OWASP AppSensor Project

Patterns for Logging, Architecture & Signalling

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

- Application-specific attack detection
- Logging
- Architectures
- Signalling
- Example web applications
- Dashboard demonstrations

One issue

- Advanced 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 (TLS, formerly SSL)

2) Firewall

3) Deep packet inspection

4) Web application firewall

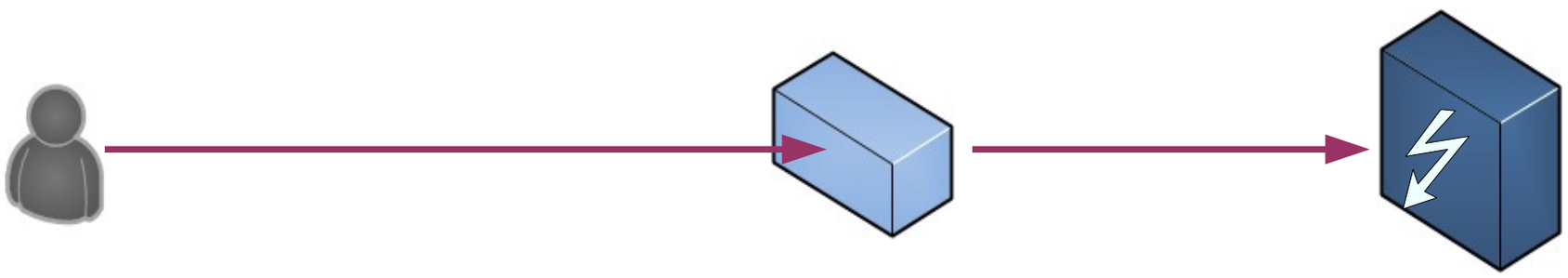
Transport layer security (SSL)

3) Payment transfer exceeding limit

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

☐ Protected ☒ Unprotected

Firewall



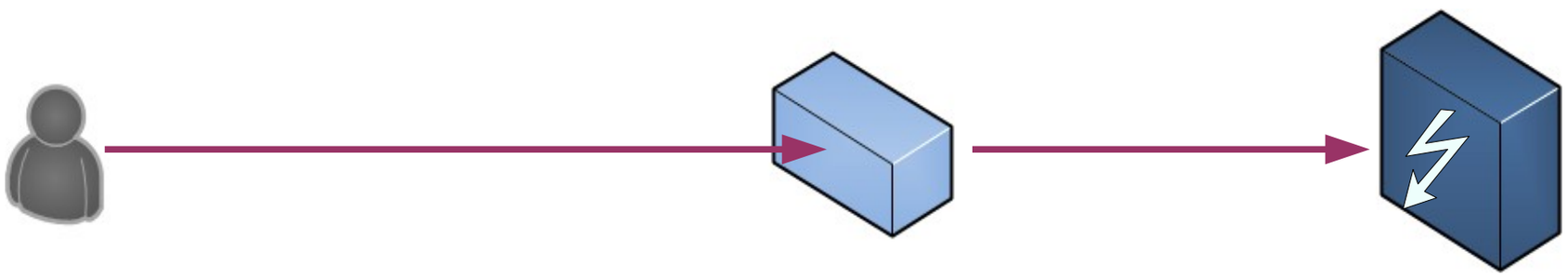
3) Payment transfer exceeding limit

Send 27 pounds,
then send rather more

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

☐ Protected ☒ Unprotected

Deep packet inspection



1) Stepping through a process in the incorrect order

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

☐ 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

Proper attack detection

- Integrated
  - Understands the application
  - Understands normal vs. suspicious use
  - Updated when the business process changes
- Effective
  - Minimal false positives
  - Immediate response
- Scalable
  - 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

Some things your application may already do

- Blocking certain HTTP verbs
- 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 when they use the browser's “back” button
- Terminating a session if a user's geo-location changes
- Blocking access by certain IP addresses when malicious behaviour is detected
- Disable non-core function
- Recording unexpected actions
- Application logging

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

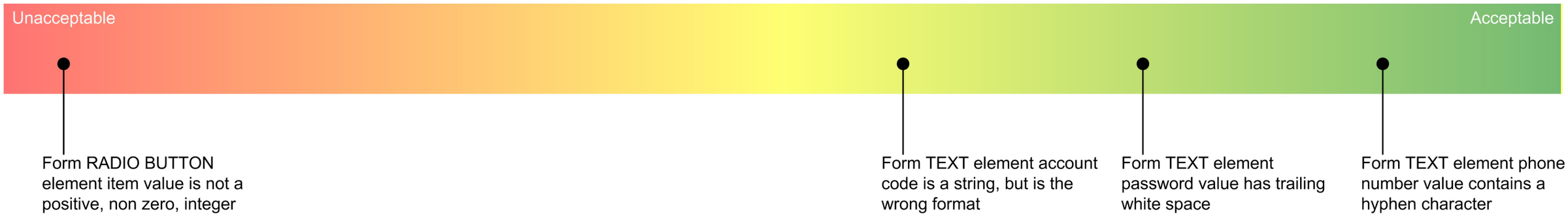
Detecting Malicious Users

- "Users" are not perfect

- Application-specific actions

Importance of Context

- Server-side validation only



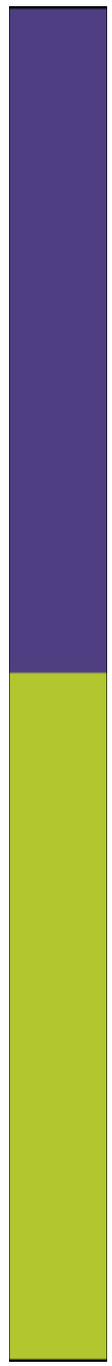
- Server-side with duplicate client-side validation

Unknown attacks

- [This list is intentionally left blank]

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

Further Explanations and Detailed Documentation

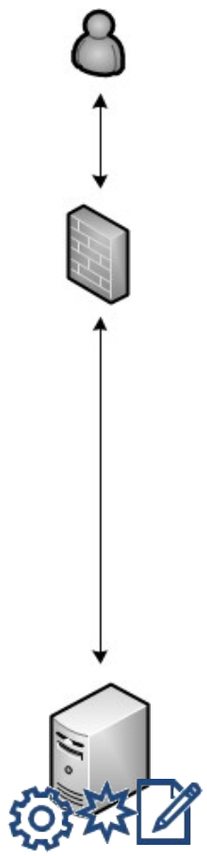
- Video presentations by Michael Coates, AppSensor Project Leader:
  - Automated Application Defenses to Thwart Advanced Attackers, June 2010
    http://michael-coates.blogspot.com/2010/06/online-presentation-thursday-automated.html
  - Attack Aware Applications, April 2011
    https://www.owasp.org/index.php/Minneapolis_St_Paul#tab=Video.2FAudio.2FSlides.2FHandouts
- 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

Implementation

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

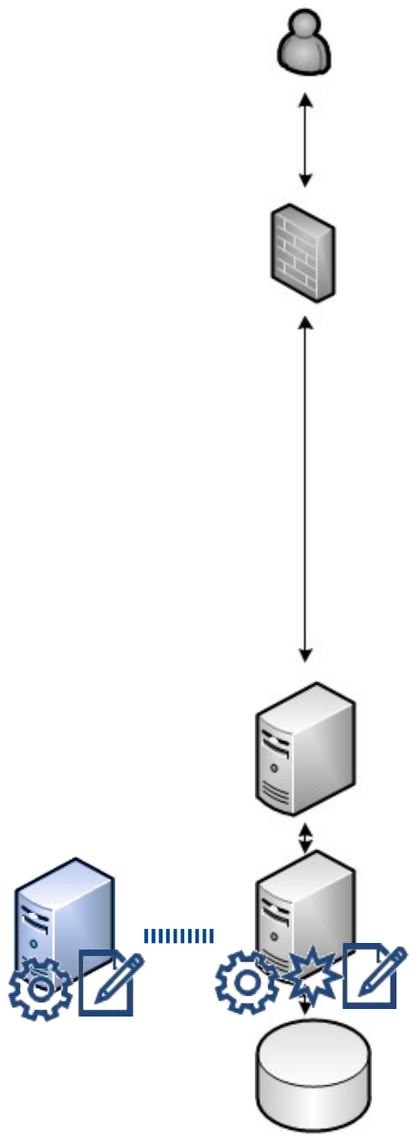
Architectures

- Logging
- Detection
- Analysis



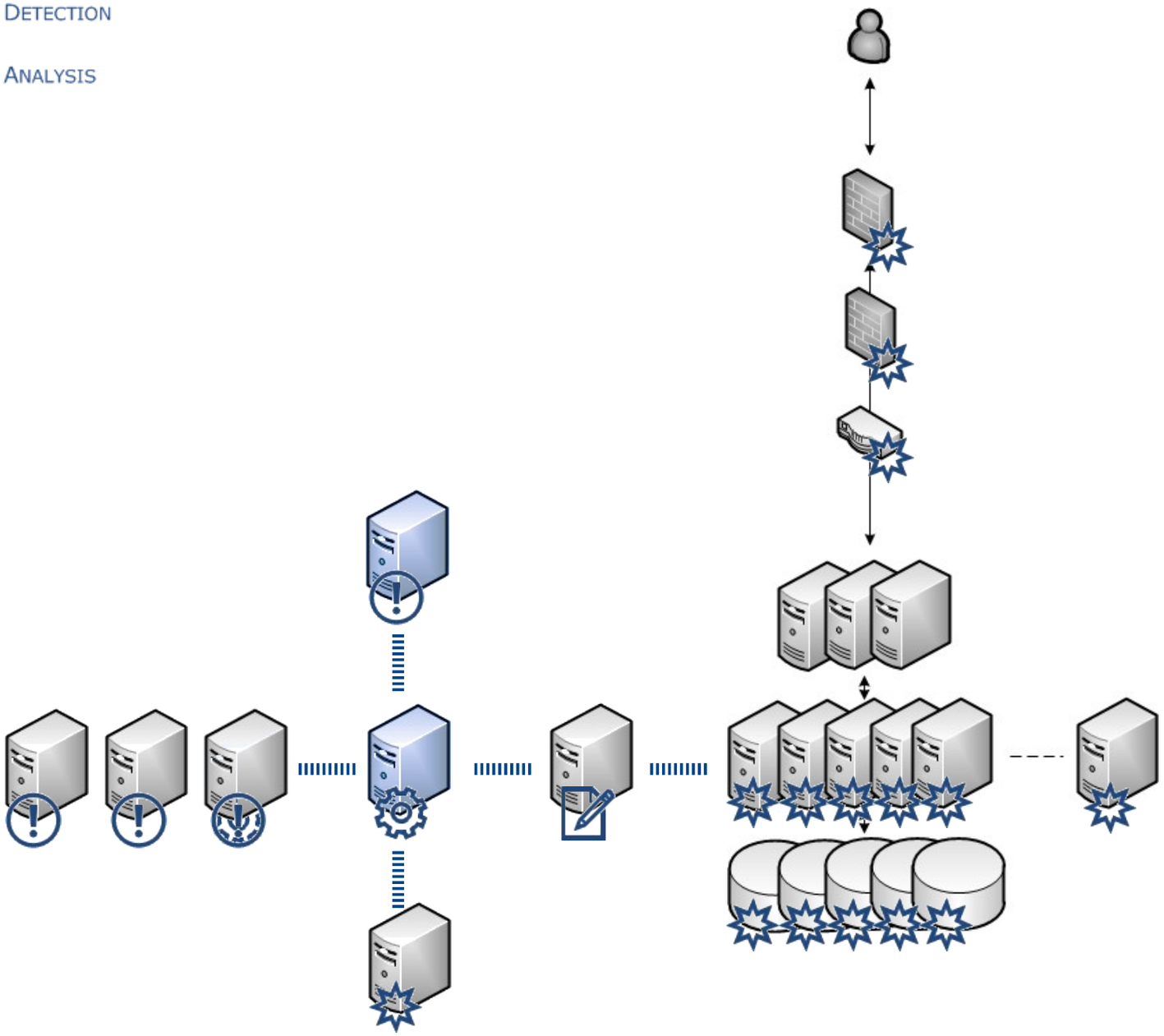
Architectures (continued)

- Logging
- Detection
- Analysis



Architectures (continued)

- Logging
- Detection
- Analysis



Application Logging Inspiration

- See:
  - How to Do Application Logging Right, Anton Chuvakin and Gunnar Peterson, IEEE Security & Privacy Journal
    http://arctecgroup.net/pdf/howtoapplogging.pdf
  - OWASP ESAPI Logger (Java), OWASP
    http://owasp-esapi-java.googlecode.com/svn/trunk_doc/latest/org/owasp/esapi/Logger.html
- See also:
  - SP 800-92 Guide to Computer Security Log Management, NIST
    http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf
  - OWASP Logging Project, OWASP
    https://www.owasp.org/index.php/Category:OWASP_Logging_Project#tab=Main
- Some commentary:
  - Application Security Logging, (own blog)
    http://www.clerkendweller.com/2010/8/17/Application-Security-Logging
  - AppSensor Project Mailing List, OWASP
    https://lists.owasp.org/pipermail/owasp-appsensor-project/2011-March/000215.html

Application Event Logging Aspiration

When
Event date/time
Log date/time

Security Event
Type
Severity
Confidence
Custom classifications
Owner

Location
Host
Service/application name
Port
Protocol
HTTP method
Entry point
Request number

Request
Purpose
Target

AppSensor Detection
Sensor ID
Sensor location
AppSensor ID(s)
Description

Result
Status
Reason for status
HTTP status code
AppSensor Result ID(s)
Description
Message

Who/what
Source
User identity
HTTP User Agent
Client fingerprint

Extra?
Request headers
Request body
Response headers
Response body
Error stack trace
Error message

Record integrity
Identity
Hash

AppSensor Signalling

- Standards
    - Common Event Format (CEF)
    - Common Event Expression (CEE)
- Custom
    - Devices elsewhere on the network
        - Firewalls
        - Web application firewalls
        - Traffic management
    - Other business systems
        - Management reporting
        - CRM
        - Correlation engines (e.g. fraud management)
    - Broadcasting
    - Third parties

Common Event Format

- Prefix
  - Timestamp Host Message
  - June 10 16:48:53 appserver02 *Message*
- Message
  - CEF:Version|Device Vendor|Device Product|Device Version|Signature ID|Name|Severity|*Extension*
  - CEF:0|widgetco|shoponline|3.7.03|R03|XSS attempt blocked|7|*Extension*
- Extension
  - Collection key-value pairs
    - Predefined keys
    - Device custom strings and numbers (x6)
    - Custom dictionary extensions

Common Event Format (continued)

- src=10.25.102.65
- suser=W0005

- proto=TCP
- dpt=80
- dproc=httpd
- request=/catalogue/showProduct/
- requestMethod=GET

- deviceExternalID=AppSensor06
- msg=Cross site scripting attempt in parameter prodid
- cat=detection
- act=block
- cs1Label=requestClientApplication cs1=Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-GB; rv:1.9.2.17) Gecko/20110420
- cs2Label=AppSensorSensorID cs2=R03
- cs3Label=AppSensorDetectionType cs3=IE1
- cs4Label=StatusCode cs4=403
- cn1Label=RequestID cn1=000070825566
- cn2Label=AppSensorLogID cn2=1650833
- cn3Label=Confidence cn3=100

Common Event Format (continued)

1. Auth Failed Event

<165>Jun 08 20:47:29 someapp.mozilla.com CEF:0|mozilla|someapp|1.3|AuthFail|User Authentication Failed|5| cs1Label=requestClientApplication cs1=Mozilla/5.0 (Windows; U; Windows NT 5.1; id; rv:1.9.2.17) Gecko/20110420 FireDownload/2.0.1 Firefox/3.6.17 96690903 Service 2.02155 requestMethod=GET request=https://someapp.mozilla.com/1.0/someuser/info/collections src=1.2.3.4 dst=2.3.4.5 suser=joeuser

2. Invalid Channel Event (custom event)

<166>Jun 08 20:48:42 someapp.mozilla.com CEF:0|mozilla|someapp|1.3|Invalid X-KeyExchange-Channel|Invalid X-KeyExchange-Channel|5|cs1Label=requestClientApplication cs1=Mozilla/5.0 (Windows NT 6.1; rv:2.0b9) Gecko/20100101 Firefox/4.0b9 requestMethod=GET request=/4xjq src=1.2.3.4 dest=someapp.mozilla.com suser=joeuser

3. Username does not match URL (custom event)

<165>Jun 08 20:50:16 someapp.mozilla.com CEF:0|mozilla| someapp |1.3|AuthFail|Username Does Not Match URL|7| cs1Label=requestClientApplication cs1=Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.17) Gecko/20110420 Firefox/3.6.17 (.NET CLR 3.5.30729; .NET4.0C) requestMethod=GET request=https://someapp.mozilla.com/1.0/bobuser/info/collections src=1.2.3.4 dst=2.3.4.5 cs2Label=url_user cs2=joeuser suser=joeuser

4. Password Changed (System trend)

<166>Jun 08 20:52:08 someapp.mozilla.com CEF:0|mozilla|someapp|1.3|PasswordReset|Password Changed|5| cs1Label=requestClientApplication cs1=Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101 Firefox/4.0.1 requestMethod=POST request=/forgot src=1.2.3.4 dest=someapp.mozilla.com suser=joeuser

Application Event Logging Aspiration

When
Event date/time
Log date/time

Security Event
Type
Severity
Confidence
Custom classifications
Owner

Location
Host
Service/application name
Port
Protocol
HTTP method
Entry point
Request number

Request
Purpose
Target

AppSensor Detection
Sensor ID
Sensor location
AppSensor ID(s)
Description

Result
Status
Reason for status
HTTP status code
AppSensor Result ID(s)
Description
Message

Who/what
Source
User identity
HTTP User Agent
Client fingerprint

Extra?
Request headers
Request body
Response headers
Response body
Error stack trace
Error message

Record integrity
Identity
Hash

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

No 2 - Ecommerce Website Advanced Configuration

- Additional requirements
  - Greater granularity of input validation issues
  - Shopping basket and order processing session checks
  - User and system trends
  - Integration with reputation monitoring
- Additional AppSensor detection points
  - Valid parameter names and application entry points
  - Integrity checks on user submitted data
  - User trend for orders completed
  - System trends for site utilisation, and catalogue/basket/payment usage
  - Third party malware monitoring feed
  - Intrusion Protection System feed

No 2 - Detection Points

Area	Identifier	#	AppSensor ID(s)	Notes
Request	R04	R	RE5, RE6	Extra/duplicated/missing input parameter
	R05	R	ACE3	Invalid dynamic entry point (force browsing)
Catalogue	C02	+	IE2	Input validation white list exception
	C03	+	ACE1, ACE2	Parameter manipulation for direct object access
	C04		HT2	“Magic” product accessed
Basket	B02	+	IE2	Input validation white list exception
	B03	S	SE1	Shopping basket cookie altered
	B04	S	SE4	Shopping basket cookie substitution
Payment	P03	+	IE2	Input validation white list exception
	P04		IE4	Input data integrity exception
	P05	S	SE4	Payment cookie substitution
External	E01		RP4	Malware identified in site content by remote system
	E02		RP2	Network Intrusion Protection System (IPS) alert
User Trend	U01		UT4	High rate of order placement
System Trend	S01		STE3	High or Low rate of general page impressions
	S02		STE3	High or Low rate of catalogue page impressions
	S03		STE3	High or Low rate of shopping baskets creation
	S04		STE3	High rate of shopping basket deletion
	S05		STE3	High rate of missing file (404 not found) errors

No 2 – Response Actions

- Overall detection point threshold set with a disruptive action
- Business layer input validation exceptions:
    - High thresholds when user data entry allowed
    - Low thresholds and disruptive response actions for clearly malicious behaviour
- Strict limits on access control exceptions
- Reputational information used to help identify site malware infection for early response
- Correlation with IPS information to block users also undertaking malicious behaviour on the network
- User trend information used to change credit rating
- System trend information used for:
    - Detection of phishing attacks and application work activity
    - Advance warning of problems such as resource exhaustion, warehouse and stock utilisation
- Never block privileged accounts, but alert and log vigorously

Dashboard demonstration

- Live (during presentation) demos for Ecommerce website
  - Base configuration
  - Advanced configuration
- Video (no sound/narration) of these demos available at:
  - Base configuration
    http://www.youtube.com/watch?v=zCaYREAyiRg
  - Advanced configuration
    http://www.youtube.com/watch?v=YZ5zGQ-XLkk

Two question revisited

1) Is the application being attacked now?

2) Have any unknown vulnerabilities been exploited today?

☐ Yes ☐ No ~~☐ Don't know~~

Make contact

Colin Watson

- colin.watson(at)owasp.org

AppSensor Project

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

Full-day training at AppSec USA

- Application Attack Detection & Response - A Hands-on Planning Workshop
  http://www.appsecusa.org/training.html#watson