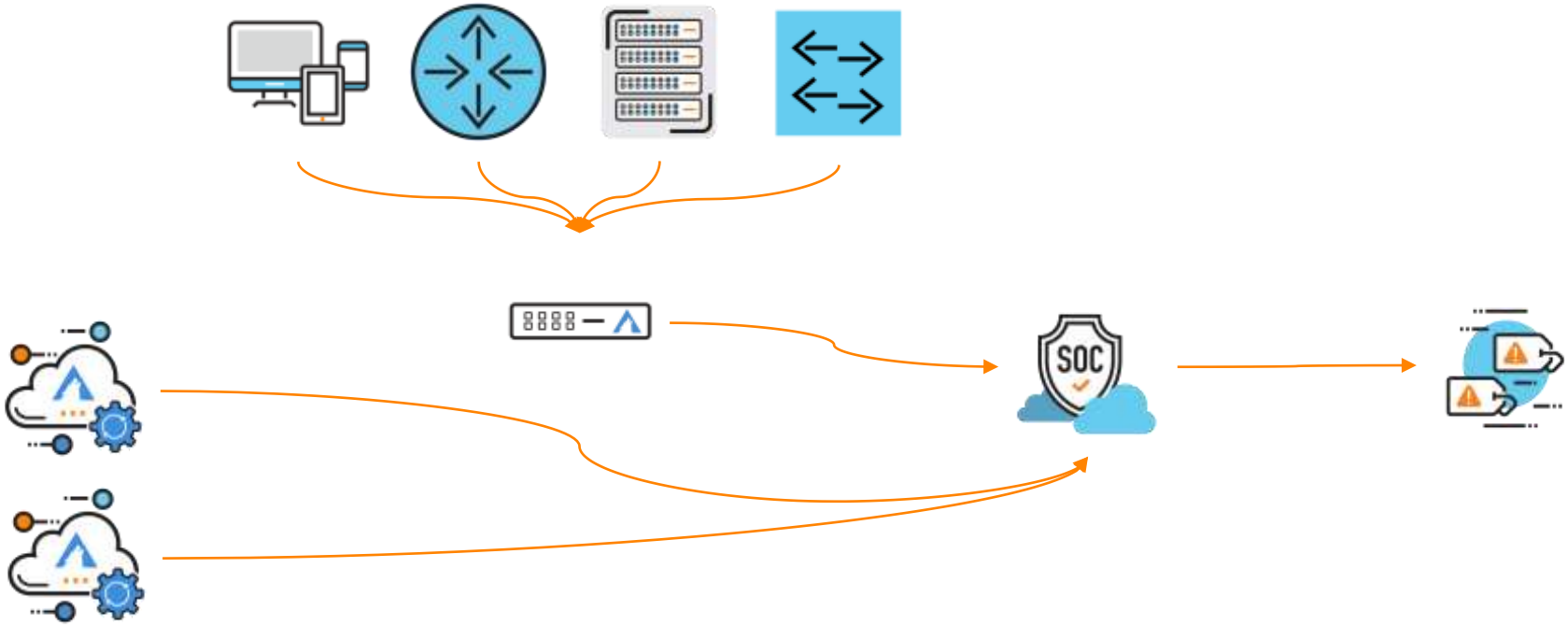
Flink/Esper at AWN

Over 30 rules

Over 13 billion events a day passing through Flink

- But only a small subset is evaluated by the rules

6 m5.2xlarge instances



Flow of Time Difficulties

Different data sources have different delays

- Events from cloud data sources (e.g., Office365) can in particular trickle in slowly

Time zone adjustments

- Normalize to UTC but *a lot* of data sources to keep track of

In summary

Together, Flink's simple deployment patterns and scalability and Esper's powerful CEP engine enable us to pick out attack chains in streams of billions of events.

Thank You

For more information, visit us at: www.arcticwolf.com

Follow us on: