

# OWASP AppSensor

## The Future of Application Security

Dennis Groves, MSc  
dennis.groves@owasp.org

November 18, 2013

# About Me



# A Thought Experiment





## THE BOTTOM LINE ON OPSEC;

We all have information that the Bad Guys need to hurt us. We don't want them to get it. The OPSEC process helps us to look at our world through the eyes of an adversary and to develop measures in order to deny them. Get it?



The Interagency  
OPSEC Support Staff  
[www.ioos.gov](http://www.ioos.gov)

### The OPSEC Process:

- ① Identify Critical Info
- ② Analyze Threats
- ③ Analyze Vulnerabilities
- ④ Assess the Risks
- ⑤ Apply Countermeasures

THINK ABOUT IT... ALL THE TIME!

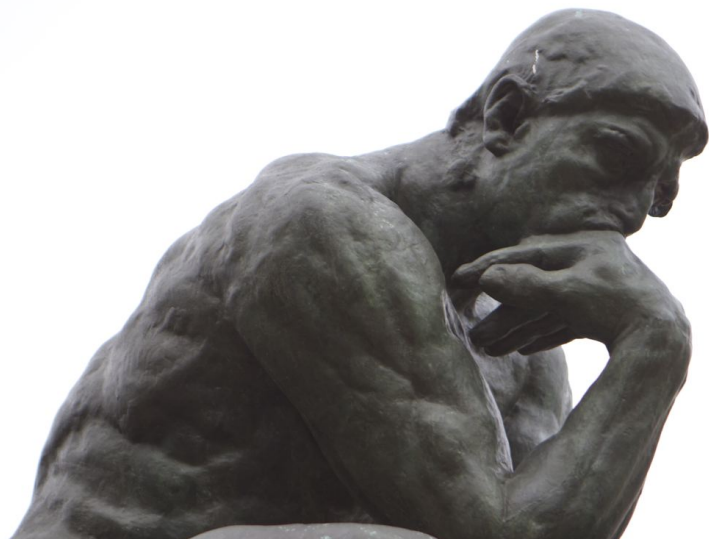


### 5 STEPS... 1 MINDSET

**WHAT IS OPERATIONS SECURITY?**  
Operations Security, or OPSEC, is a risk management methodology used to deny an adversary information concerning our intentions and capabilities by identifying, controlling, and protecting critical information associated with the planning and execution of a mission.



# 1: Identify Critical Info



# Victorian Duel





## Theorem

Protection time must be greater than or equal to detection time plus reaction time.

$$(1) \quad P_t \geq D_t + R_t$$

# Wester Duel







## Pistol Duel

- ▶ Novice Shooter
- ▶ Weekend Shooter
- ▶ Professional Shooter
- ▶ Quick Draw Champion

## Application

- ▶ Script Kiddies
- ▶ Hacktivists
- ▶ Criminals
- ▶ Disgruntled Employee
- ▶ Corporate Spy
- ▶ Cyber Warrior

# Out Gunned



## 2: Analyze Threats



### Pistol Duel

- ▶ Handgun Skills
- ▶ Nervousness
- ▶ Psychological Readiness

### Application

- ▶ Spoofing
- ▶ Tampering
- ▶ Repudiation
- ▶ Information Disclosure
- ▶ Denial of Service
- ▶ Elevation of Privilege

# 3: Analyze Vulnerabilities



## Pistol Duel

- ▶ Jam
- ▶ Misfire
- ▶ Backfire

## The OWASP Top-10

- ▶ A1 Injection
- ▶ A3 Cross-Site Scripting
- ▶ A5 Security Misconfiguration
- ▶ A7 Missing Access Control

## 4: Analyze Risks



The probable frequency and probable magnitude of future loss

$$(2) \quad \text{Risk} = P(\text{Impact})$$

$$(3) \quad \text{Risk} = P(\text{Impact} * \text{Vulnerability})$$

$$(4) \quad \text{Risk} = \text{Impact} * \text{Vulnerability} * \text{Threat}$$

$$(5) \quad \text{Risk} = P(\text{Impact} * \text{Vulnerability} * \text{Threat})$$

$$(6) \quad \text{Risk} = \frac{\text{Impact} * \text{Vulnerability} * \text{Threat}}{\text{Countermeasures}}$$

$$(7) \quad \text{Risk} = \text{Impact} * \frac{P(\text{Threat}) * P(\text{Vulnerability})}{\text{Countermeasures}}$$

## 5: Apply Countermeasures



- ▶ Tolerate: Do nothing.
- ▶ Transfer: Outsource the risk.
- ▶ Terminate: Eliminate the asset.
- ▶ Treat: Reduce the risk.



Reducing the risk (treatment) is the most common strategy used today.

- ▶ Reduce the probability of a threat.
- ▶ Reduce the probability of a vulnerability.



## Pistol Duel

- ▶ Turn To The Side
- ▶ Crouch Down Low
- ▶ ???

## Application

- ▶ Penetration Testing
- ▶ Code Review
- ▶ Patching






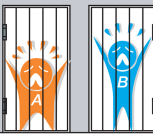


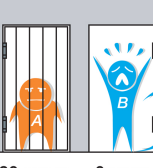

# Predicting the Future

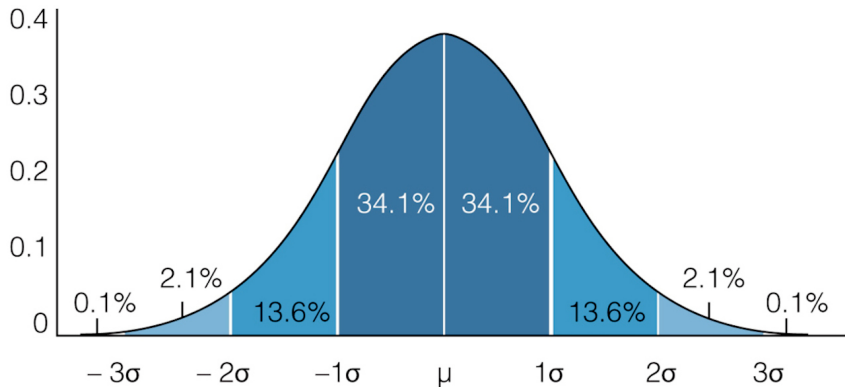


# Game Theory

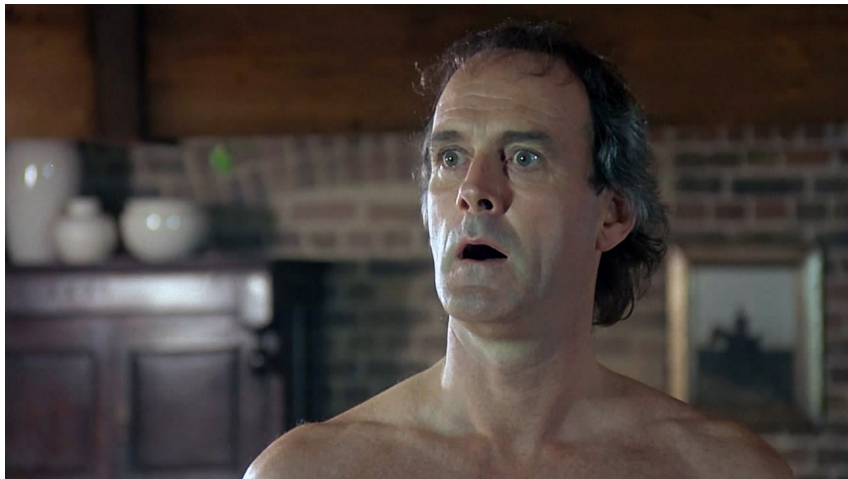


Prisoners'  
dilemma

		prisoner B	
		confess 	remain silent 
prisoner A	confess 	 5 years    5 years	 0 year    20 years
	remain silent 	 20 years    0 year	 1 year    1 year



# Now For Something Completely Different





Risk Optimisation is rarely practiced, but highly effective method.

- ▶ Reduce the impact of an event

# Bullet Proof Vest





- ▶ AppSensor is not a panacea nor is a vest
- ▶ You do not want to get shot, but if you do, you want to be wearing a vest
- ▶ If you get shot while wearing a vest, it is going to hurt, but you will survive







**Michael Coates**, Colin Watson, John Melton Ryan Barnett, Simon Bennetts, Marc Chisinevski, Robert Chonjnacki, August Detlefsen, Sean Fay, Randy Janida, Alex Lauerman, Manuel Arredondo, Bob Maier, Craig Munson, Giri Nambari, Abdul Rauf, Jay Reynolds, Eric Sheridan, John Steven, Alex Thissen, Don Thomas, Kevin Wall, Mehmet Yilmaz, Jim Manico, Dinis Cruz, myself and many, many others...



- ▶ Detection & Reaction in the App
- ▶ Attack-Aware Detection
- ▶ Normal and Malicious Behavior
- ▶ Evasion and Unknown Attacks

# Over 50 Detection Points



Type	Code	Name
Signature	RE	Request Exceptions
	AE	Authentication Exceptions
	SE	Session Exceptions
	ACE	Access Control Exceptions
	IE	Input Exceptions
	EE	Encoding Exceptions
	CIE	Command Injection Exceptions
	FIO	File IO Exceptions
Behavioural	HT	Honey Trap
	UTE	User Trend Exceptions
	STE	System Trend Exceptions
	RP	Reputation



Response Type	Examples
Logging Change	Full stack trace of error messages logged Record DNS data on user's IP address
Account Logout	Session terminated and user redirected Session terminated only (no redirect)
Account Lockout	User account locked permanently One user's IP address range blocked
Application Disabled	Website shut down and replaced with static page Application taken offline



- ▶ AppSensor-core
- ▶ AppSensor-ws-soap
- ▶ AppSensor-ws-rest
- ▶ AppSensor Guide



- ▶ Goal: Produce viable implementation that allows intrusion detection to move towards a functional primitive in any language
- ▶ It should be as simple as possible to detect and respond to events in your environment



- ▶ Existing V1
  - ▶ Java only (requires developers to re-implement full system in any other language)
  - ▶ Built to work with ESAPI (difficult to remove dependency)
  - ▶ Functional, but missing many features
- ▶ It should be as simple as possible to detect and respond to events in your environment

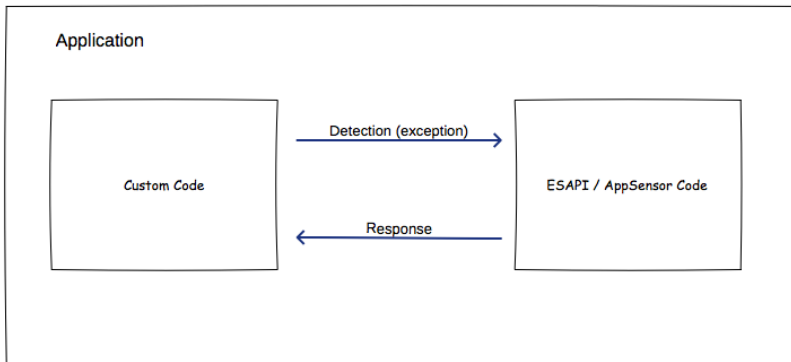


- ▶ New V2 (in progress)
  - ▶ Java core backend
  - ▶ Services (rest/soap) enable front-end in any lang
    - ▶ only re-implement minor portions, significant analysis done 1.me
    - ▶ We can build several reference front-ends (help!)
  - ▶ Basic correlation between applications
  - ▶ Allows input from external systems (WAF, IDS, etc.)
  - ▶ Enables reporting





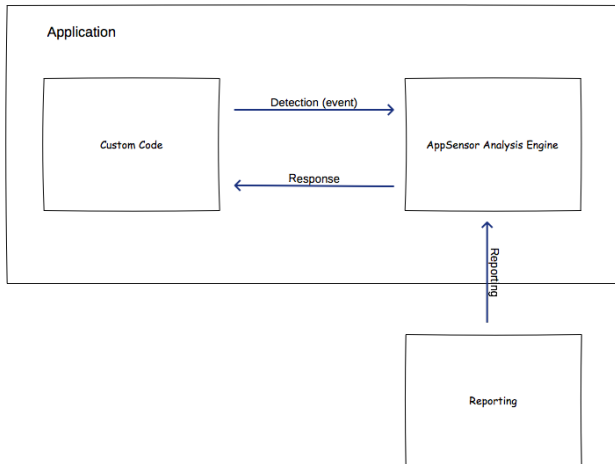
## AppSensor V1 (Java Only)



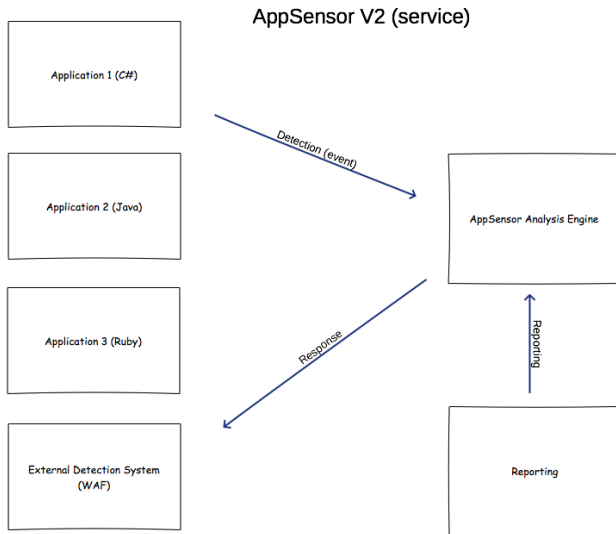
# AppSensor Version 2 Local



## AppSensor V2 (local)



# AppSensor Version 2 Service



# How Can You Help?



- ▶ Join the Mailing List and Participate
- ▶ Help us develop reference implementations
- ▶ Tell your friends, and employers

Obrigado!



# Questions?



# Q&A

You have

# Questions

We have

# Answers

