OWASP AppSensor Project

Real-time attack detection and response

- Colin Watson
  colin.watson(at)owasp.org

- Project primary contributors
  - Michael Coates
  - Dennis Groves
  - John Melton
  - Ryan Barnett

- Software defences
- Application-specific attack detection
- Architectures
- Signalling
- Example configuration

One issue

- Skilled and motivated attackers

Two questions

1) Is the application being attacked now?

2) Have any unknown vulnerabilities been exploited today?

☐ Yes ☐ No [✓ Don't know]

Three test cases

1) Stepping through a process in the incorrect order

Step five,	/step5/
then step two	/step2/

2) Requesting an unauthorised resource identifier

Show my account,	/updateProfile?id=1005
then show me someone else's	/updateProfile?id=1006

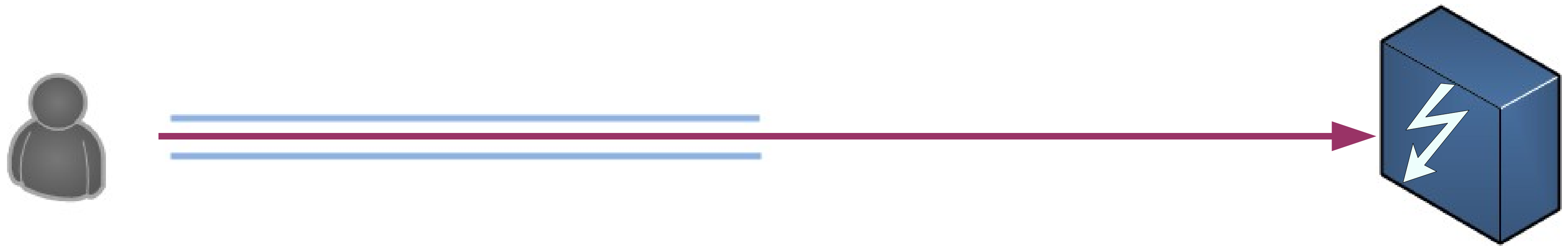
3) Payment transfer exceeding limit

Send 27 pounds,	/transfer?amount=27.00
then send rather more	/transfer?amount=270000

Four conventional defenses

1) Transport layer security
2) Firewall (stateful/deep packet inspection)
3) Web application firewall
4) Application aware firewall (“next generation”)

Transport layer security (SSL)

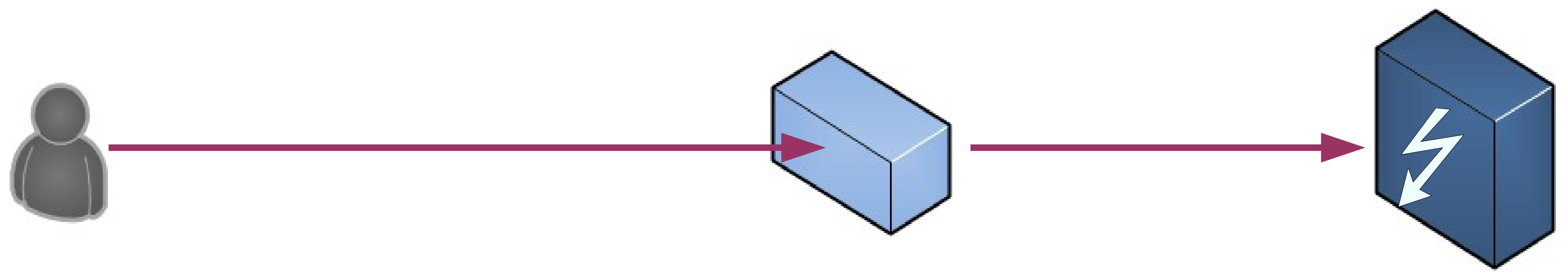


3) Payment transfer exceeding limit

Send 27 pounds,	/transfer?amount=27.00
then send rather more	/transfer?amount=270000

☐ Protected [✓ ☒ Unprotected]

Firewall



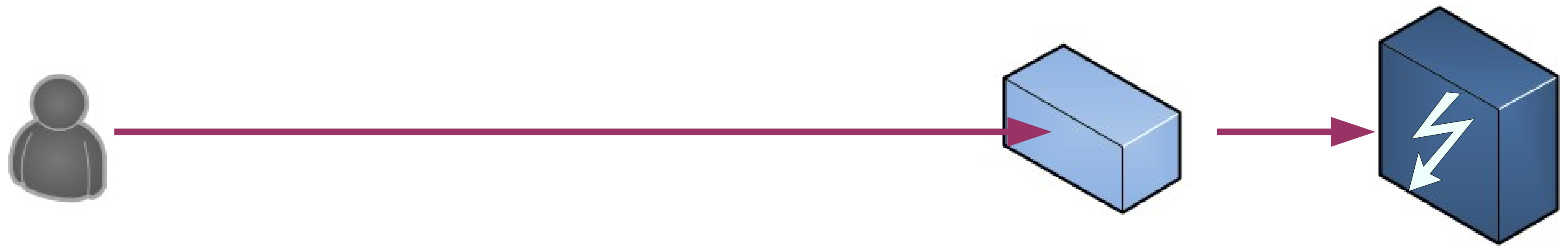
3) Payment transfer exceeding limit

Send 27 pounds,
then send rather more

/transfer?amount=27.00
/transfer?amount=270000

☐ Protected [✓ Unprotected]

Web application firewall

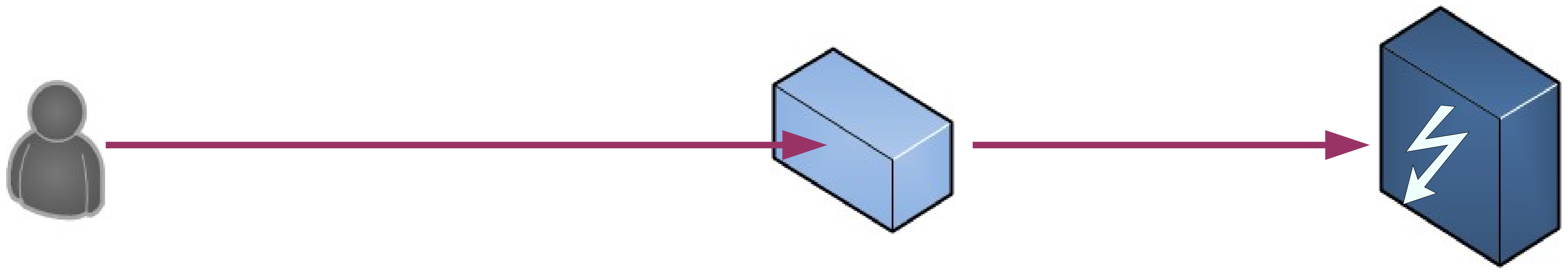


2) Requesting an unauthorised resource identifier

Show my account,	/updateProfile?id=1005
then show me someone else's	/updateProfile?id=1006

☐ Protected [✓ Unprotected]

Application aware firewall



1) Stepping through a process in the incorrect order

Step five,	/step5/
then step two	/step2/

☐ Protected ☒ Unprotected

Proper attack detection

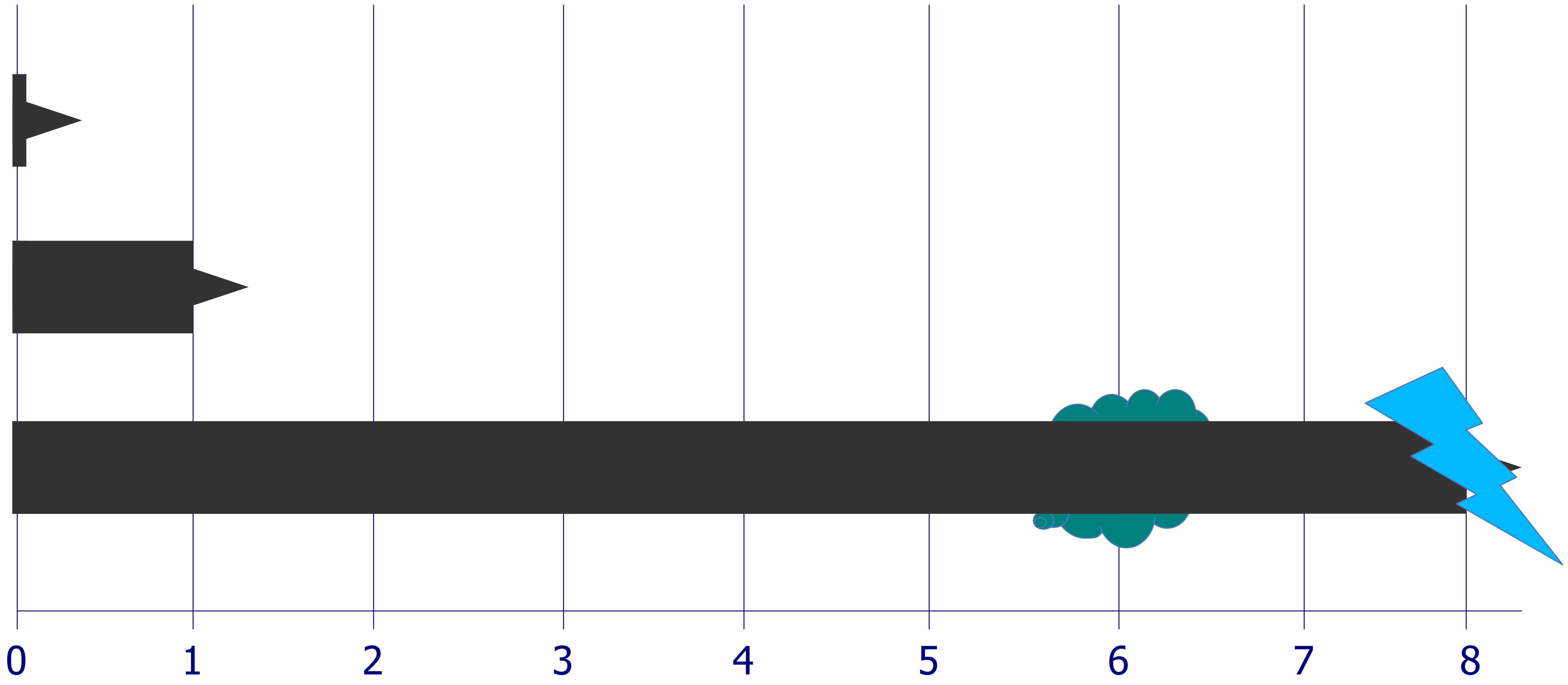
- Integrated
    - Understands the application
    - Understands normal vs. malicious use
    - Updated when the business process changes
- Effective
    - Minimal false positives
    - Immediate response
- Scalable and performant
    - Automatic detection
    - Real time

Inside the application

- Applications have:
    - Full knowledge of the business logic
    - An understanding of the roles & permissions of users
    - Knowledge of malicious vs. normal use
    - Access to user and system history and trends
    - Information to instantly detect attackers
    - The ability to respond automatically in real-time such as taking a more defensive posture

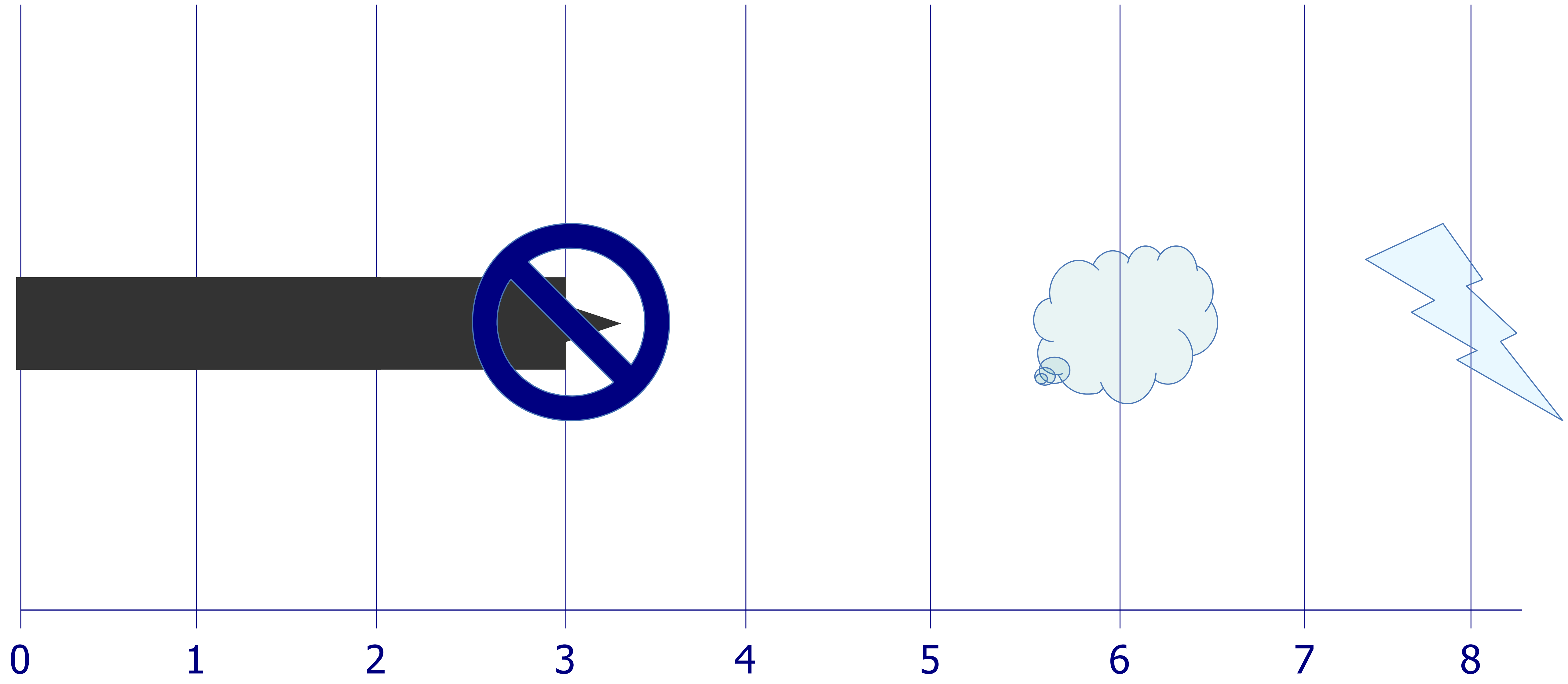
The concept

- Detect clearly malicious activity
- Stop an attacker before they can find vulnerabilities and exploit them



Identification of attackers, not particular attacks

- Think tripwires rather than perimeter walls



Existing application countermeasures

- Non-critical functions disabled by a car engine management system when intrusion detected
- Blocking access to an airplane's avionics control system when the source is identified as coming from the passenger network
- Tamper detection erases encryption keys
- Raising an alert when the external time is detected to be different to an internal time reference
- Logging of system power outages
- Application disabled by an operator due to unusual conditions
- Access blocked when single sign on message fails integrity check
- Application logging
- Disable non-core function

Existing countermeasures (continued)

- Terminating a request when blacklisted inputs are received
- Fraud detection
- Adding time delays to each successive failed authentication attempt
- Locking a user account after a number of failed authentication attempts
- Application honey pot functionality
- Logging a user out if the browser's “back” button is used
- Terminating a session if a user's geo-location changes
- Blocking access by certain IP addresses when malicious behavior is detected
- Recording unexpected actions
- Blocking certain HTTP verbs

Attack-Aware with Active Defences

1) Event detection
2) Analysis
3) Attack determination
4) Response selection
5) Response execution

Application attack detection points

- Request
- Authentication
- Session
- Access control
- Input
- Encoding
- Command injection
- File input/output
- Honey trap

- Custom

- User trend
- System trend

- Reputation

Pseudo code

- Existing error/exception/event trapping code

```
IF event1 THEN
   log
   display error message
ENDIF
```

- Extend existing

```
IF event1 THEN
   log
   send detection point data to analysis engine
   receive analysis engine response decision
   IF response THEN
       execute response
   ELSE
       display error message
   ENDIF
ENDIF
```

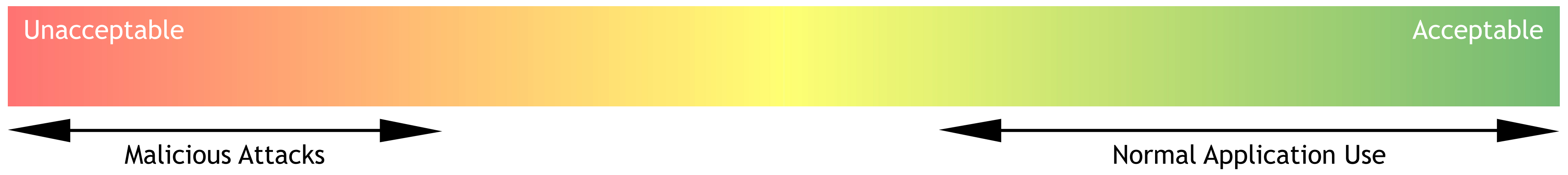
Pseudo code (continued)

- Create new error/exception/event trapping code

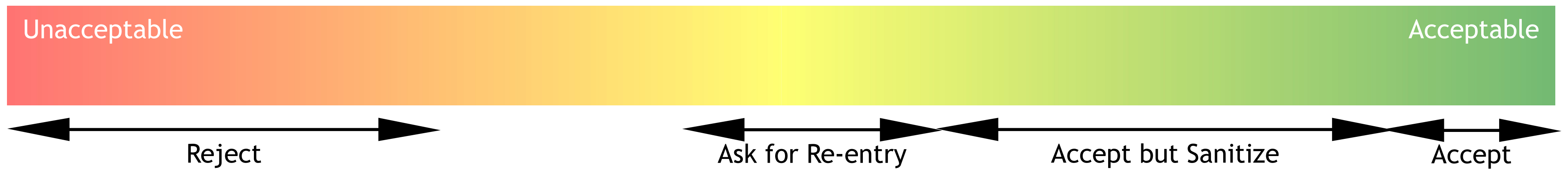
```
IF event3 THEN
   log
   send detection point data to analysis engine
   receive analysis engine response decision
   IF response THEN
       execute response
   ENDIF
ENDIF
```

Detecting Malicious Users

- “Users” are not perfect

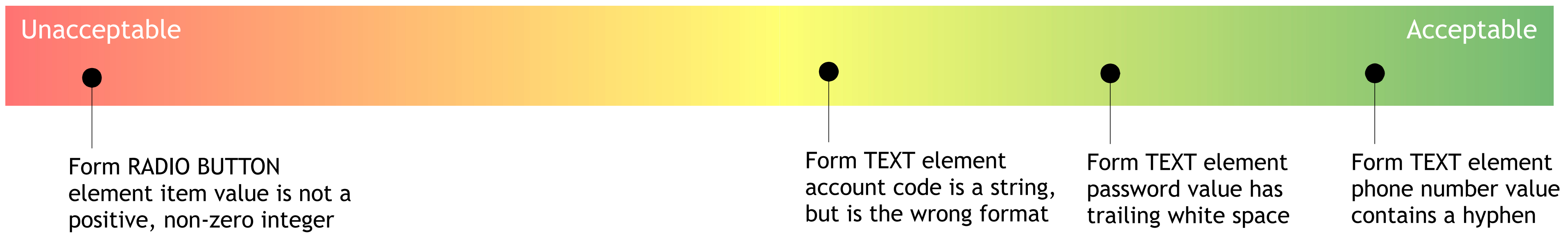


- Application-specific actions

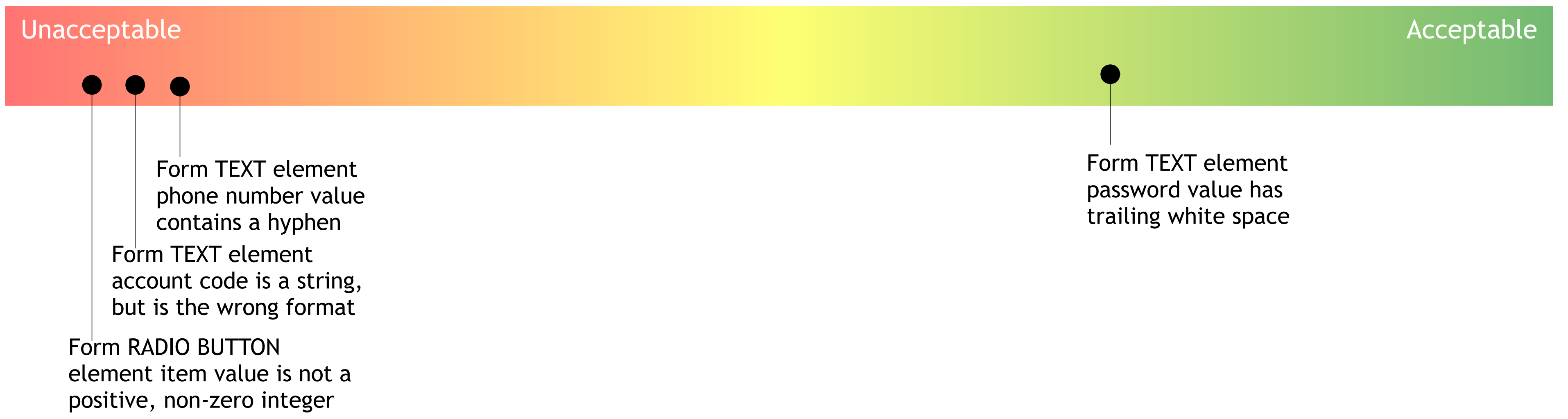


Importance of Context

- Server-side validation only


Unacceptable
Acceptable
Form RADIO BUTTON element item value is not a positive, non-zero integer
Form TEXT element account code is a string, but is the wrong format
Form TEXT element password value has trailing white space
Form TEXT element phone number value contains a hyphen

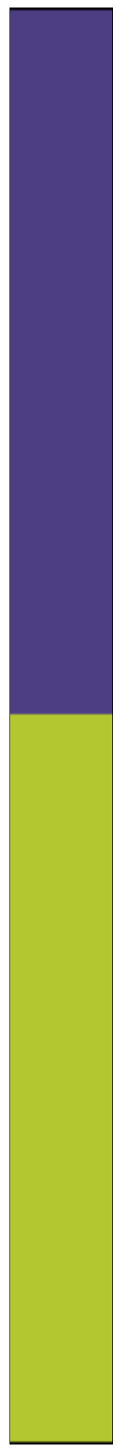
- Server-side with duplicate client-side validation


Unacceptable
Acceptable
Form TEXT element phone number value contains a hyphen
Form TEXT element account code is a string, but is the wrong format
Form RADIO BUTTON element item value is not a positive, non-zero integer
Form TEXT element password value has trailing white space

Conventional attack responses

- No change (e.g. just continue logging)

- Process terminated (e.g. reset connection)



Full spectrum responses

- **No change**
- Logging increased
- Administrator notification
- Other notification (e.g. other system)
- Proxy
- User status change
- User notification
- Timing change
- **Process terminated**
- Function amended
- Function disabled
- Account log out
- Account lock out
- Application disabled
- Collect data from user

Application response capabilities

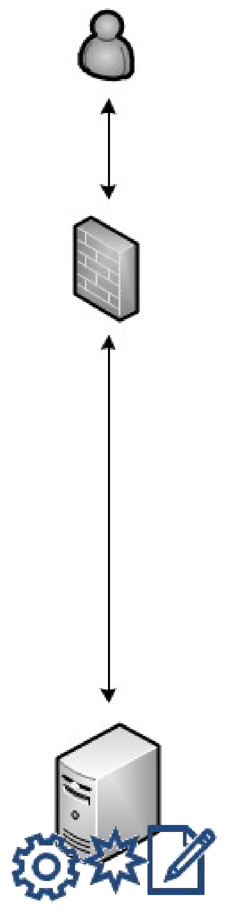
- Often already exist
    - Logging level
    - Alerting (email?)
    - User messages
    - Logout
    - Account lockout
    - Redirects
- Much less likely to exist
    - Proxy
    - Adding delays
    - Disabling individual functions/processes
    - Disabling the application

Implementation

- New project requirements
- Retrofitting existing applications
- Preliminary requirements
    - Application logging
    - Application risk assessment
    - Secure coding
- Monitoring and tuning

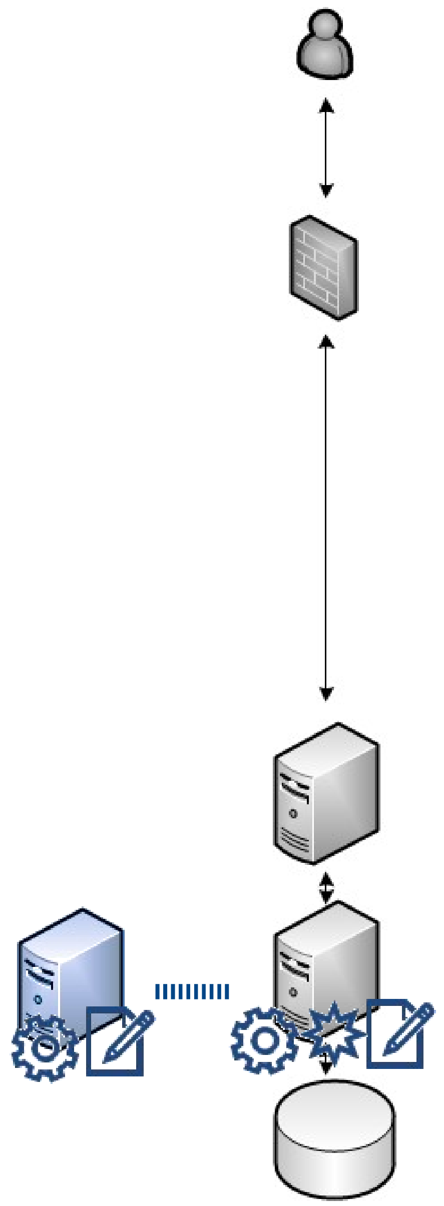
Architectures

- Logging
- Detection
- Analysis



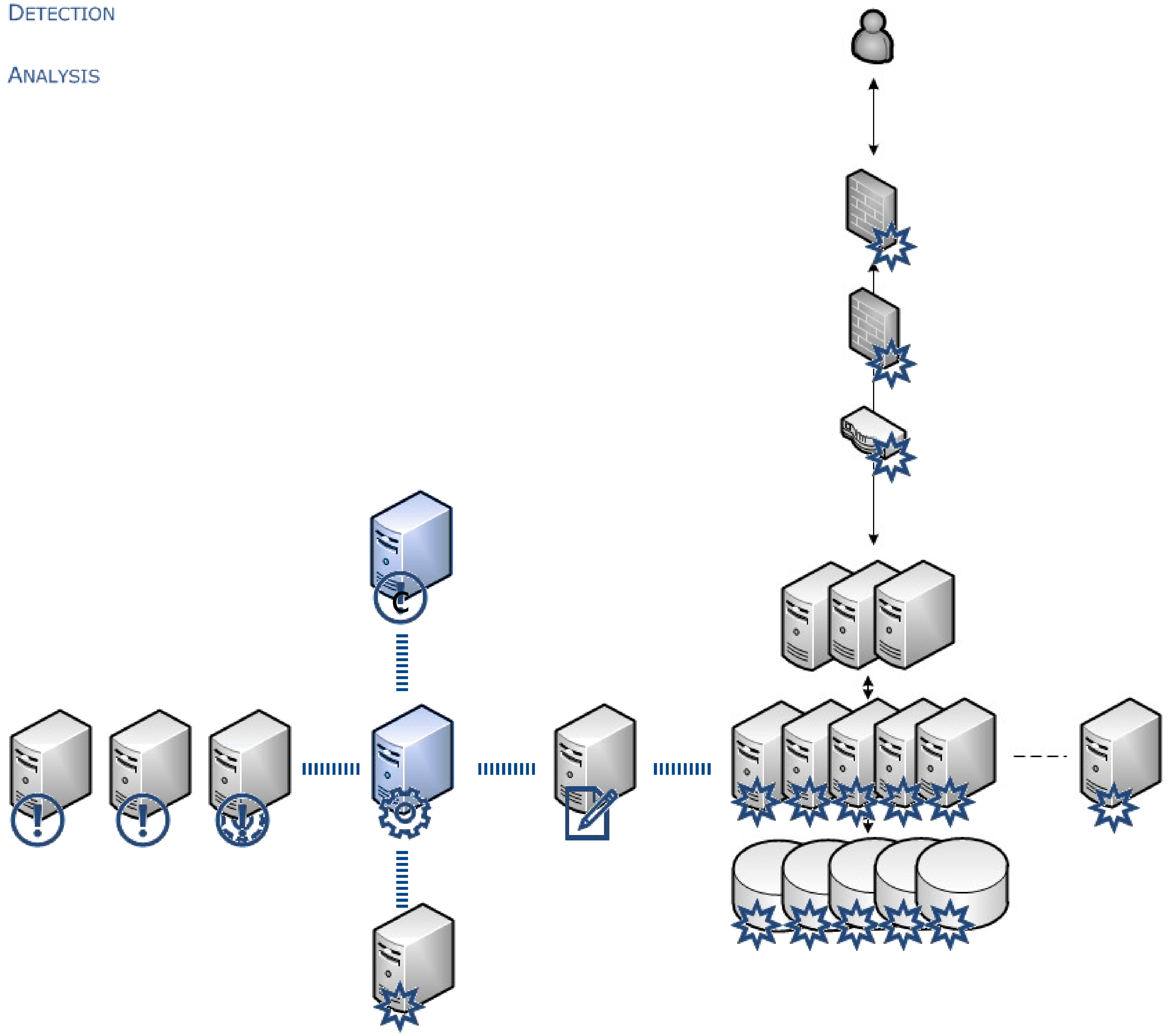
Architectures (continued)

- Logging
- Detection
- Analysis



Architectures (continued)

- Logging
- Detection
- Analysis



No 1 - Ecommerce Website Base Configuration

- Key risks
  - Product pricing errors, discounts and fiddles
  - Order process manipulation
  - Payment card mis-use
  - Personal data loss
- AppSensor detection points
  - General request filtering
  - Catalogue, basket and payment functions
  - Database

No 1 - Detection Points

Area	Identifier	#	AppSensor ID(s)	Notes
Request	R01	R	RE1, RE2, RE3, RE4	Invalid and incorrect HTTP verb
	R02	R	CIE1	SQL injection attempt
	R03	R	IE1	Cross site scripting (XSS) attempt
Catalogue	C01		IE4	Product value mismatch
Basket	B01		IE4	Basket value mismatch
Payment	P01		-	Card authorisation failure
	P02		IE4	Price mismatch between order and payment
Database	D01	+	CIE2	Returned record set size incorrect
	D02	+	IE5	Database table integrity fault

AppSensor detection point type identities and descriptions
https://www.owasp.org/index.php/AppSensor_DetectionPoints

No 1 – Response Actions

Area/Sensors	Description	Threshold	AppSensor ID(s)
Request R01, R02, R03	Block request	1	G
	Log out authenticated user	3	J
	Block IP address (and customer account if known) for whole site (manual reset)	6	L (and K)
Catalogue/Basket C01, C02	Alert operations staff	1	B
	Block IP address for dynamic areas (1 day, auto reset)	2	I
Payment P01	Alert operations staff / Redirect back to from checkout pages to the shopping basket summary	3	B / G
Payment P02	Alert operations staff / Put order on hold / Block future order check-out for the customer (manual reset)	1	B / D / I
Database D01	Alert operations staff / Abort process / Display error page / Block customer account (manual reset)	1	B / G / E / K
Database D02	Alert DBA and operations staff	1	B
[All]	Increase application logging granularity / Indicate on monitoring dashboard	1	A / C

AppSensor response action type identities and descriptions
https://www.owasp.org/index.php/AppSensor_ResponseActions

Unknown attacks

- [This list is intentionally left blank]

Two question revisited

1) Is the application being attacked now?

2) Have any unknown vulnerabilities been exploited today?

☐ Yes ☐ No ~~☐ Don't know~~

Further Explanations and Detailed Documentation

- Video presentations by Michael Coates, AppSensor Project Leader:
  - Automated Application Defenses to Thwart Advanced Attackers, June 2010
    http://michael-coates.blogspot.com/2010/06/online-presentation-thursday-automated.html
  - Attack Aware Applications, April 2011
    https://www.owasp.org/index.php/Minneapolis_St_Paul#tab=Video.2FAudio.2FSlides.2FHandouts
- Videos of AppSensor attack detection demonstrations:
  - AppSensor Project media
    https://www.owasp.org/index.php/Minneapolis_St_Paul#tab=Video.2FAudio.2FSlides.2FHandouts
- Written guidance:
  - OWASP AppSensor, v1.1, Michael Coates, 2008
    https://www.owasp.org/images/2/2f/OWASP_AppSensor_Beta_1.1.pdf
  - Implementation Planning Methodology, Colin Watson, 2010
    https://www.owasp.org/index.php/File:Appsensor-planning.zip
  - Developer Guide (for use with ESAPI)
    https://www.owasp.org/index.php/AppSensor_Developer_Guide

Make contact

Colin Watson

- colin.watson(at)owasp.org

AppSensor Project

- https://www.owasp.org/index.php/Category:OWASP_AppSensor_Project