ADDO

ALL DAY DEVOPS

NOVEMBER 6, 2019

John Melton

Building Self-Defending Applications With OWASP AppSensor

Introduction

thesis:
modern secure applications protect themselves against attackers

The Idea

The Tool

a (brief) history

Not too long ago dev

- mostly web apps [RoR, PHP, .NET, Java)
- ajax (jquery) use growing
- mobile just getting started
- deployment to VMs
- BI tools
- AWS starting
- cloud hype cycle (NIST)

~now dev

- service mesh
- functional / rx programming
- cloud everything
- nosql / CAP light
- containers / orchestration
- big data
- stream processing
- microservices
- config management
- ci/cd
- beacons [usage, ads, errors, performance]
- actors/csp
- cqrs / event sourcing
- mobile
- JS Everywhere

1 .. * of [scale, speed, cloud, lack of environmental access]

ADDO
ALL DAY DEVOPS









last ~5 years security

- 3rd party libs (dep-check)
- bug bounties
- sast / dast evolve (ZAP)
- iast / rasp
- http security headers
- automatic encoding (JXT)
- *-monkey –NetflixOSS
- bdd-security/gauntlt
- ci/cd plugins
- 2fa
- osquery

1 .. * of [scale, speed, cloud, lack of environmental access]

dev vs. security

- dev is exploiting fundamental architectural and deployment changes to add business value

- security is iterating on existing solutions - and - trying to close gaps (known problems)

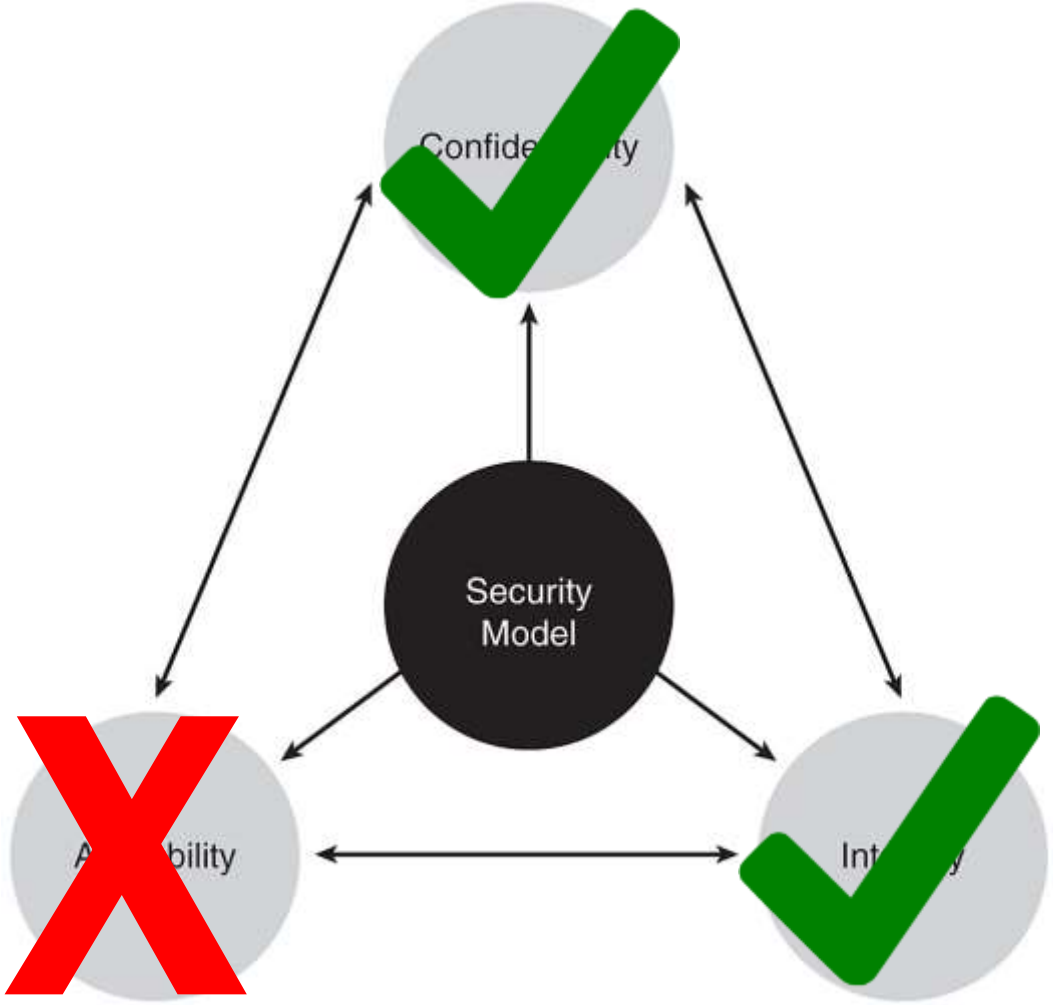
security is sharpening hand tools

while dev has moved to power tools

motivations

traditional “security”

- confidentiality and integrity important
- availability often **ignored** by security (informs the whole industry- eg. tooling)
- if availability important, runtime important



Service Temporarily Unavailable

The server closed the connection without sending any data.

The server is temporarily unable to service your request due to maintenance downtime or capacity problems.

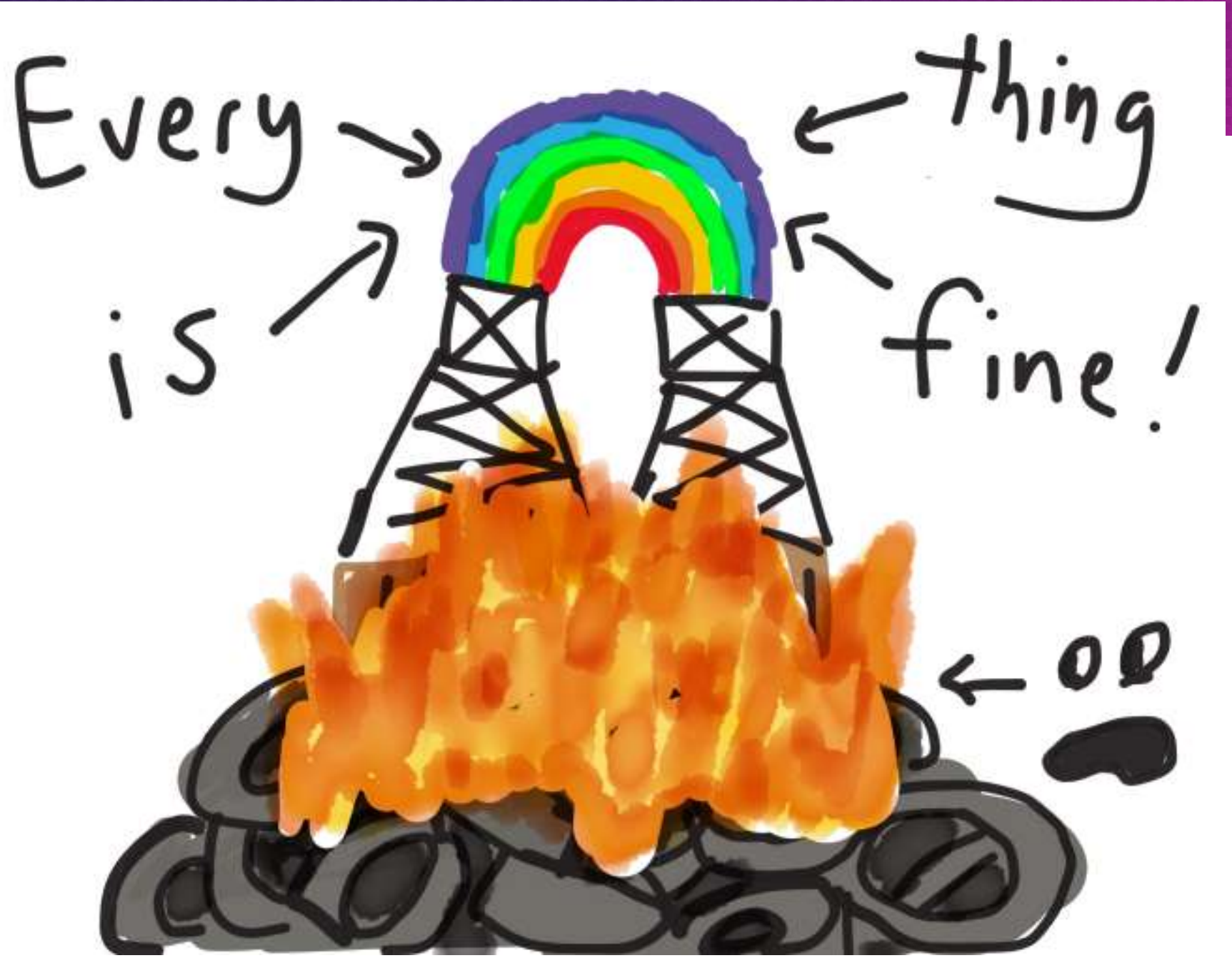
Due to heavy load on the server, connections may be temporarily blocked from locations that fetch an unusually high number of pages.

We apologize for the inconvenience.

Yep, that's secure!

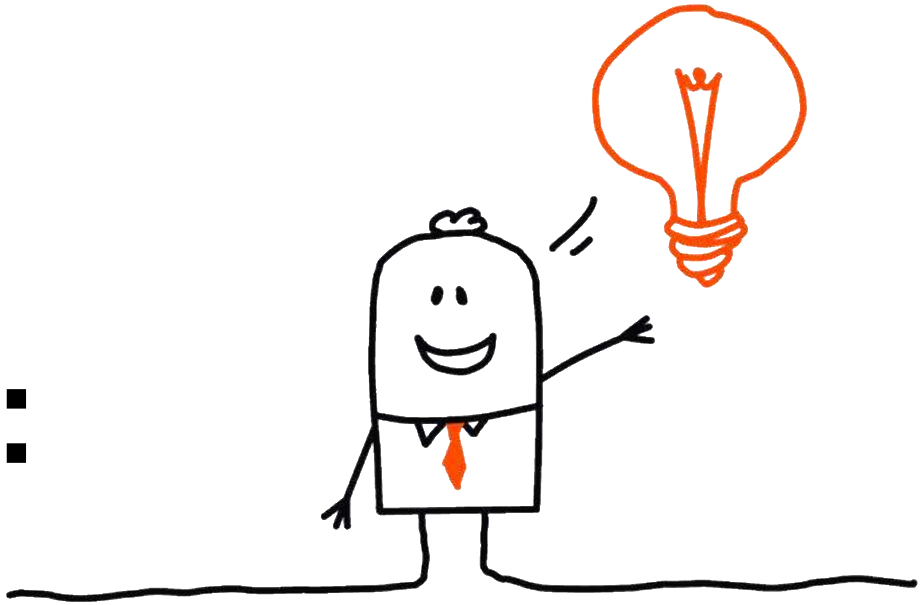
your environment

- how many concurrent users do you have right now?
- what are your users doing in the app?



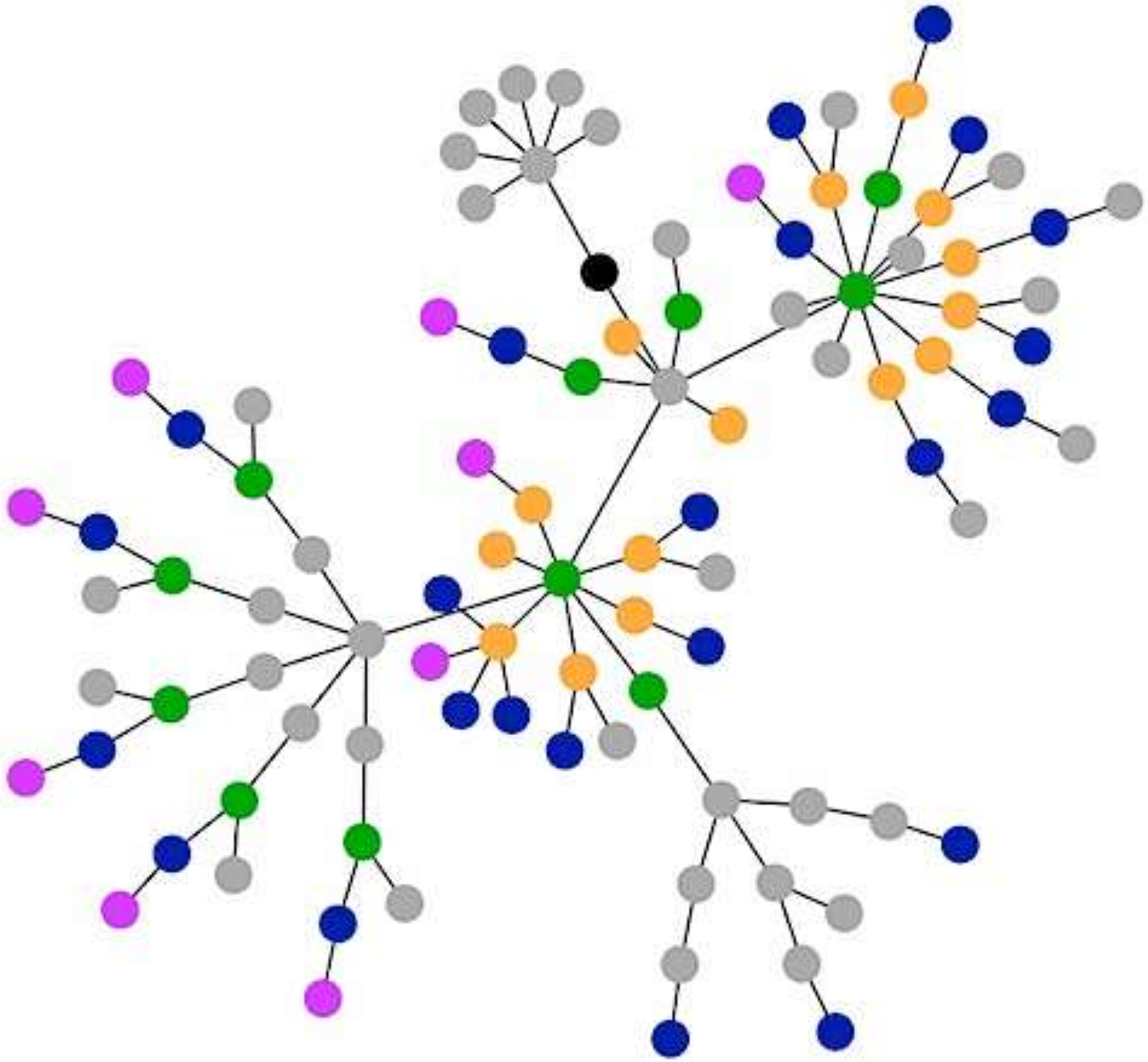
https://github.com/aphyr/jepsen-talks/blob/master/2015/goto/goto.pdf

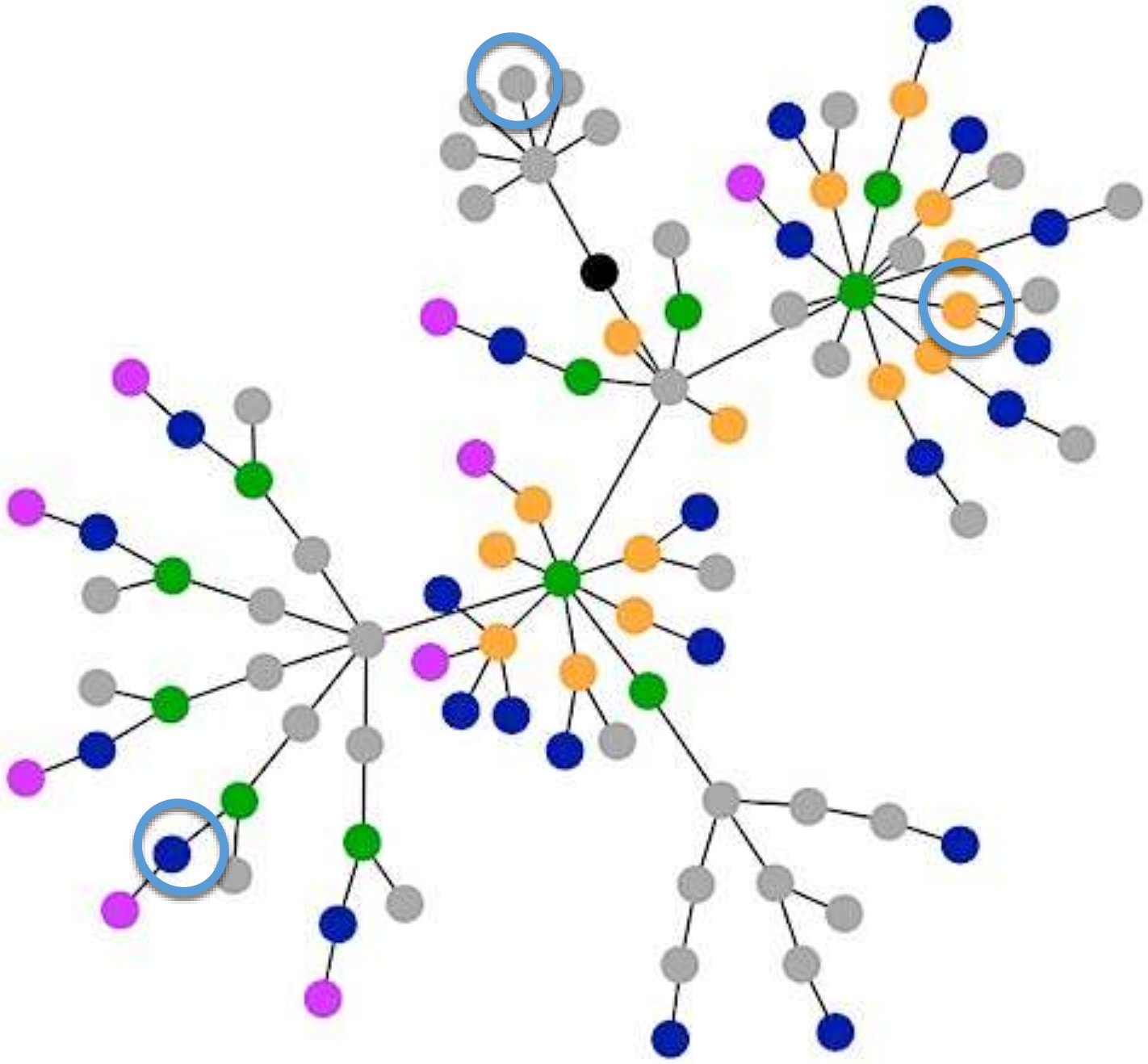
Intuition:



"traditional" security, dev, ops doesn't know what's going on in the app at runtime (holistically)

Security defects are a subset of all defects

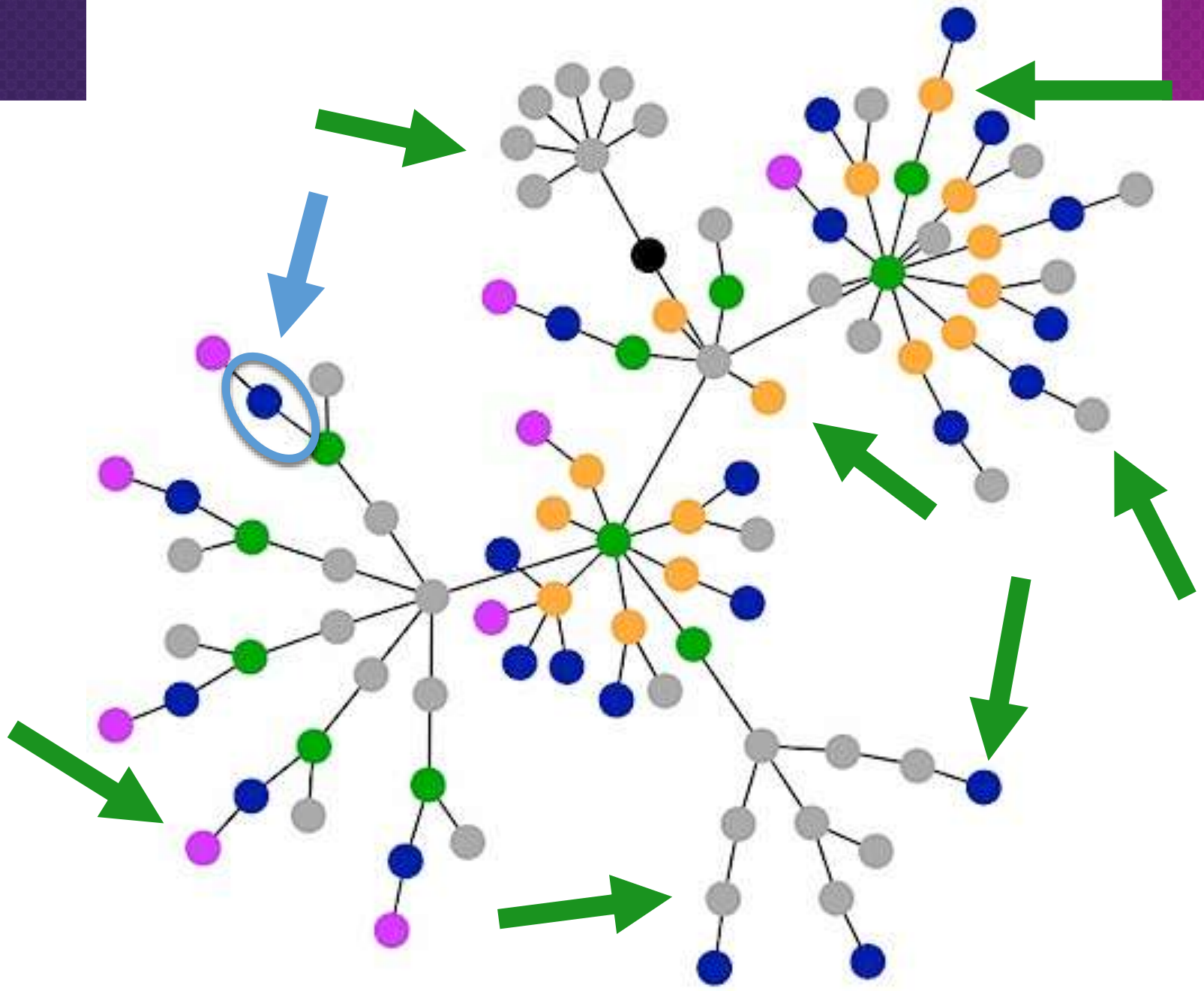




catching defects

- what do dev/qa do for functionality?
  - test [unit, integration, system, manual, tools]
- what do attackers do for security?
  - test [automated tools, manual]


1
2
3

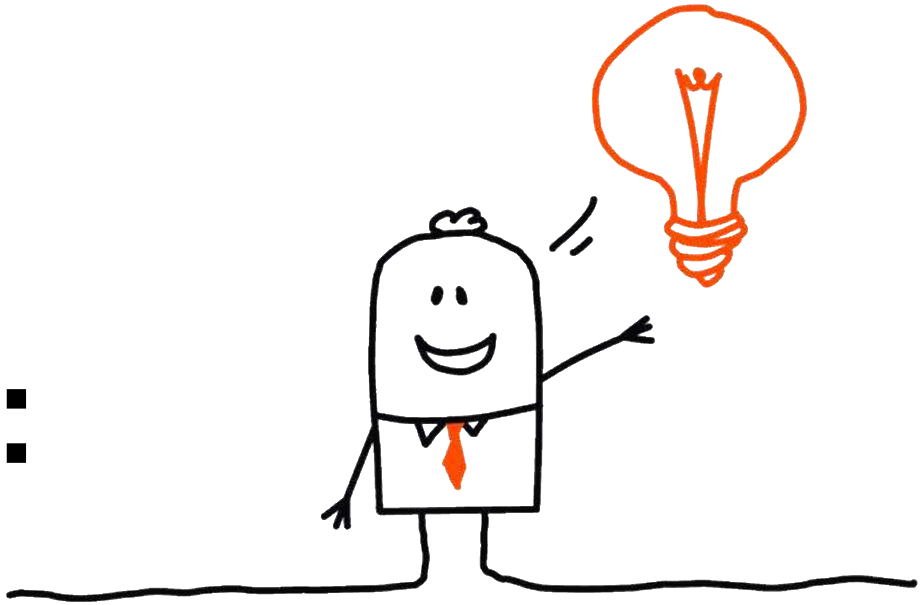


observations

- attackers do bad things
- bad things often easily recognizable (to you … in your business … if you're looking)
- attacker success often* requires > 1 attempt

*** If not, you lose**

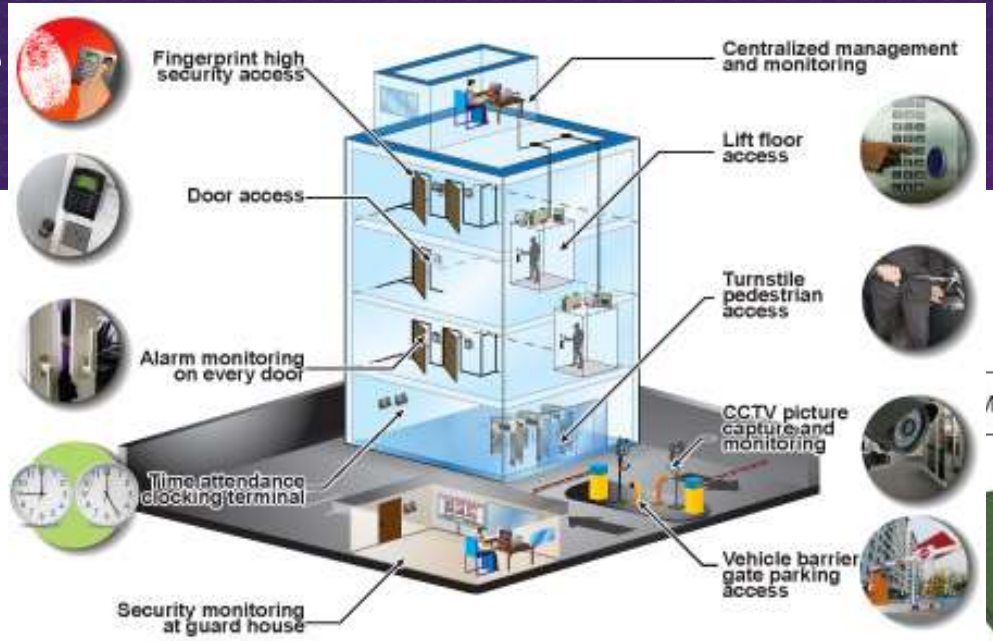
Intuition:

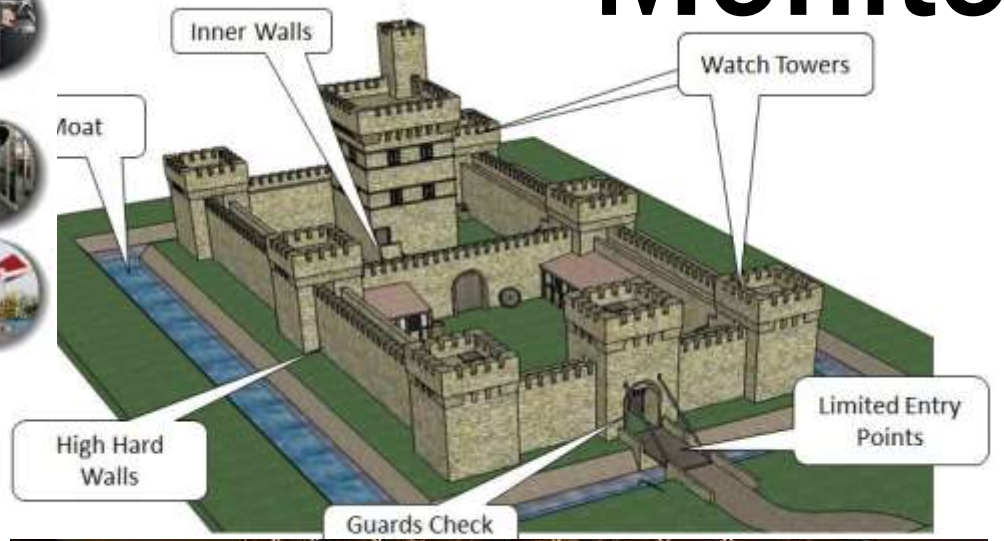


security defects exist

attackers don’t magically know what’s vulnerable

Monitoring

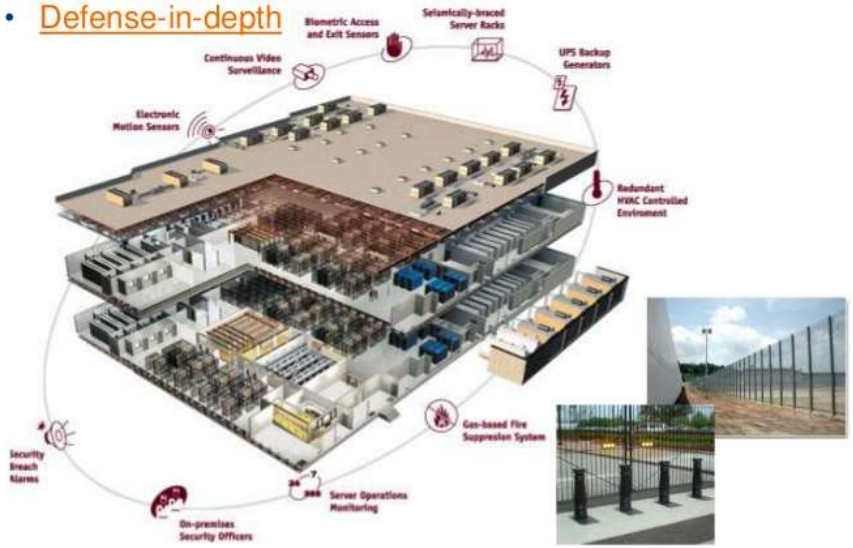




Security Countermeasures & Technologies

Strategic Approach to Physical Security ...(2/2)

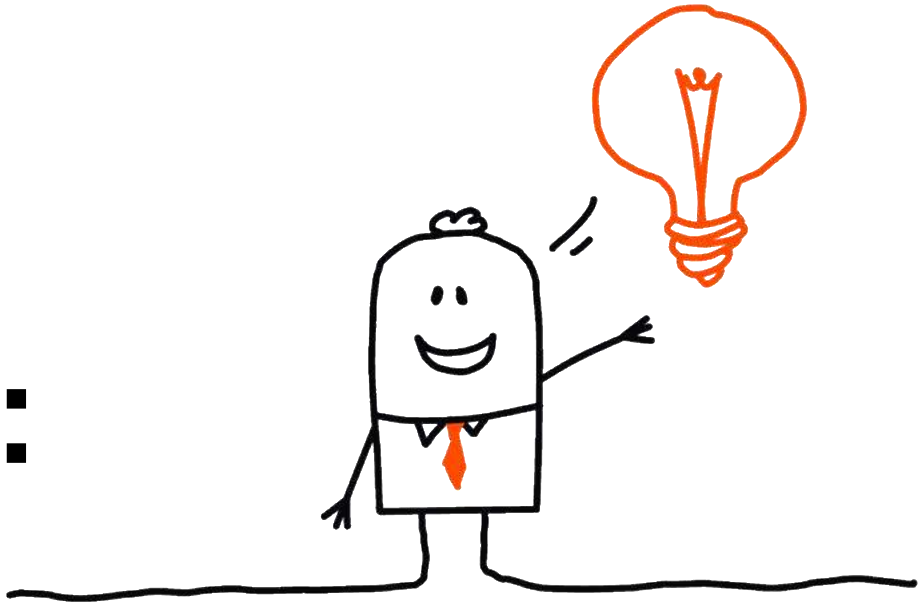
- Defense-in-depth



Source: Global Crossing website

http://worth1000.s3.amazonaws.com/submissions/414000/414200_9830_1024x2000.jpg

Intuition:



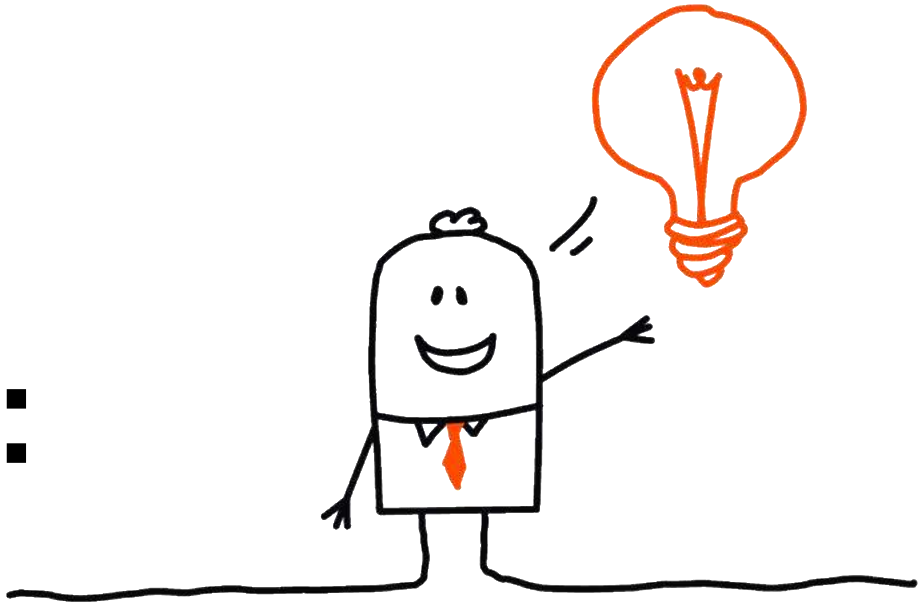
existing (security) “monitoring” is usually terrible

“security”

- 18.2 million devs
- 200K security (all, not appsec only)
- ~ 1.1 sec : 100 dev

- 1.75 sec : 100 dev (bsimm)

Intuition:



there will never be enough
“security” people

security tooling

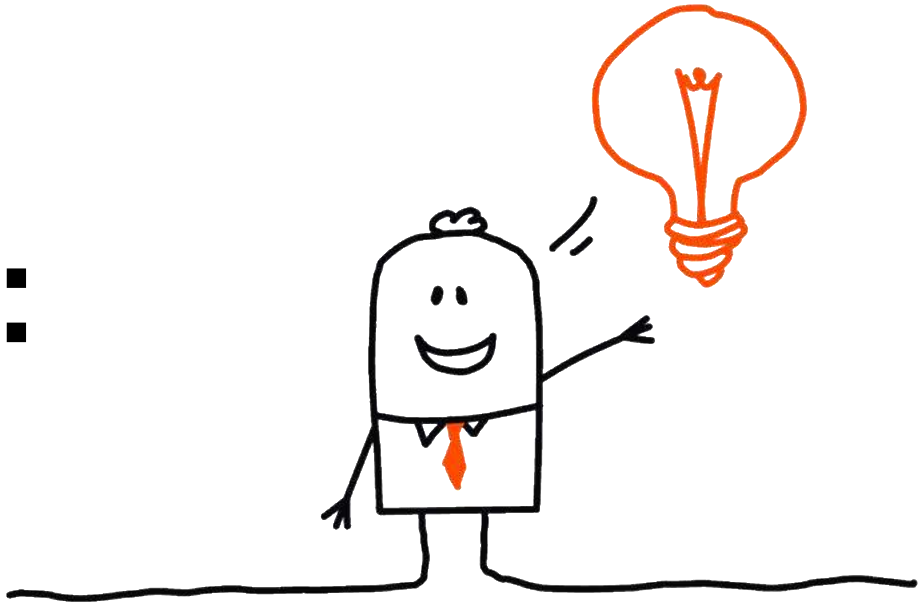
- a single mature, static language
- monolith
- http (really html) endpoints

modern dev

- polyglot static and dynamic languages
- microservices / soa
- json, thrift, protobuf, grpc, etc. endpoints
- WebAssembly ???

@petecheslock

Intuition:



“traditional” security tooling doesn’t fit modern dev

… and is unlikely to be able to keep up

defender's dillemma

- attacker needs ONE successful attack
- defender * must defend ALL attacks

*** you are defenders**

in summary (so far) ...

- "traditional" security, dev, ops doesn't know what's going on in the app at runtime (holistically)
- security defects exist
- attackers don't magically know what's vulnerable
- existing (security) "monitoring" is usually terrible
- there will never be enough "security" people
- "traditional" security tooling doesn't fit modern dev

... actual defense is _really_ hard

the pitch

(a humble proposal)

having to deal with [scale, speed, cloud, lack of environmental access]..

..this as of now incomplete transition..

..is a *huge* <u>opportunity</u> for improving security

the pitch (#0)

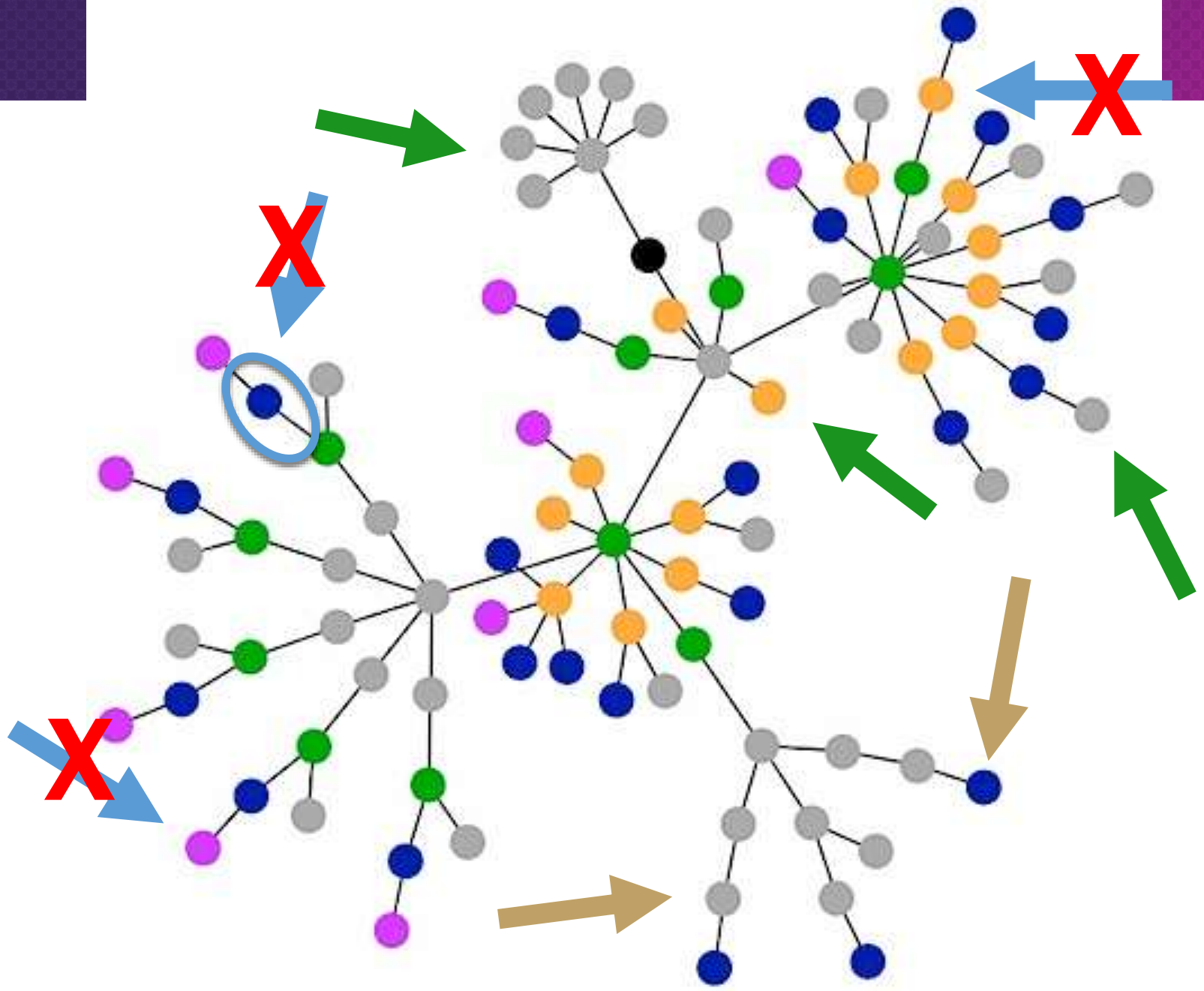
- in addition to a secure SDLC … (ie. > 1 request/attack)
- if you're not at this stage, work on it first


CRAWL
WALK
RUN
FLY

the pitch

- figure out what's happening at runtime
- make intrusion detection primitives available in app
- exploit automated response > manual response
- stop attacker before success *
- get self-protecting applications and valuable intel

*** define success**

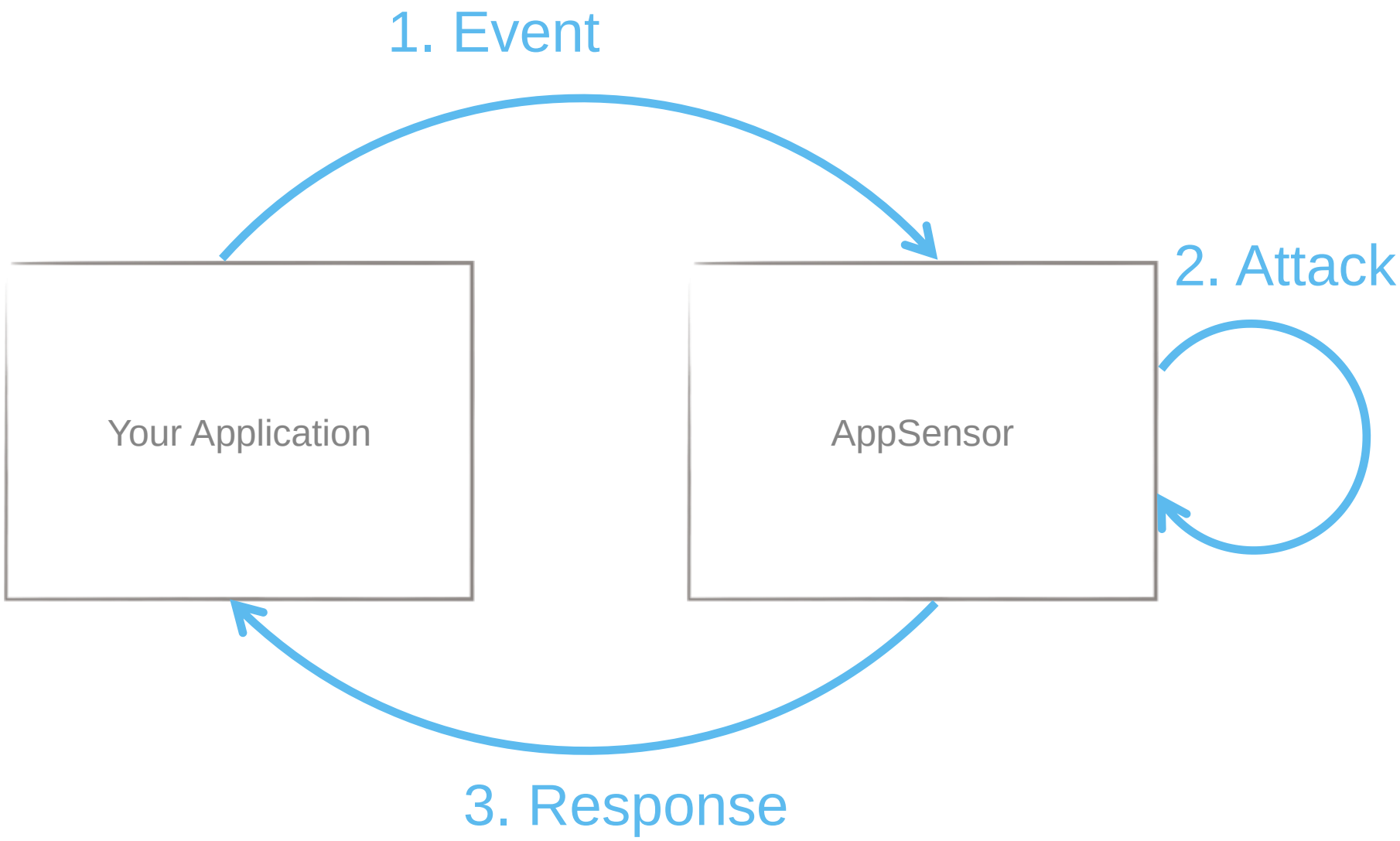


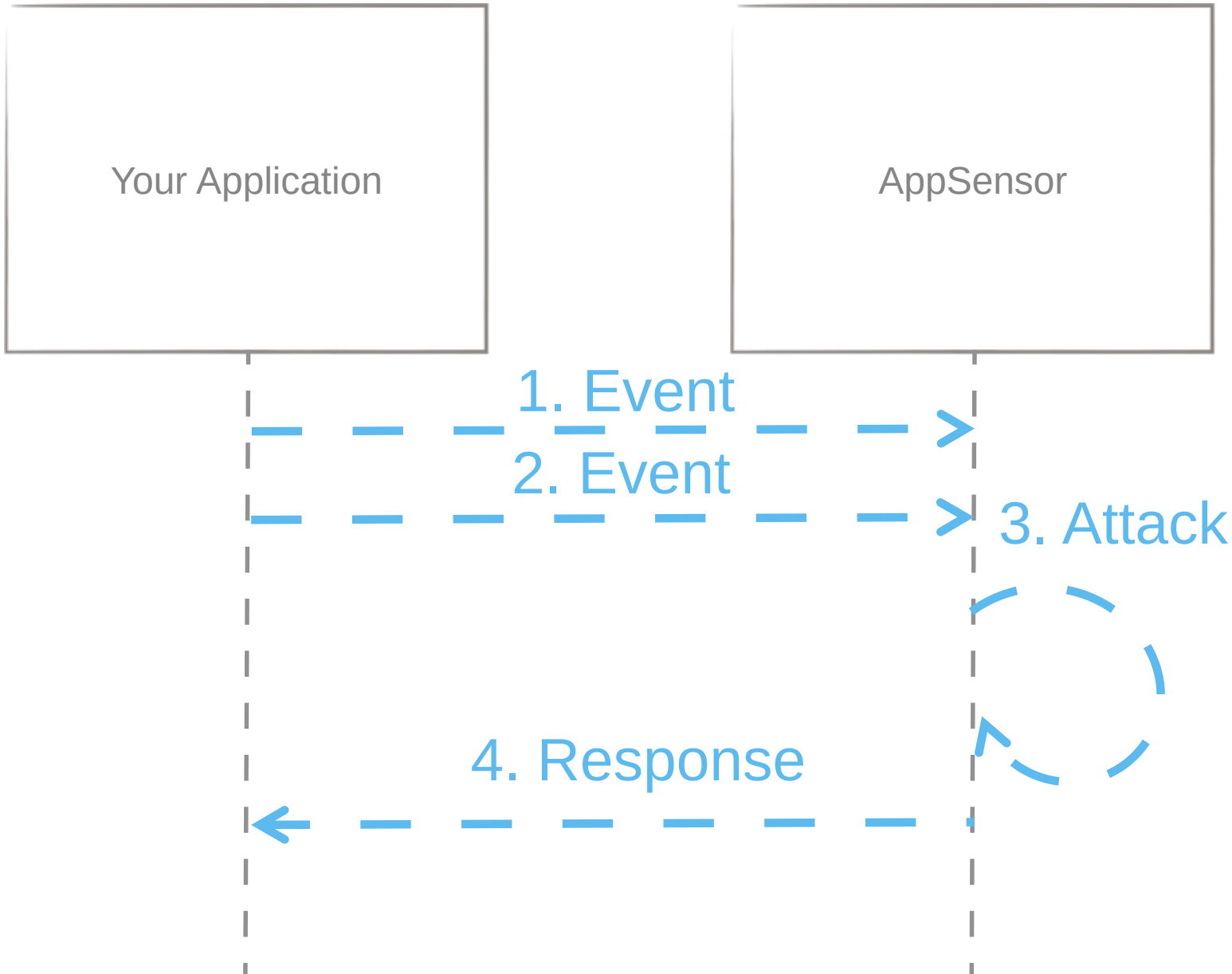
terminology

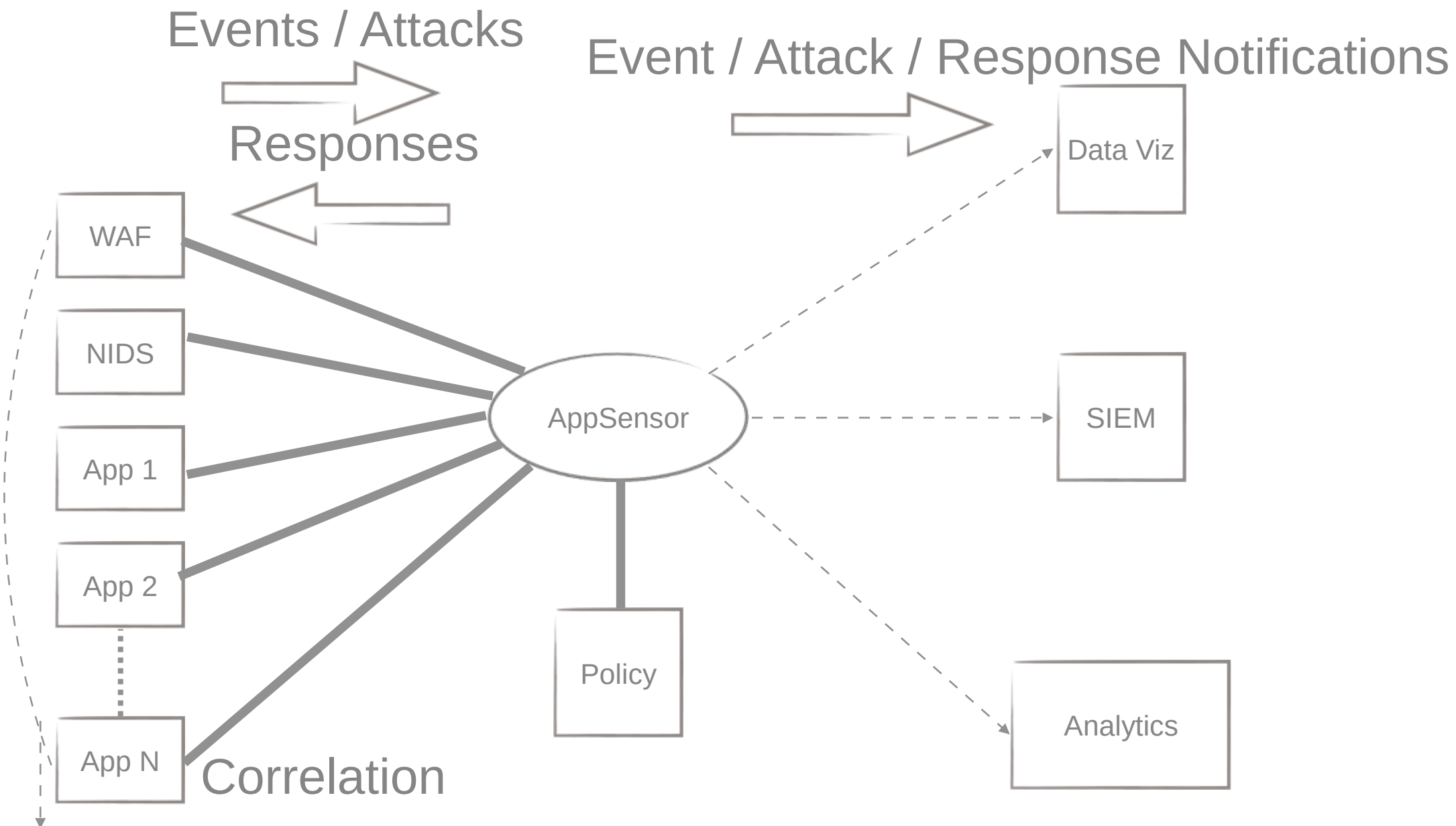
- **event** - suspicious
- attack - malicious (1 .. * events)
- response - take action (1 .. 1 attack)
- detection point - activity category (e.g. cookie modification)

… On the shoulders of giants …

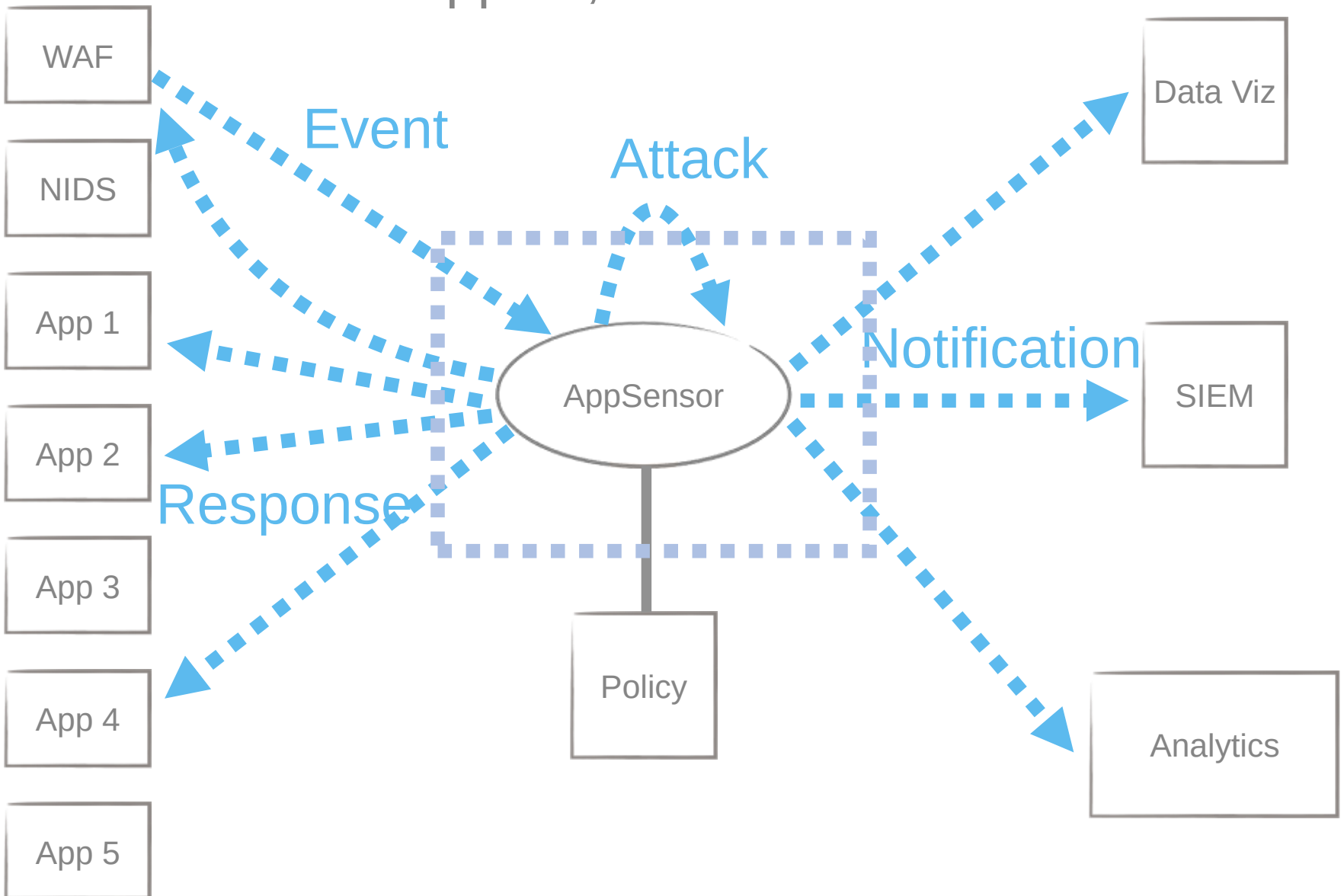
the tech

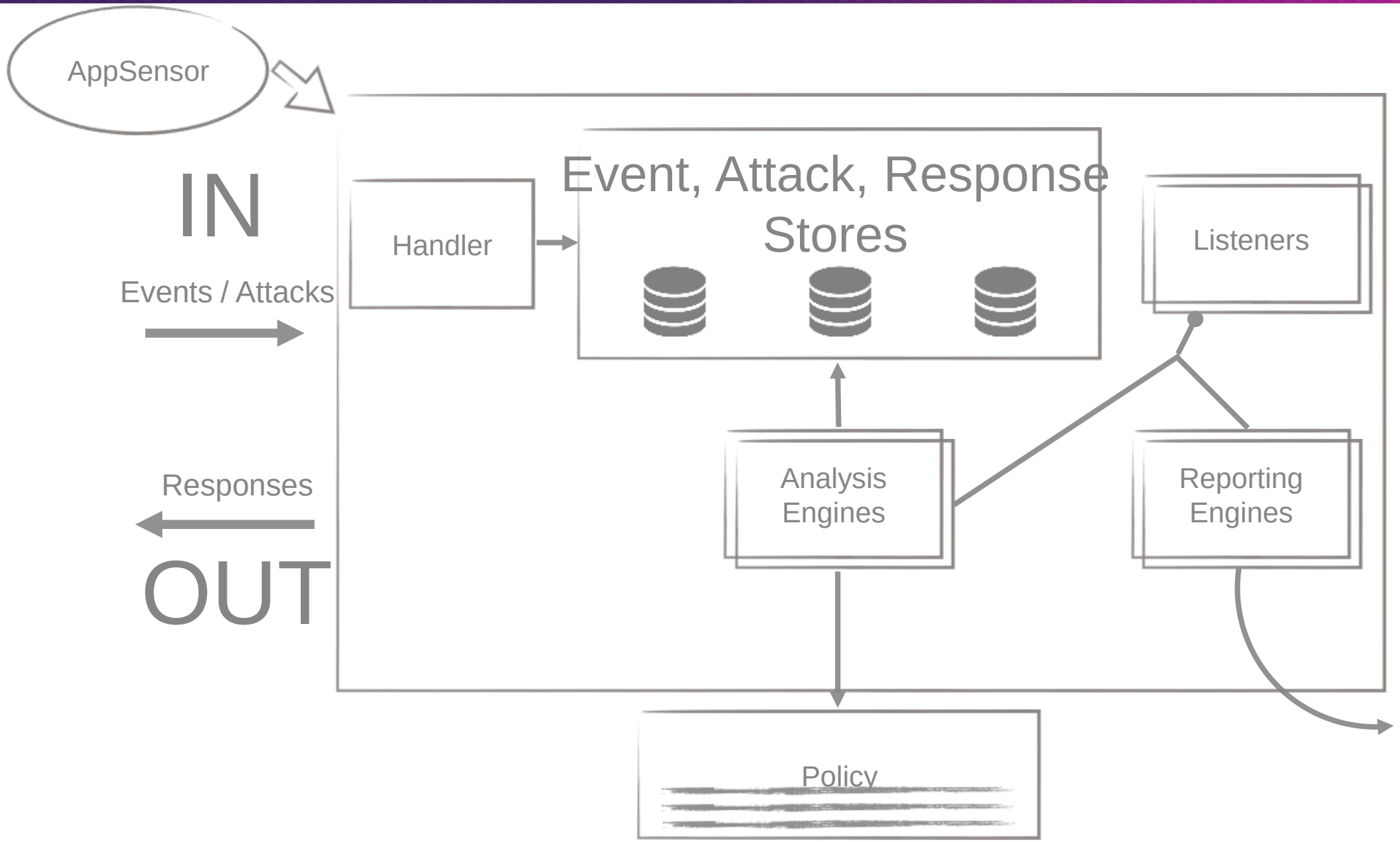

1. Event
2. Attack
Your Application
AppSensor
3. Response


Your Application
AppSensor
1. Event
2. Event
3. Attack
4. Response


Events / Attacks
Responses
Event / Attack / Response Notifications
WAF
NIDS
App 1
App 2
App N
AppSensor
Policy
Data Viz
SIEM
Analytics
Correlation

WAF & Apps 1, 2 and 4 are correlated




AppSensor
IN
Events / Attacks
Handler
Event, Attack, Response Stores
Listeners
Analysis Engines
Reporting Engines
Responses
OUT
Policy

Emitters

- ELK
- CEF / Syslog
- Influx / Grafana
- WebSocket
- JMX
- Prometheus

Framework Integration

- Spring Security

Configuration

- XML

Execution Modes

- REST
- Kafka
- ActiveMQ
- RabbitMQ
- Thrift
- SOAP
- embedded (jvm)

Storage Providers

- JPA2
- ElasticSearch
- Mongo
- Riak
- Influx
- File
- In-memory (testing)

adding detection points

- manually
- appsensor-reverse-proxy
- WAF (e.g. OWASP CRS in ModSecurity)

manual

```
POST /account/transfer HTTP/1.1

Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Win…)
Accept: text/html,application/xhtml+xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/account.php
Cookie: PHPSESSID=l9…lgt5
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
from_acct=xxx1234&to_acct=xxx9876&amt=20.00
```

manual

POST /account/transfer HTTP/1.1

Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Win…)
Accept: text/html,application/xhtml+xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/account.php
Cookie: PHPSESSID=l9…lgt5
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
from_acct=xxx1234&to_acct=xxx9876&amt=20.00

manual

```
@POST
public Response transfer(
      String from,
      String to,
      String amount) {

   transfer(from, to, amount);

   return Response.ok();
}
```

manual

```
@POST
public Response transfer(
      String from,
      String to,
      String amount) {

   if ( currentUser.owns(from) ) {
      transfer(from, to, amount);
   }

   return Response.ok();
}
```

manual

```
@POST
public Response transfer(
      String from,
      String to,
      String amount) {

   if ( currentUser.owns(from) ) {
      transfer(from, to, amount);
   } else {
      showErrorPage();    // normal error handling
   }

   return Response.ok();
}
```

manual

```
@POST
public Response transfer(
      String from,
      String to,
      String amount) {

   if ( currentUser.owns(from) ) {
      transfer(from, to, amount);
   } else {
       appsensor.addEvent( new Event(currentUser, "ACE2") );
      showErrorPage();      // normal error handling
   }

   return Response.ok();
}
```

appsensor-reverse-proxy

GitHub | This repository | Search | Explore | Features | Enterprise | Pricing | Sign up | Sign in

jtmelton / appsensor-reverse-proxy

Watch 1 | Star 0 | Fork 3

Code | Issues 0 | Pull requests 1 | Pulse | Graphs

Reverse proxy to front-end appsensor and implement various detection points

14 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | New file | Find file | HTTPS | https://github.com/jtmeltor | Download ZIP

jtmelton adding lots of core capabilities as well as tests — Latest commit be02566 13 days ago

Godeps	Adding redigo dependency	13 days ago
blocks	updating block refresh	13 days ago
connections	Adding redigo dependency for redis	13 days ago
ids	Adding redigo dependency for redis	13 days ago
middleware	code cleanup	13 days ago
testdata	Adding redigo dependency for redis	13 days ago
.gitignore	Initial gitignore	2 months ago
LICENSE	License update for OWASP team	2 months ago

WAF

- Send events and/or attacks
- Receive and process responses
- OWASP CRS in ModSecurity has AppSensor rules already
- https://www.trustwave.com/Resources/SpiderLabs-Blog/Implementing-AppSensor-Detection-Points-in-ModSecurity/

viewing data

- ELK stack (OWASP SoC)
- influxdb / grafana (OWASP SoC)
- appsensor-ui

AppSensor UI | Dashboard | GEO Map | Trends | Welcome, analyst | Configuration | About | Logout

Month 1171 / 71 | Week 1171 / 71 | Day 1171 / 71 | Shift 1172 / 71 | Hour 1172 / 71

Last Refresh: 2016-01-29T23:00:26-05:00

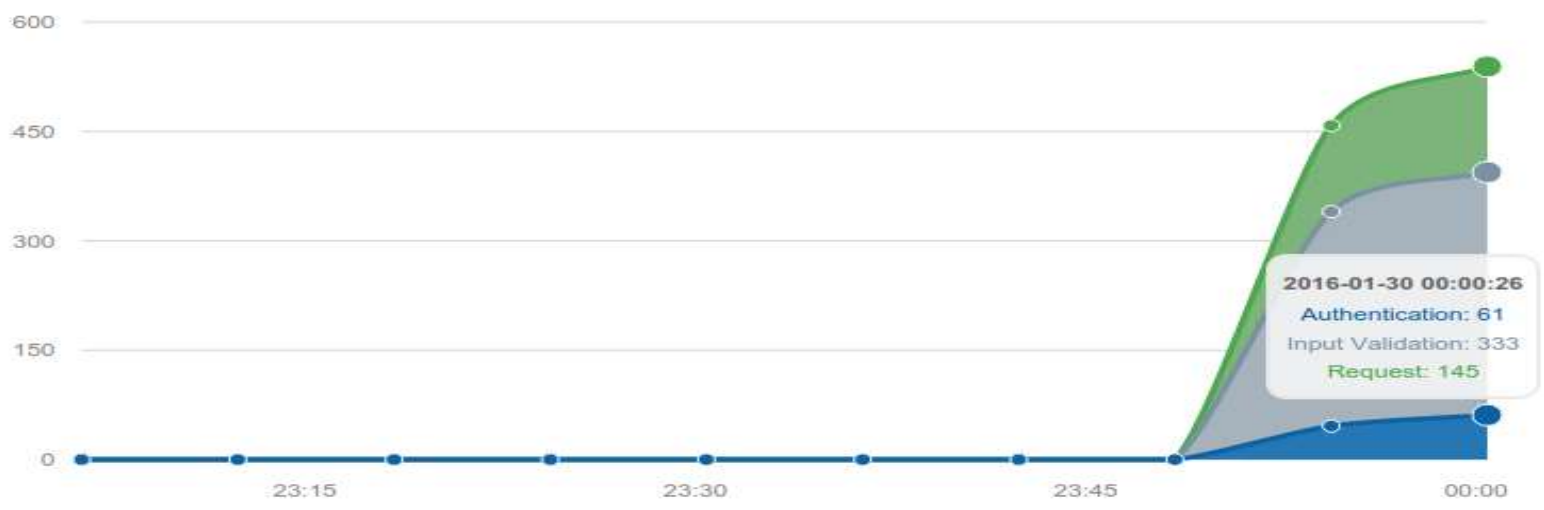
Detection Point Categories

Authentication 107 / 0

Input Validation 627 / 51

Request 263 / 20

Count By Categories



Most Active Detection Points

- IE1 (Input Validation) (454 events)
- RE3 (Request) (263 events)
- IE2 (Input Validation) (173 events)
- AE4 (Authentication) (107 events)

Most Active Users

- bob (515 events)
- cherie (130 events)
- susan (129 events)
- stephen (119 events)
- frank (104 events)

Active Responses

- disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:51:08")
- disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:50:29")
- disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:52:10")
- disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29

disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:50:29")

disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:52:10")

disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:52:46")

disableComponentForSpecificUser (33 minutes) (bob) (client app: "myclientapp") (started at: "2016/01/29, 23:51:38")

Activity Log (most recent)

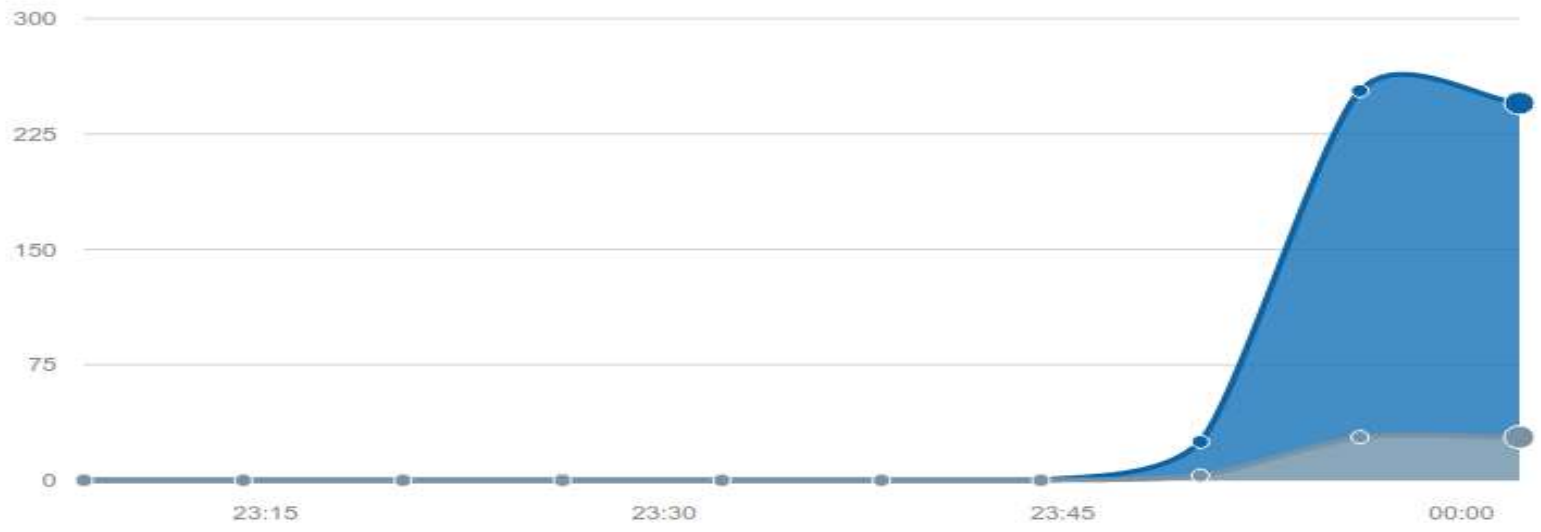
Type	Category	From	To	Timestamp
Event	RE3 (Request)	cherie (10.10.10.4) (-25.423505 / 27.106885)	myclientgeoapp2 (10.10.10.6) (-7.471493 / -47.248578)	2016-01-30T05:01:55.574Z
Event	IE1 (Input Validation)	stephen (10.10.10.3) (29.66889 / -8.576706)	myclientgeoapp3 (10.10.10.7) (59.164625 / 123.96234)	2016-01-30T05:01:55.162Z
Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:01:54.796Z
Event	AE4 (Authentication)	frank (10.10.10.1) (37.596758 / -121.647992)	myclientgeoapp3 (10.10.10.7) (59.164625 / 123.96234)	2016-01-30T05:01:54.479Z
Event	RE3 (Request)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:01:54.464Z
Event	IE1 (Input Validation)	stephen (10.10.10.3) (29.66889 / -8.576706)	myclientgeoapp1 (10.10.10.5) (52.629678 / -7.873585)	2016-01-30T05:01:54.150Z
Event	IE2 (Input Validation)	cherie (10.10.10.4) (-25.423505 / 27.106885)	myclientgeoapp4 (10.10.10.8) (12.875989 / 77.5561)	2016-01-30T05:01:52.986Z
Event	IE2 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:01:51.558Z
Event	IE1 (Input Validation)	cherie (10.10.10.4) (-25.423505 / 27.106885)	myclientgeoapp3 (10.10.10.7) (59.164625 / 123.96234)	2016-01-30T05:01:51.139Z
Response	logout	myclientapp (no IP Address) (no geo)	bob (10.10.10.1) (37.596758 / -121.647992)	undefined

Month 609 / 59 Week 609 / 59 Day 609 / 59 Shift 609 / 59 Hour 609 / 59

Last Refresh: 2016-01-29T23:02:11-05:00

Information For Detection Point IE1

Events By Count



2016-01-30 00:02:11
Events: 245
Attacks: 28

Latest Events

Category	User	Detection System	Timestamp
Input Validation	cherie	myclientgeoapp3	2016-01-30T05:02:11.232Z
Input Validation	bob	myclientapp	2016-01-30T05:02:09.907Z
Input Validation	bob	myclientapp	2016-01-30T05:02:08.887Z
Input Validation	frank	myclientgeoapp3	2016-01-30T05:02:08.220Z
Input Validation	bob	myclientapp	2016-01-30T05:02:05.867Z

Seen By These Client Applications

- myclientgeoapp1 (69 events)
- myclientgeoapp2 (84 events)
- myclientapp (297 events, 59 attacks)
- myclientgeoapp3 (73 events)

Most Active Users

- bob (297 events)
- susan (67 events)
- cherie (60 events)
- stephen (53 events)
- frank (46 events)

Associated Configuration

Category: Input Validation
Threshold: 5 in 20 seconds
Responses:
- log (for 10 minutes)
- logout

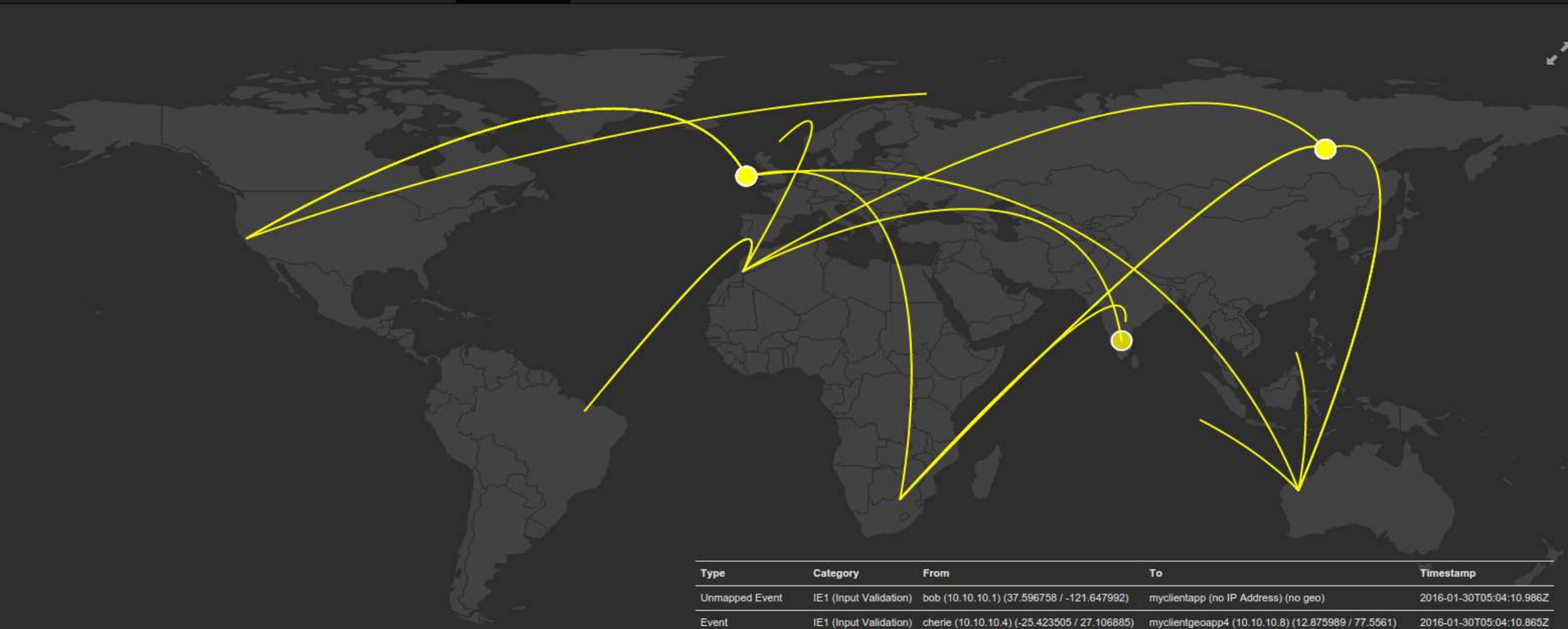
Category: Input Validation
Threshold: 30 in 1 minutes
Responses:
- disable (for 10 minutes)
- logout

Trends

Month (30 Days)	Week	Day	Shift (8 hours)	Hour
30 Dec - 30 Jan	23 Jan - 30 Jan	29 Jan - 30 Jan	17:23 - 01:23	00:23 - 01:23
2	9	67	201	0
0	0	4	12	0

Last updated: 2016/01/30, 01:23

AppSensor UI | Dashboard | GEO Map | Trends | Welcome, analyst | Configuration | About | Logout



Type	Category	From	To	Timestamp
Unmapped Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:04:10.986Z
Event	IE1 (Input Validation)	cherie (10.10.10.4) (-25.423505 / 27.106885)	myclientgeoapp4 (10.10.10.8) (12.875989 / 77.5561)	2016-01-30T05:04:10.865Z
Unmapped Event	IE2 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:04:09.760Z
Unmapped Response	logout	myclientapp (no IP Address) (no geo)	bob (10.10.10.1) (37.596758 / -121.647992)	undefined
Unmapped Attack	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	undefined
Unmapped Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:04:08.956Z
Event	IE2 (Input Validation)	susan (10.10.10.2) (-23.342331 / 117.810003)	myclientgeoapp2 (10.10.10.6) (-7.471493 / -47.248578)	2016-01-30T05:04:08.325Z
Event	RE3 (Request)	susan (10.10.10.2) (-23.342331 / 117.810003)	myclientgeoapp4 (10.10.10.8) (12.875989 / 77.5561)	2016-01-30T05:04:08.044Z
Unmapped Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:04:07.936Z
Unmapped Event	IE1 (Input Validation)	bob (10.10.10.1) (37.596758 / -121.647992)	myclientapp (no IP Address) (no geo)	2016-01-30T05:04:06.914Z

rules engine goals

- Expand detection capabilities by providing boolean logic and new span primitives
- Reduce false positives by leveraging several suspicious events to discover a malicious event

rules engine

- Multiple sensors grouped into single “Rule” to trigger an attack
- Rule combines sensors with AND/OR/NOT/THEN operators
- Thresholds can be lowered without increasing false-positive rate because there are multiple indicators
    - I.e. many SUSPICIOUS factors can define a MALICIOUS factor

example – default engine

Sensor1 - Multiple failed login attempts (50 attempts / 1 minute)

Rule: Sensor1

AND

Sensor1 - Multiple failed login attempts

Sensor2 - Use of blacklisted characters

Sensor3 - Password attempt too long

Sensor4 - Multiple usernames attempted from single IP

Rule: Sensor1 AND Sensor2 AND Sensor3 AND Sensor4

OR

Sensor1 - Multiple failed login attempts

Sensor2 - Use of blacklisted characters

Sensor3 - Password attempt too long

Sensor4 - Multiple users attempting to login from single IP

Rule: Sensor1 AND (Sensor2 OR Sensor3 OR Sensor4)

THEN

Sensor1 - Use of blacklisted characters

Sensor2 - Large file upload

Sensor3 - Large file download

Sensor1 THEN (Sensor2 OR Sensor3)

combination rules

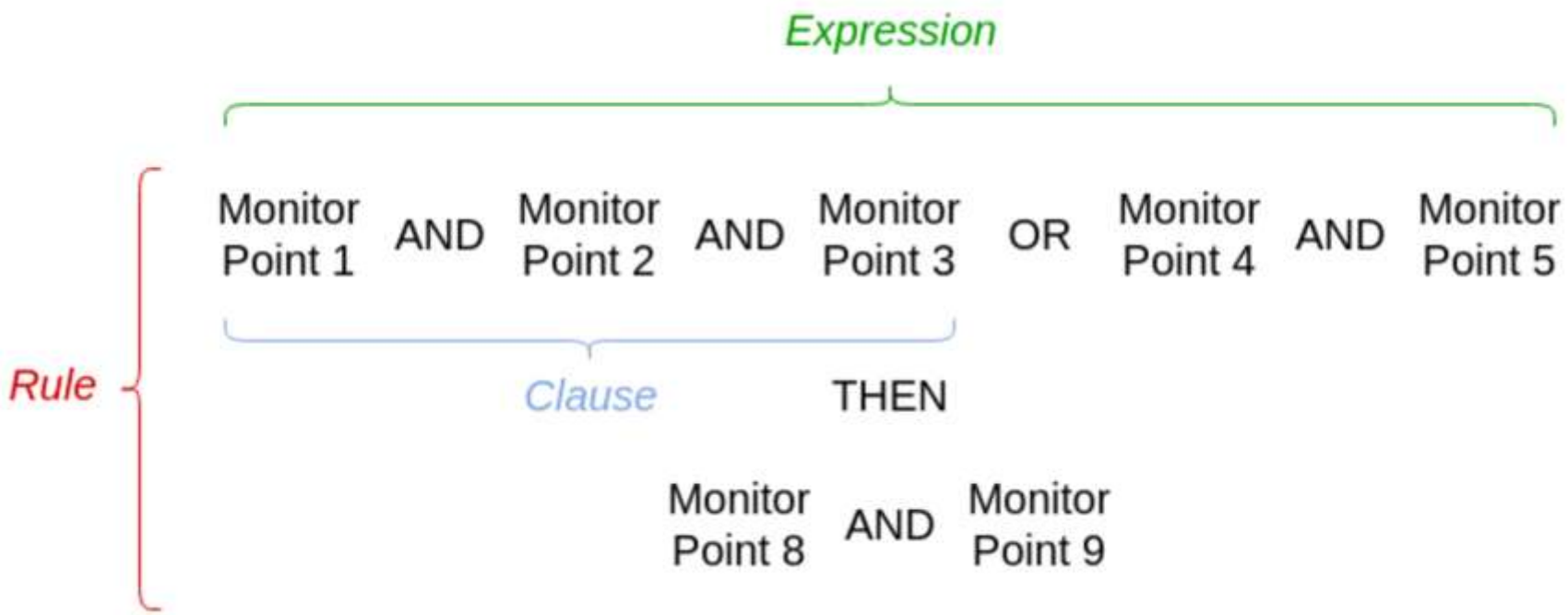
Sensor1 OR Sensor2

THEN

Sensor3 AND (Sensor4 OR Sensor5)

THEN

Sensor6 AND Sensor7 AND Sensor8 AND Sensor9 AND Sensor10


Expression
Monitor Point 1
AND
Monitor Point 2
AND
Monitor Point 3
OR
Monitor Point 4
AND
Monitor Point 5
Rule
Clause
THEN
Monitor Point 8
AND
Monitor Point 9

Under the Hood

Sensor1
THEN
Sensor2

1. Collect Events

Under the Hood

Sensor1
THEN
Sensor2

1. Collect Events

Under the Hood

Sensor1
THEN
Sensor2

Under the Hood

1. Collect Events

2. First Expression

Sensor1
THEN
Sensor2

Under the Hood

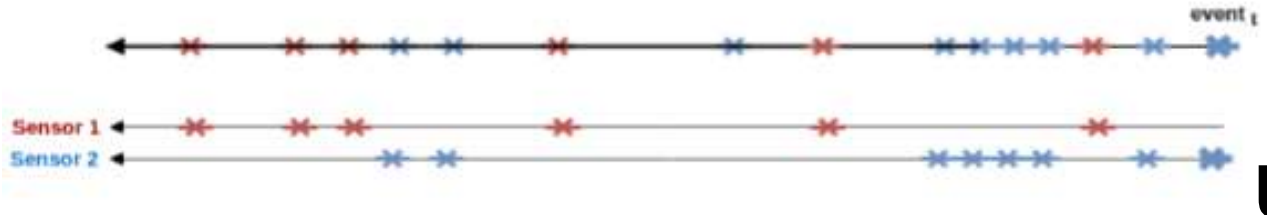
1. Collect Events

2. First Expression

Sensor1
THEN
Sensor2

Under the Hood

1. Collect Events

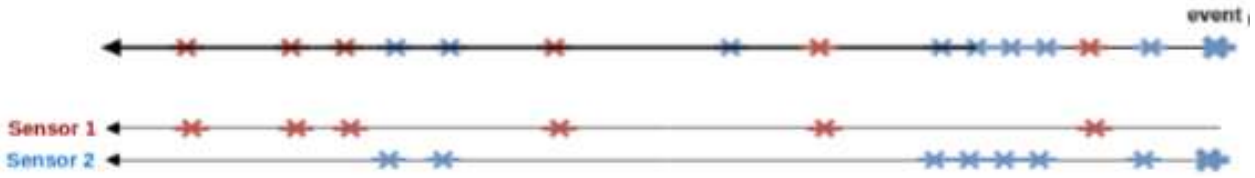


2. First Expression

Sensor1
THEN
Sensor2

Under the Hood

1. Collect Events

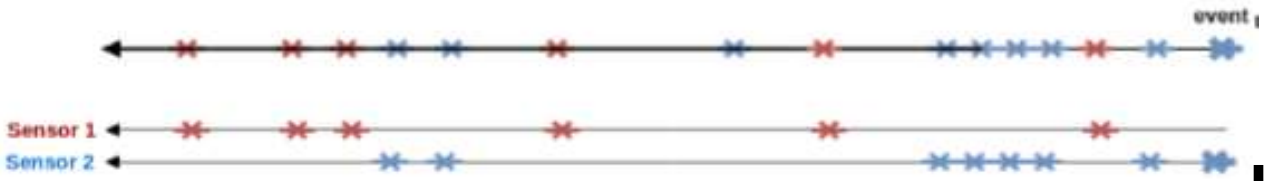


2. First Expression

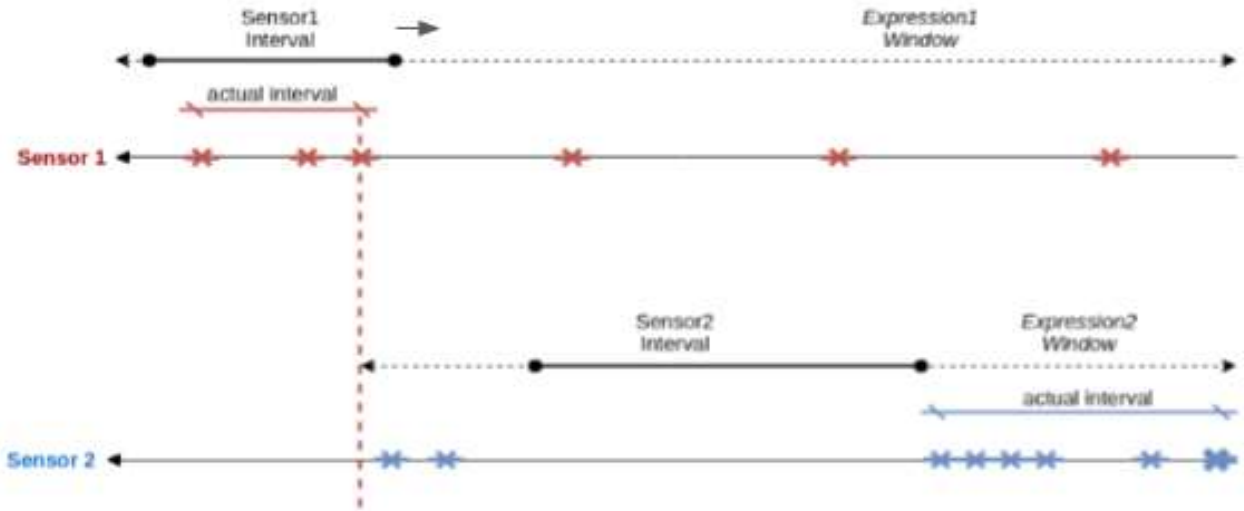
Sensor1
THEN
Sensor2

Under the Hood

1. Collect Events



2. First Expression

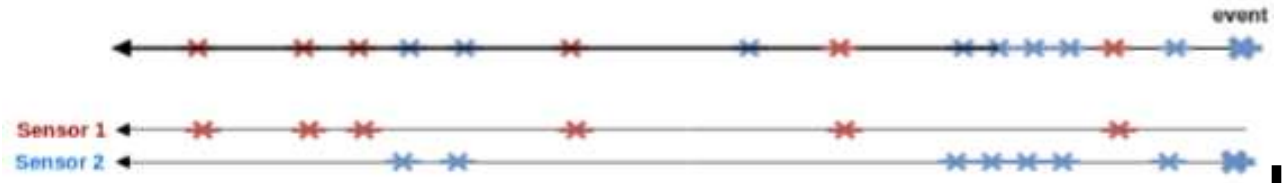


3. Second Expression

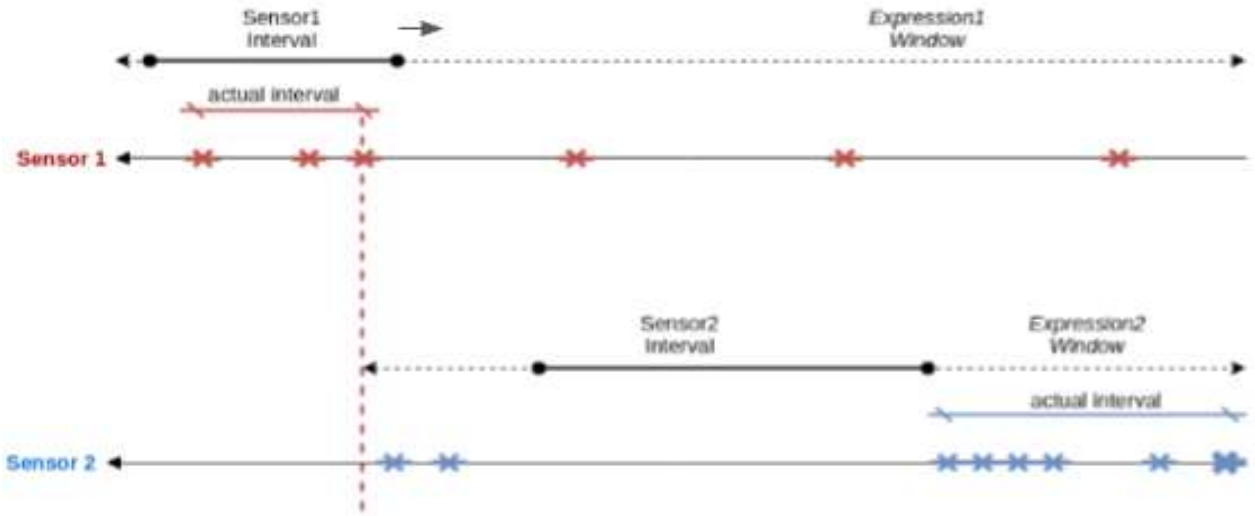
Sensor1
THEN
Sensor2

Under the Hood

1. Collect Events



2. First Expression



3. Second Expression

Sensor1
THEN
Sensor2

4. Rule Triggered

Thanks David Scrobonia - Landed!

analysis engines

Basic	Trend	Machine Learning	Rules
Simple thresholds	Large user changes in user base or application	Anomaly Detection	Aggregation of simple thresholds

server assembler

- Generate your server app!
- Easily select your components and generate a proper app
- Instructions for what config changes to make (db passwords, header names, etc.)

APPSENSOR STARTER From detection to response

Generate a Pom file With Websocket Boot with Appsensor

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Appsensor dependencies

Search for dependencies

Selected Dependencies

Generate Project ⌘ + ⏎

Don't know what to look for? Want more options? Switch to the full version.

Too many options? Switch back to the simple version.

Generate Project ⌘ + ⏎

Execution Modes

- [ ] ActiveMQ
  It is ActiveMQ
- [ ] RabbitMQ
  Follow the Rabbit
- [ ] Kafka
  The Kafka
- [ ] Thrift
  Thrift!
- [ ] Rest
  Just like all the rest
- [ ] Soap
  Clean up

Geolocators

- [ ] Appsensor Geolocation - GeoIP2
  Geolocate everyone

Integrations

- [ ] CEF over Syslog
  All Syslog all the time
- [ ] InfluxDB
  Acid Influx DB
- [ ] JMX
  To JMX or not to JMX
- [ ] Prometheus
  And Bob
- [ ] Spring Security
  Secure all the Springs

ADDO
ALL DAY DEVOPS

Reporting Engines

- [ ] Simple Logging
  Log simple
- [ ] WebSocket
  Socket all the Webs

Storage Providers

- [ ] In Memory
  Do it in memory
- [ ] Flat file on disk
  The flatter the file...
- [ ] InfluxDB
  Influx the DB
- [ ] Jpa2
  Basically any SQL-compliant DB
- [ ] Elastic Search
  Search with Elastic
- [ ] MongoDB
  Mongo
- [ ] Riak
  riak

Generate Project ⌘ + ⏎

mvnw	Today, 9:46 AM	7 KB	Unix e...c
mvnw.cmd	Today, 9:46 AM	5 KB	Docume
pom.xml	Today, 9:46 AM	11 KB	XML
src	Today, 9:47 AM	--	Folder
main	Today, 9:47 AM	--	Folder
java	Today, 9:47 AM	--	Folder
com	Today, 9:47 AM	--	Folder
example	Today, 9:47 AM	--	Folder
DemoApplication.java	Today, 9:46 AM	2 KB	Java sou
resources	Today, 9:47 AM	--	Folder
application.properties	Today, 9:46 AM	Zero bytes	Java p...t
test	Today, 9:47 AM	--	Folder
java	Today, 9:47 AM	--	Folder
com	Today, 9:47 AM	--	Folder
example	Today, 9:47 AM	--	Folder
DemoAppl...nTests.java	Today, 9:46 AM	326 bytes	Java sou

```java
package com.example;

import org.owasp.appsensor.core.AppSensorClient;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.FilterType;
import org.springframework.web.socket.server.standard.ServerEndpointExporter;


@Configuration
@EnableAutoConfiguration
@ComponentScan(value="com.example", excludeFilters = @ComponentScan.Filter(value = AppSensorClient.class, type =
FilterType.ASSIGNABLE_TYPE))
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

    @Bean
    public ServerEndpointExporter serverEndpointExporter() {
        return new ServerEndpointExporter();
    }

}
```

wrap-up

related projects

- repsheet
- ensnare
- fido
- riemann
- apache eagle
- devsecops
- elastalert
- fouroneone

pick a tool (or 2) ...

but <u>use</u> the idea

contributors

- https://www.owasp.org/index.php/OWASP_AppSensor_Project#tab=Acknowledgements
- https://github.com/jtmelton/appsensor/graphs/contributors

Project Founder

- Michael Coates @

Project Leaders

- Dennis Groves @
- John Melton @
- Colin Watson @

- Josh Amishav-Zlatin
- Ryan Barnett
- Simon Bennetts
- Joe Bernik
- Rex Booth
- Luke Briner
- Rauf Butt
- Juan C Calderon
- Fabio Cerullo
- Marc Chisinevski
- Robert Chojnacki
- Michael Coates
- Dinis Cruz
- Sumanth Damaria
- August Detlefsen
- Ryan Dewhurst
- Sean Fay
- Timo Goosen
- Dennis Groves
- Randy Janida
- Chetan Karande
- Eoin Keary
- Alex Lauerman
- Junior Lazuardi
- Benjamin-Hugo LeBlanc
- Jason Li
- Manuel López Arredondo
- Bob Maier
- Jim Manico
- Sherif Mansour Farag
- John Melton
- Mark Miller
- Rich Mogull
- Craig Munson
- Louis Nadeau
- Giri Nambari
- Erlend Oftedal
- Jay Reynolds
- Chris Schmidt
- Sahil Shah
- Eric Sheridan
- John Steven
- Raphael Taban
- Alex Thissen
- Don Thomas
- Christopher Tidball
- Stephen de Vries
- Kevin W Wall
- Colin Watson
- Mehmet Yilmaz

links

- https://www.owasp.org/index.php/OWASP_AppSensor_Project (download book, dev guide, etc.)
- http://appsensor.org/ (end user / dev docs)
- https://github.com/jtmelton/appsensor

ADDO
ALL DAY DEVOPS

SPONSORS

Sponsorship packages for All Day DevOps are available. If your organization is interested, please contact us for details.

DIAMOND SPONSORS

sonatype

StateFarm

GOLD SPONSORS

vmware

Fannie Mae

NORTHROP GRUMMAN

COMMUNITY ADVOCATES AND VIEWING PARTY SPONSORS

XebiaLabs

cloudbees

Carnegie Mellon University Software Engineering Institute

TIGERA

CONTINO

NowSecure

MEDIA SPONSORS

DZone

TechBeacon

DevOps.com

MediaOps

ITSP MAGAZINE

the cyberwire

?