



# Real Time Application Defenses

## The Reality of AppSensor & ESAPI

**Michael Coates**

Mozilla - Web Security Lead  
mcoates@mozilla.com

<http://michael-coates.blogspot.com>  
@\_mwc

**March 23, 2011**

**The OWASP Foundation**

<http://www.owasp.org>

# Agenda

- Power of Application Intrusion Detection
- ESAPI & AppSensor
- Release of AppSensor-Tutorial
- AppSensor @ Mozilla

# AppSensor Team

## AppSensor Core Team

Michael Coates

John Melton

Colin Watson

## Contributors

Ryan Barnett

Simon Bennetts

August Detlefsen

Randy Janida

Jim Manico

Giri Nambari

Eric Sheridan

John Stevens

Kevin Wall

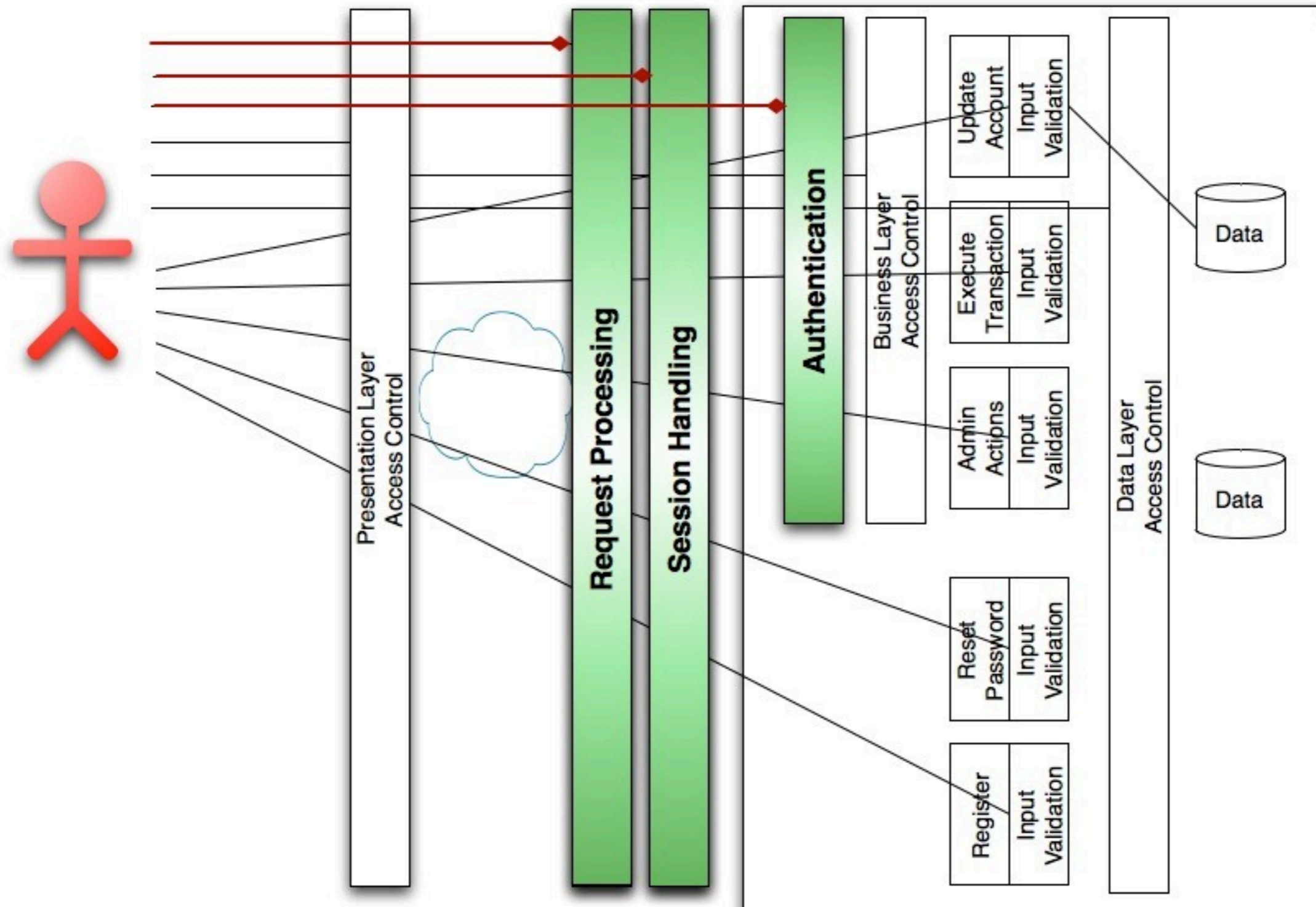
# Power of Application Intrusion Detection

# Status Quo

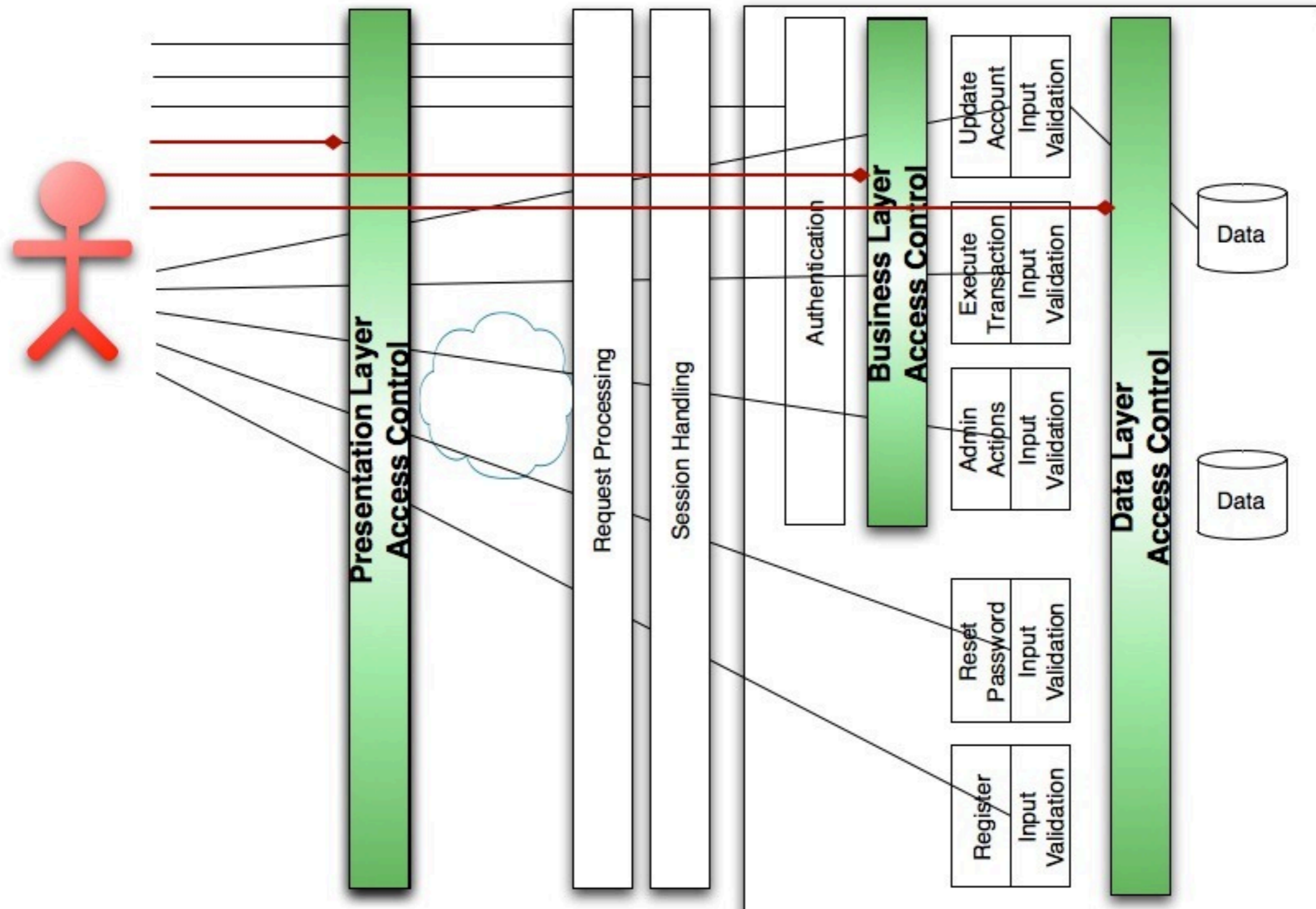
## Defense Capabilities

- Build secure & hope for the best
- Would you know if your application was currently under attack?
- How confident are you against a skilled attacker?
- Is your attack alert system based on watching the NYT for a front page article?

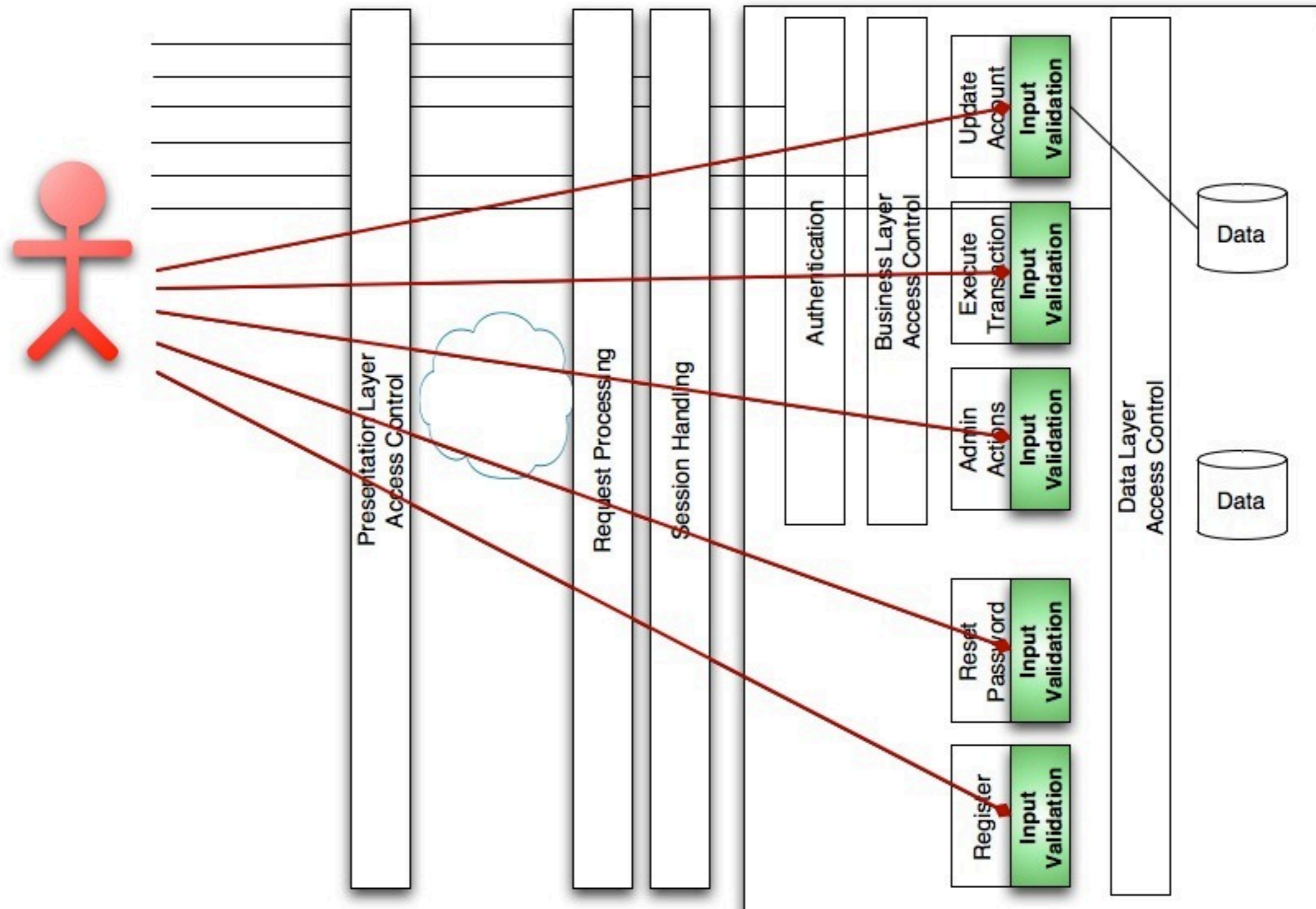
# Attack Points: Requests, Auth, Session



# Attack Points: Access Control

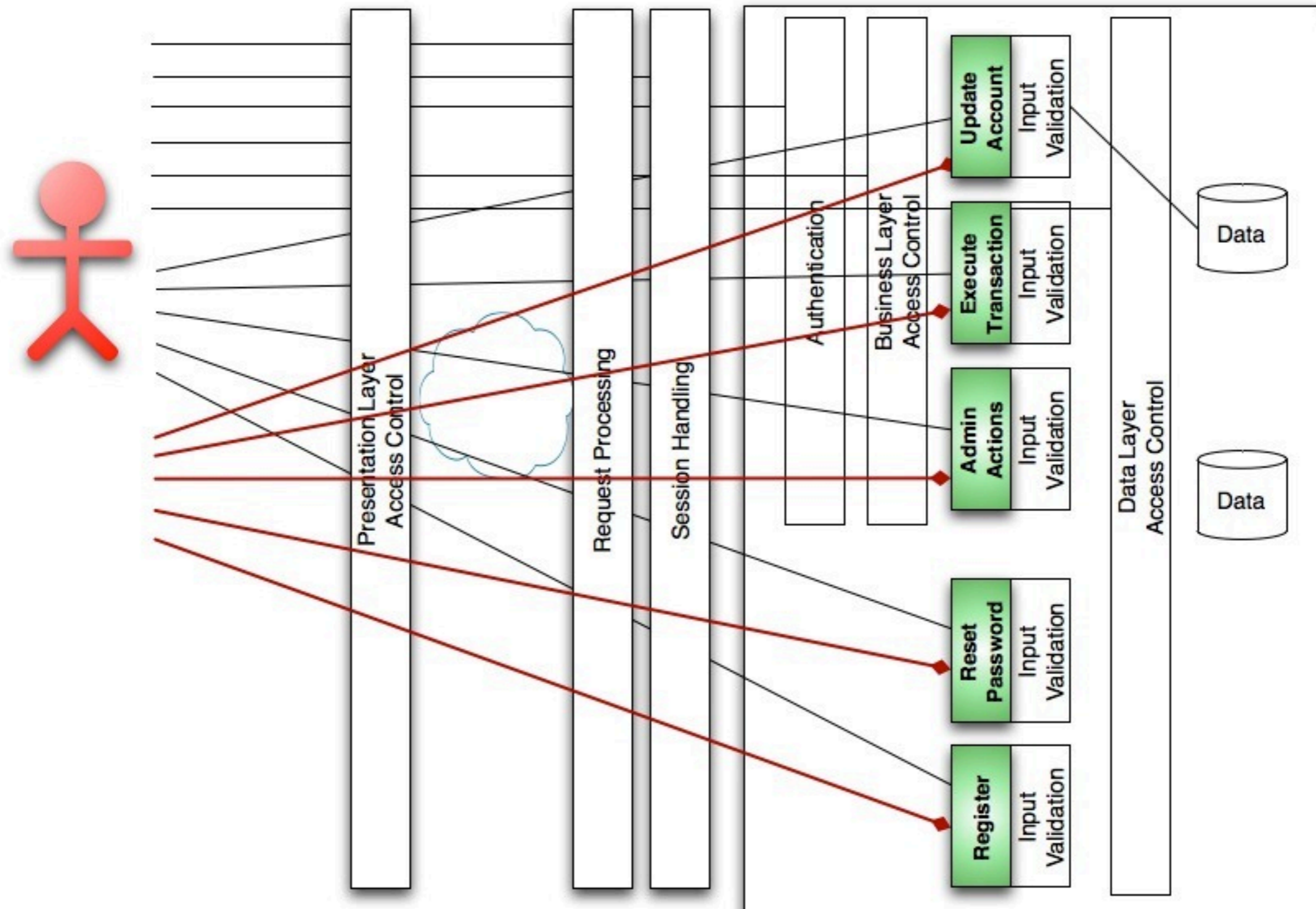


# Attack Points: Input Validation

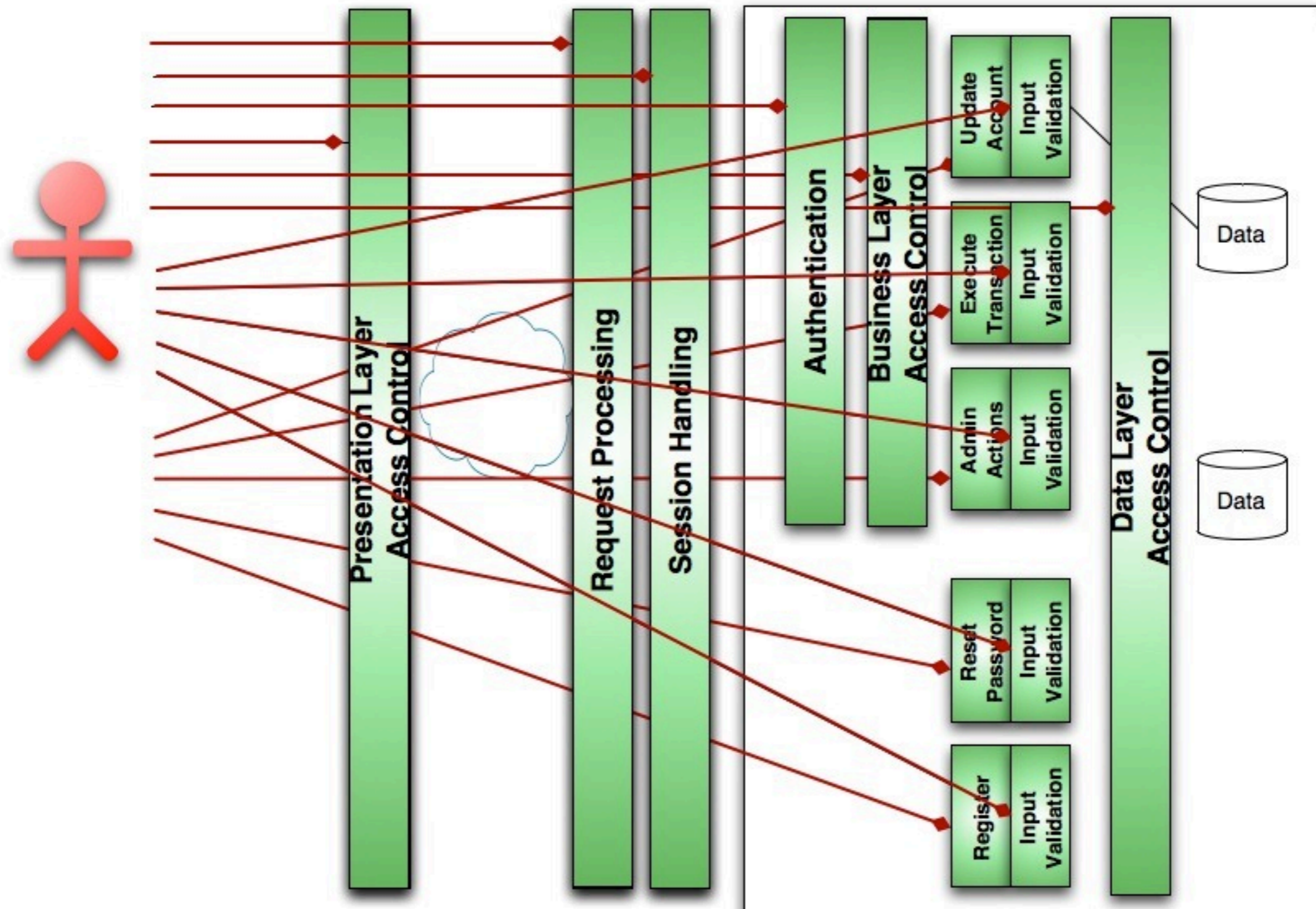




# Attack Points: Business Logic

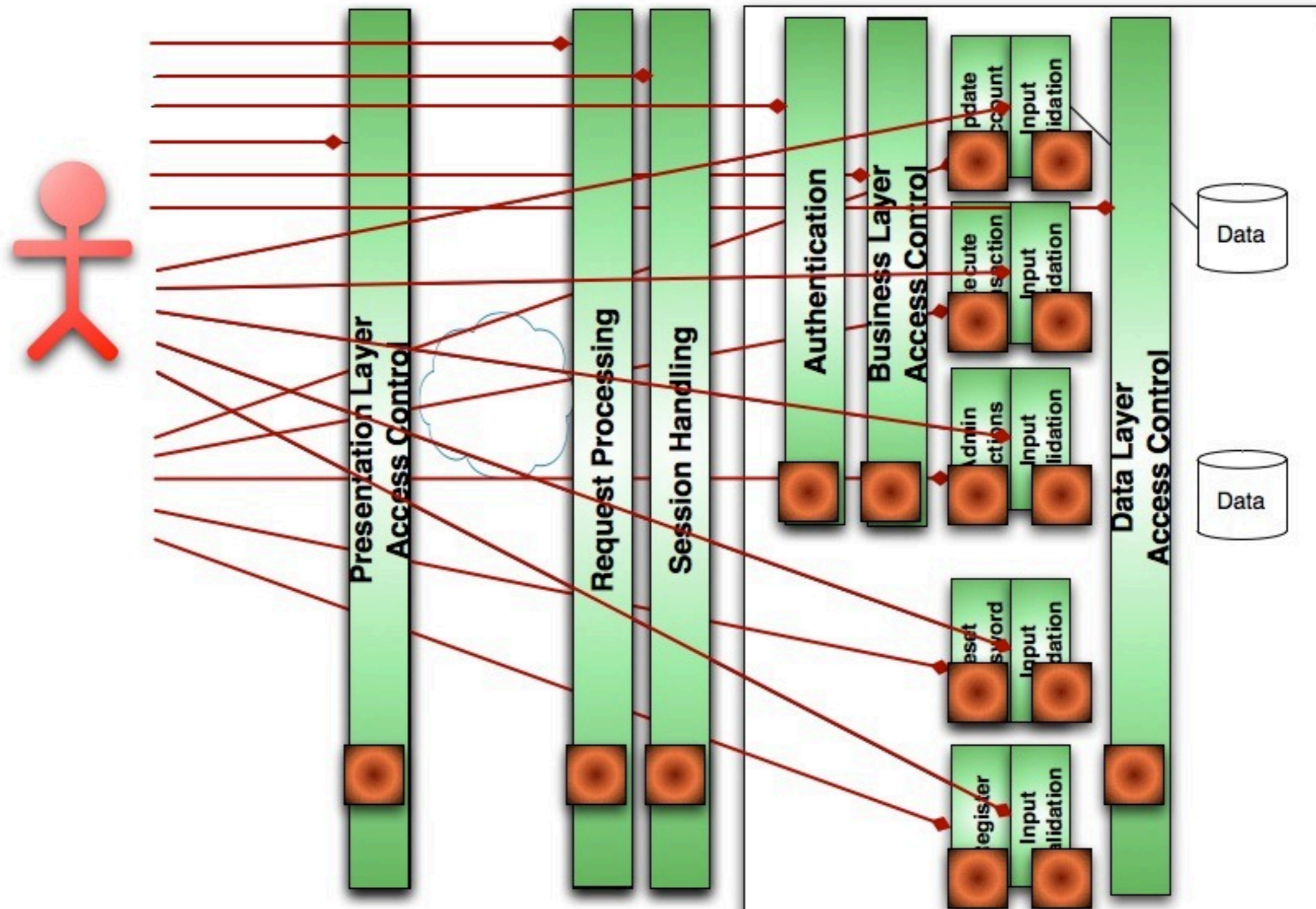


# Numerous Attack Points



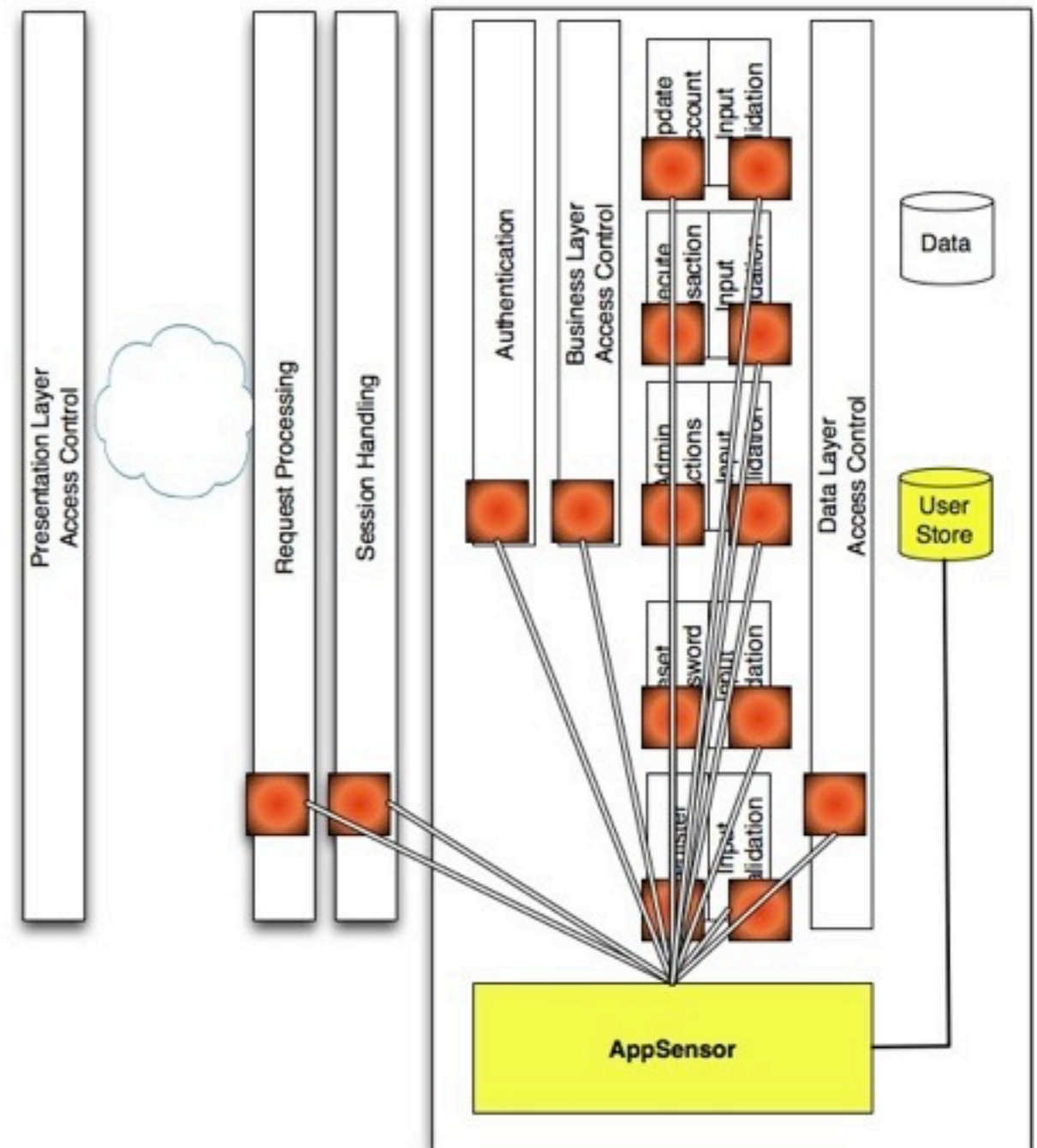


# Defend with: Detection Points



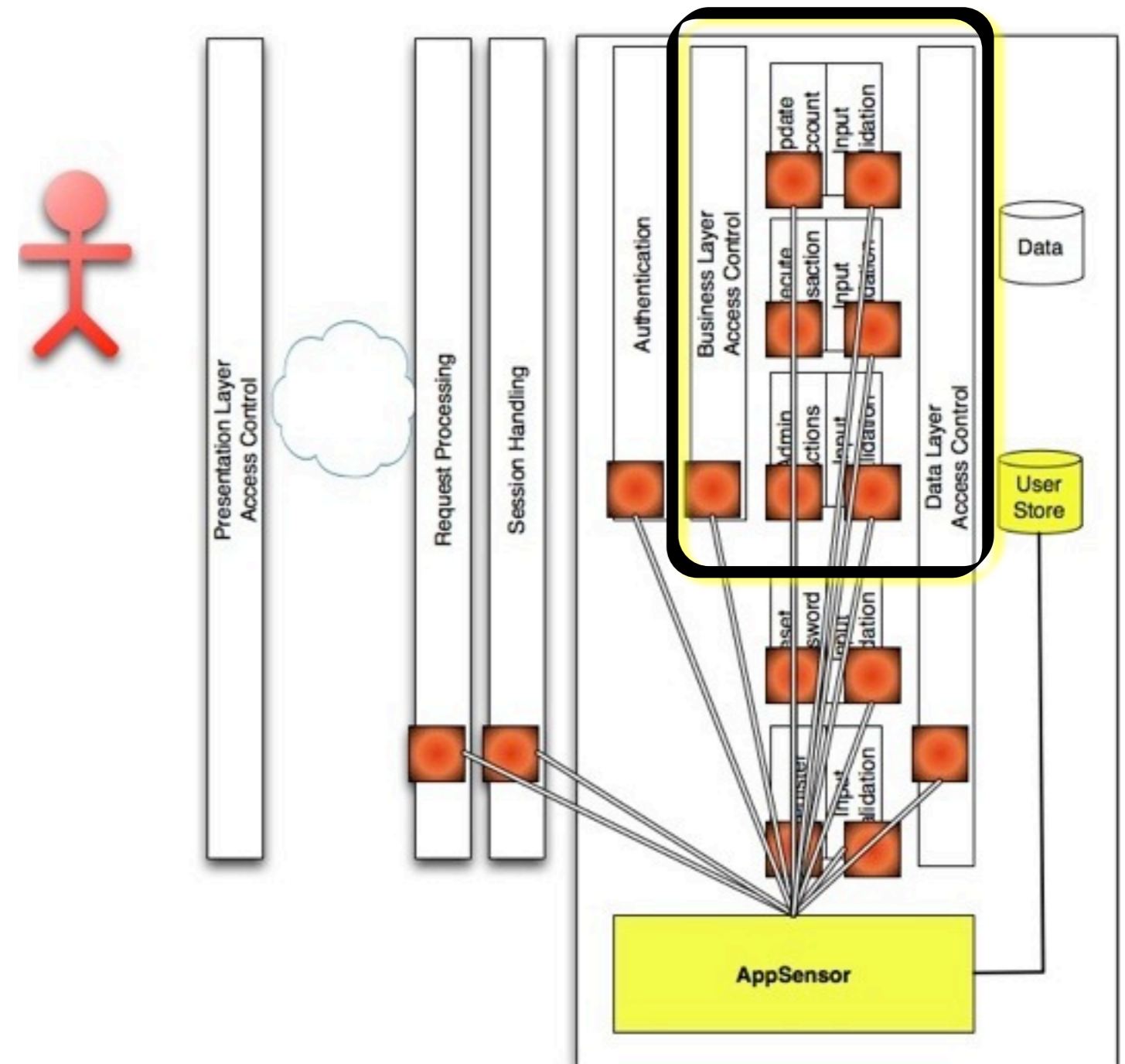
# Defend with: AppSensor Integration

- Detection Points Report to AppSensor
- AppSensor Integrates w/User Store
- Enables Response Actions against User Object



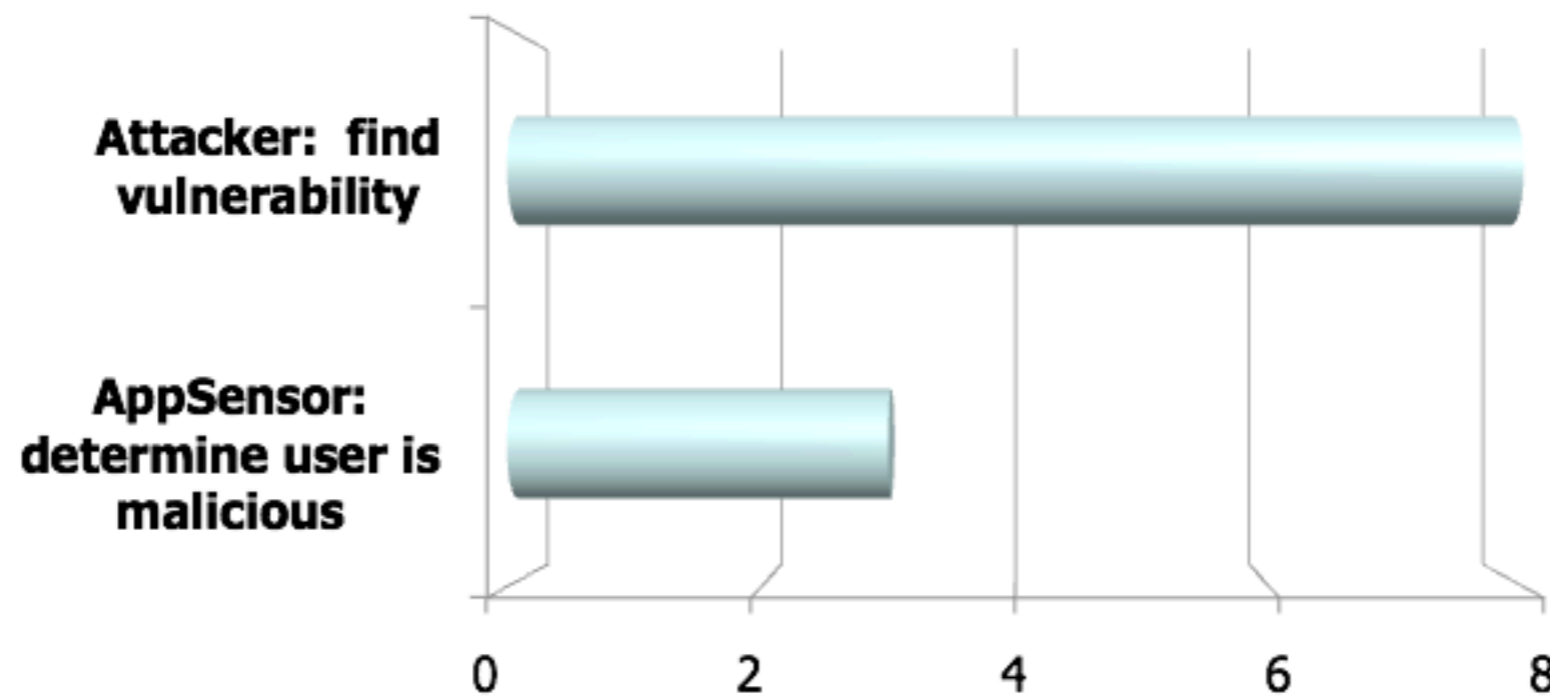
# Detect & Eliminate Threat

- Strong control of authenticated portion
- Lockout user
- Disable account
- Effective attack reporting for unauthenticated portion



# AppSensor Eliminates Threats

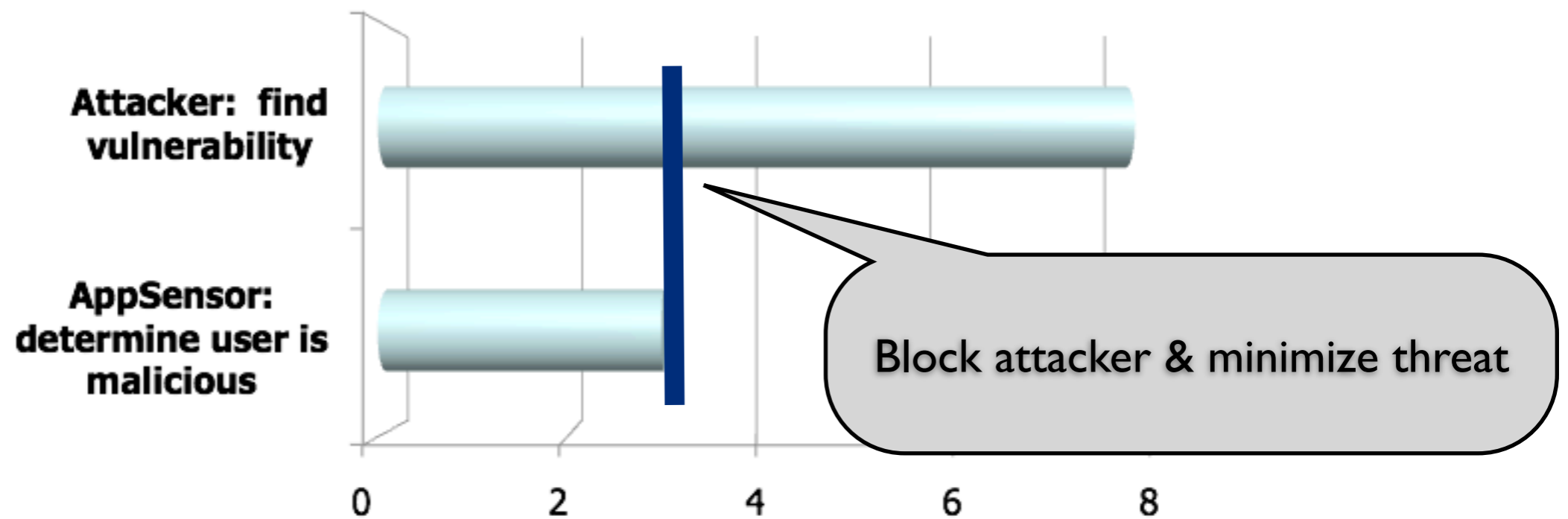
**Requests Needed for Attacker vs. AppSensor**



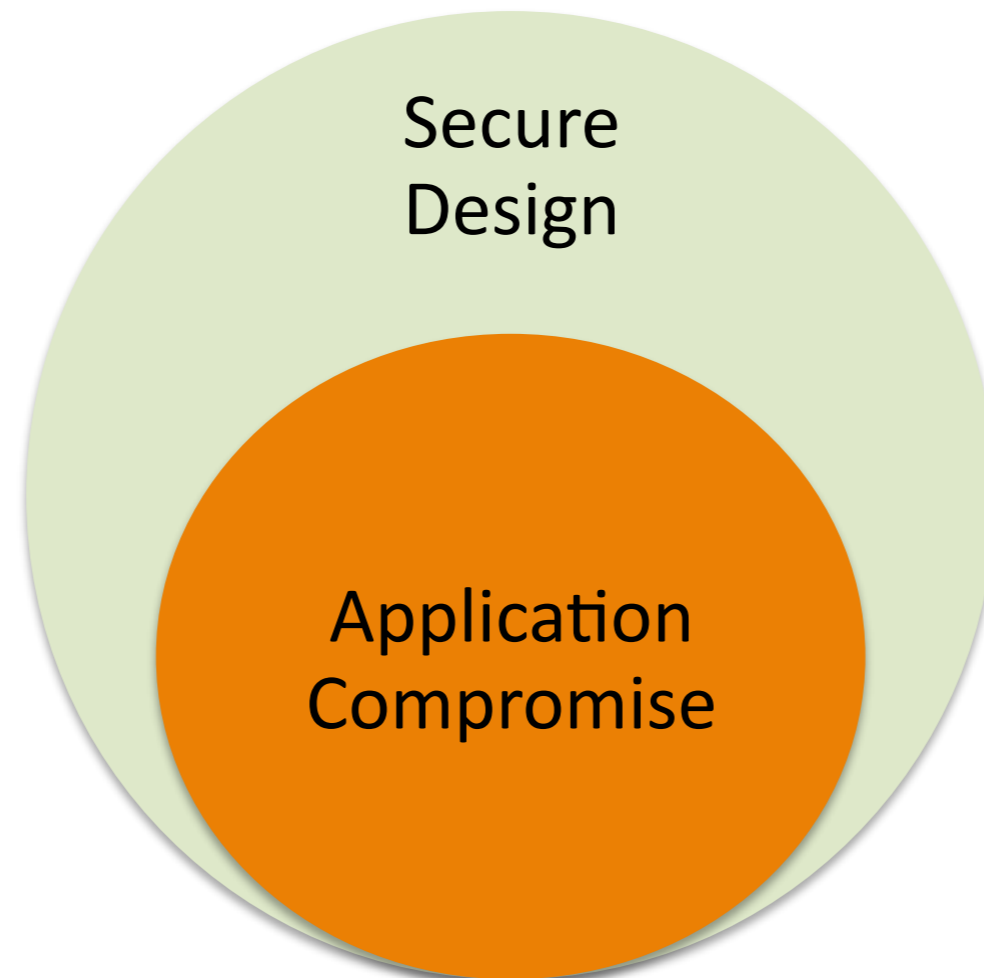


# AppSensor Eliminates Threats

**Requests Needed for Attacker vs. AppSensor**



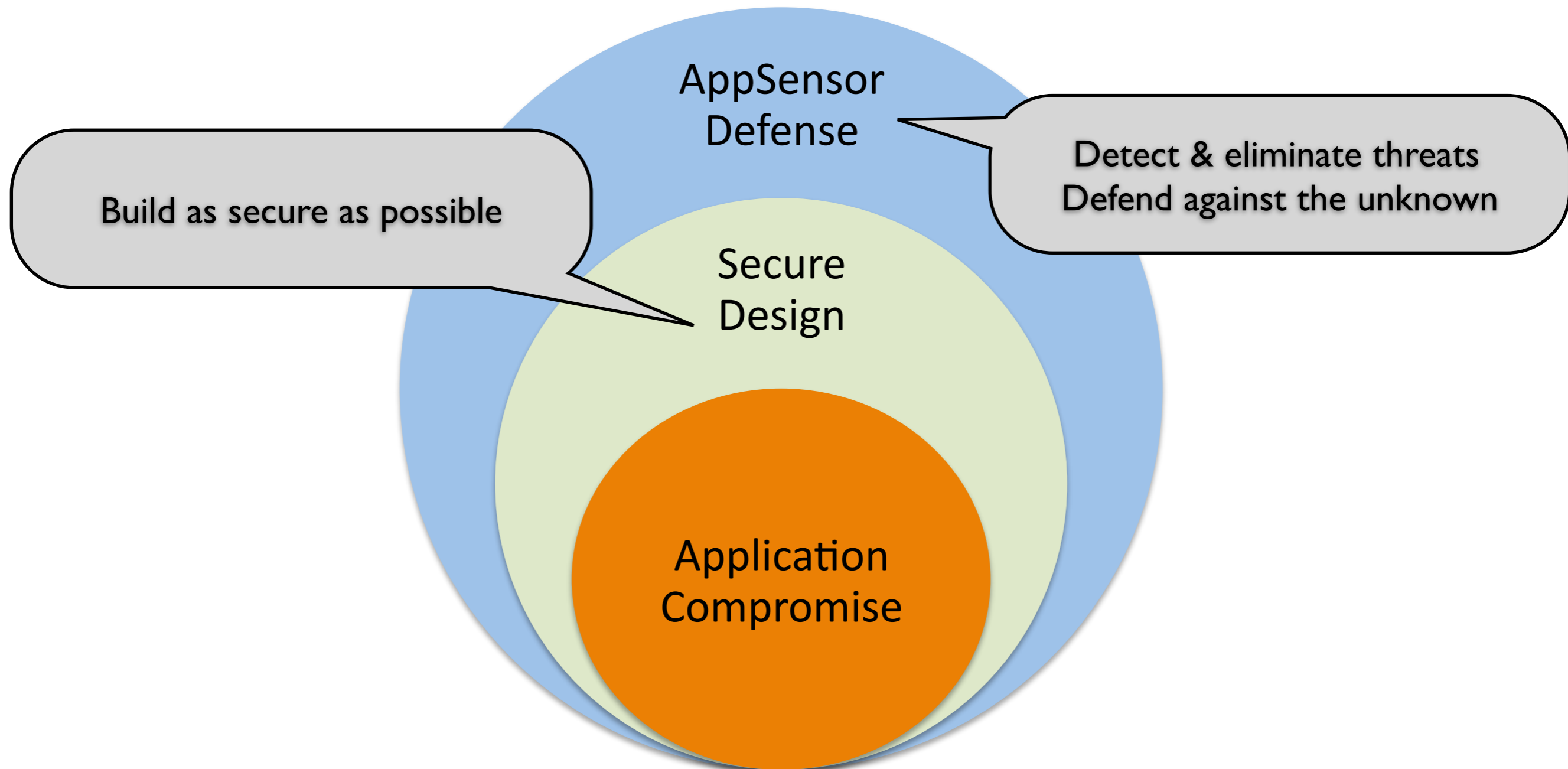
# Current Approach



**Build secure & hope for the best**



# AppSensor Approach



**Add layer of attack detection & prevention**

# Enhancing App Security

## Build Secure

Integrate Security into SDLC

Security Code Review &  
Penetration Testing



## Actively Defend

Attack Detection  
Points

Application Trend  
Anomaly Detection

Automated Response  
to Quarantine  
Attackers

# Why This Approach?

- AppSensor - in the app, full user object interaction, full app knowledge
- WAF - generic attack detection
- Log Analysis - slow, reactive, ineffective

# ESAPI & AppSensor

# Integration Status

- **appsensor.jar ready to use w/ESAPI**
- **AppSensor developer guide available**  
[http://www.owasp.org/index.php/AppSensor\\_Developer\\_Guide](http://www.owasp.org/index.php/AppSensor_Developer_Guide)
- **AppSensor + ESAPI bundle planned for ESAPI 2.0 rc8**

# ESAPI / AppSensor Adoption

- **AppSensor**
  - ModSecurity
  - Major Insurance Company - AppSensor standard for all new web apps
  - Mozilla - AppSensor detection integrated into web apps
- **ESAPI** - American Express, Apache Foundation, Booz Allen Hamilton, Aspect Security, Foundstone(McAfee), The Hartford, Infinite Campus, Lockheed Martin, MITRE, Nationwide Insurance, U.S. Navy - SPAWAR, The World Bank, SANS Institute

[http://www.owasp.org/index.php/Category:OWASP\\_Enterprise\\_Security\\_API#tab=Home](http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API#tab=Home)

# AppSensor.jar

- Drop-in support for ESAPI
- 3 Line configuration in ESAPI.properties
- Define policies in appsensor.properties
- Add detection points in code (2-3 lines each)
- Done!

# How Easy To Setup?

```
ESAPI.IntrusionDetector=org.owasp.appsensor.intrusiondetection.AppSensorIntrusionDetector
```

**ESAPI.properties**

```
IntrusionDetector.X1.count=2  
IntrusionDetector.X1.interval=35  
IntrusionDetector.X1.actions=log,logout,disableComponentForUser  
IntrusionDetector.X1.disableComponentForUser.duration=30  
IntrusionDetector.X1.disableComponentForUser.timeScale=m
```

**appsensor.properties**

```
if (AttackDetected){  
    new AppSensorException( "X1","User Error Message",  
        "Logged Error Message" + "("+ request.getRequestURI()+ ")"  
        + " user (" + ESAPI.authenticator().getCurrentUser().getAccountName() + ")");  
}
```

**code**



# Detecting Attacks

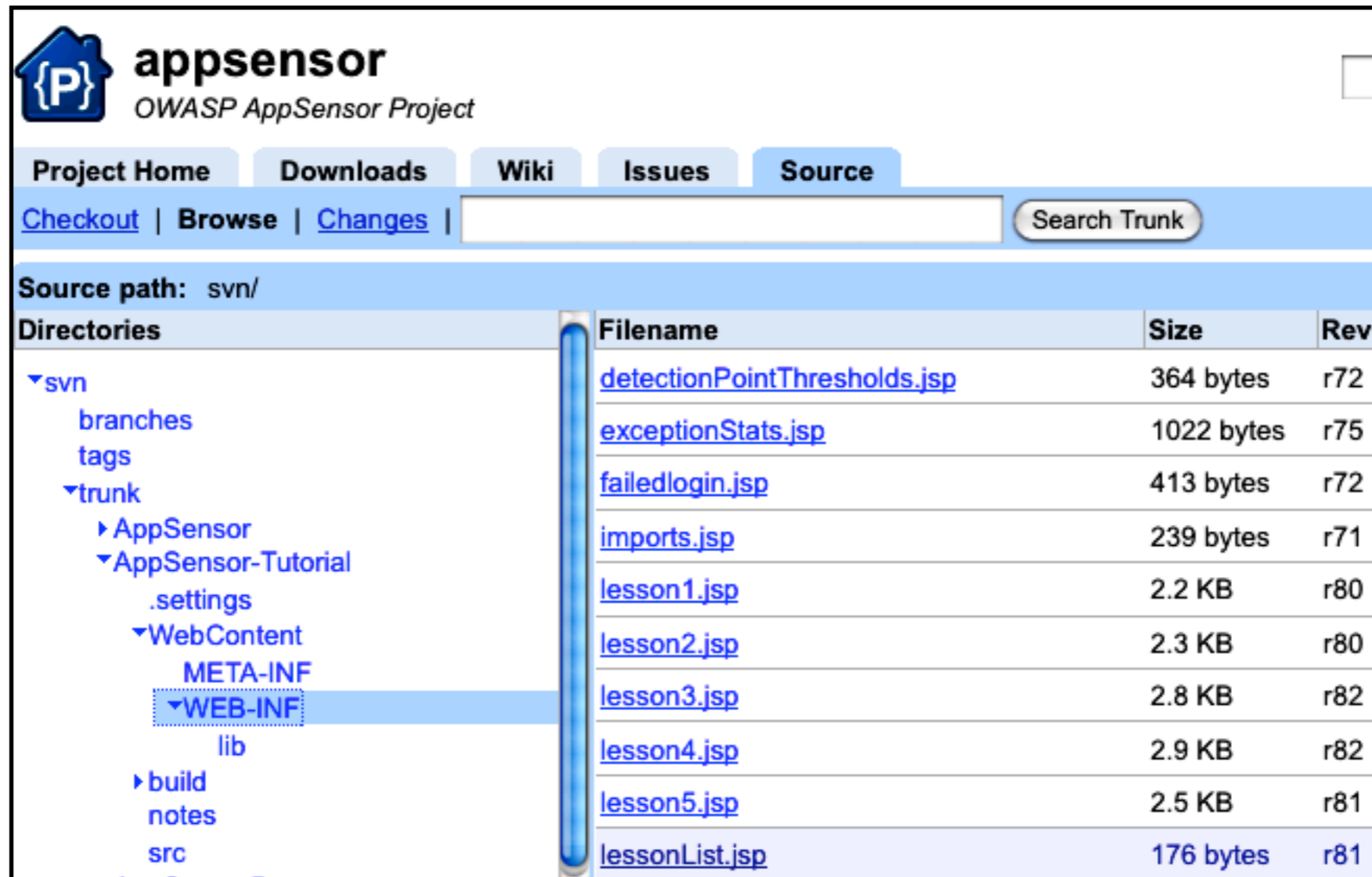
- 50+ attack detection points and growing
- Grouped into logical areas
  - Request, Auth, Input, Access etc
- Most have nearly zero false positive rate
  - POST When Expecting GET
  - Evading Presentation Access Control Through Custom POST
  - Attempt to Invoke Unsupported HTTP Method

<b>RE1: Unexpected HTTP Command</b>	
ID	RE1
TITLE	Unexpected HTTP Command
CATEGORY	RequestException
DESCRIPTION	An HTTP request is received which contains unexpected/unbalanced commands.
CONSIDERATION	A list of accepted commands should be generated (i.e. GET and POST) and all other HTTP commands should generate an error. Instead of a GET or POST request, the user sends a TRACE request to the application.
EXAMPLES	Cross references: <ul style="list-style-type: none"><li>• OWASP ModSecurity Core Rule Set Project v2.0.0</li><li>• [RE: HTTP Policy Enforcement: Method is Not Allowed by Policy (40000)]</li></ul>
CODE	[Link] [New] [PDF]
<b>RE2: Attempt to Invoke Unsupported HTTP Method</b>	
ID	RE2
TITLE	Attempt to Invoke Unsupported HTTP Method
CATEGORY	RequestException
DESCRIPTION	An HTTP request is received which contains a non-existent HTTP command (does not match anything in this list: HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT).
CONSIDERATION	Instead of a GET or POST request, the user sends a TEST request to the application. (TEST is not a valid HTTP request method).
EXAMPLES	
CODE	[Link] [New] [PDF]
<b>RE3: GET When Expecting POST</b>	
ID	RE3
TITLE	GET When Expecting POST
CATEGORY	RequestException
DESCRIPTION	A page which is expecting only POST requests, is responded by HTTP method GET.
CONSIDERATION	Some pages may be designed to receive both GET and POST requests.
EXAMPLES	The user sends a GET request to a page which has only been used for POSTs.
CODE	[Link] [New] [PDF]
<b>RE4: POST When Expecting GET</b>	
ID	RE4
TITLE	POST When Expecting GET
CATEGORY	RequestException
DESCRIPTION	A page which is expecting only GET requests, receives a POST.
CONSIDERATION	The user utilizes a proxy tool to build a custom POST request and sends it to a page which has been accessed by GET requests.
EXAMPLES	
CODE	[Link] [New] [PDF]
<b>RE5: Additional/Duplicated Data in Request</b>	

[http://www.owasp.org/index.php/AppSensor\\_DetectionPoints](http://www.owasp.org/index.php/AppSensor_DetectionPoints)

# Release of AppSensor-Tutorial

# AppSensor-Tutorial



The screenshot shows the source code browser for the AppSensor project. The page title is "appsensor" with the subtitle "OWASP AppSensor Project". The navigation menu includes "Project Home", "Downloads", "Wiki", "Issues", and "Source". The "Source" tab is active, and the "Browse" link is selected. The source path is "svn/". The directory tree on the left shows the following structure:

- svn
  - branches
  - tags
  - trunk
    - AppSensor
    - AppSensor-Tutorial
      - .settings
      - WebContent
        - META-INF
        - WEB-INF (highlighted)
        - lib
      - build
      - notes
      - src

The table on the right lists the files in the selected directory:

Filename	Size	Rev
<a href="#">detectionPointThresholds.jsp</a>	364 bytes	r72
<a href="#">exceptionStats.jsp</a>	1022 bytes	r75
<a href="#">failedlogin.jsp</a>	413 bytes	r72
<a href="#">imports.jsp</a>	239 bytes	r71
<a href="#">lesson1.jsp</a>	2.2 KB	r80
<a href="#">lesson2.jsp</a>	2.3 KB	r80
<a href="#">lesson3.jsp</a>	2.8 KB	r82
<a href="#">lesson4.jsp</a>	2.9 KB	r82
<a href="#">lesson5.jsp</a>	2.5 KB	r81
<a href="#">lessonList.jsp</a>	176 bytes	r81

<http://code.google.com/p/appsensor/source/browse/#svn/trunk/AppSensor-Tutorial>

# AppSensor-Tutorial

- Lesson Based Application
- Concise & Simple Demo of ESAPI & AppSensor Code
- Purely JSP w/Java libs
- <http://defendtheapp.com/>

Currently logged in as: app [switch user](#)

lesson list  
[login page](#)  
[lesson 1](#)  
[lesson 2](#)  
[lesson 3](#)  
[lesson 4](#)  
[lesson 5](#)  
[lesson 6](#)  
[lesson 7](#)  
[lesson 8](#)

**Lesson:** lesson 1 - attack detection via ESAPI's input validation  
**Objective:** Observe how input is compared against the ESAPI property file. Invalid input is rejected and a generic event is created. Note that specific appsensor codes are not available when only using ESAPI validation

Please enter your data:  
Input

Received Input: null

**Detection Point: IE1**  
Count: 3  
Interval: 30  
Response: [log, logout]

**User Record**  
User Record: app  
Total Intrusions: 0  
Intrusions for IE1: 0

OWASP APPSENSOR PROJECT - [HTTP://OWASP.ORG/APPSENSOR](http://owasp.org/appsensor)

**Lesson:** lesson 4 - Feature Locking Global  
**Objective:** Observe how input is checked by AppSensor for common XSS patterns After the threshold is reached the page functionality is locked for all authenticated users. Log out to see that the anonymous user can still see the page

This page has been disabled for everyone

**Detection Point: IE14**  
Count: 2  
Interval: 35  
Response: [disableComponent]

**User Record**  
User Record: app  
Total Intrusions: 4  
Intrusions for IE14: 2

# Lesson Format

- Simple form with text input or drop down
- Malicious data checked by ESAPI or AppSensor
- Detection point listed with response actions / intrusion count

**Lesson:** lesson 1 - attack detection via ESAPI's input validation

**Objective:** Observe how input is compared against the ESAPI property file. Invalid input is rejected and a generic event is created. Note that specific appsensor codes are not available when only using ESAPI validation

Please enter your data:

Input

Received Input: some data

**Detection Point: IE1**

Count: 3

Interval: 30

Response: [log, logout]

---

**User Record**

User Record:

Anonymous

Total Intrusions: 0

Intrusions for IE1: 0

# Lesson 1: Validate w/ ESAPI

**Lesson:** lesson 1 - attack detection via ESAPI's input validation

**Objective:** Observe how input is compared against the ESAPI property file. Invalid input is rejected and a generic event is created. Note that specific appsensor codes are not available when only using ESAPI validation

```
String dataResult = "";  
try {  
    dataResult = ESAPI.httpUtilities()  
        .getParameter(request, "attackstring");  
} catch (ValidationException e){  
    //ESAPI Validation Exception  
    //Processed by AppSensor  
    // Automatically  
}
```

lesson1.jsp

```
Validator.HTTPParameterName=  
^[a-zA-Z0-9_]{0,32}$
```

```
Validator.HTTPParameterValue=  
^[a-zA-Z0-9.\-!@+=_]*$
```

ESAPI.properties

```
IntrusionDetector.Total.count=3
```

```
IntrusionDetector.Total.interval=30
```

```
IntrusionDetector.Total.actions=logout
```

appsensor.properties



# Lesson 2

## Validate w/ AppSensor

- Use AppSensor `AttackDetectorUtils.verifyXSSAttack`
- Customizable black list approach (regex)
- Catch obvious XSS probes
  - `alert(document.cookie)`
  - `<img src=.*script`
  - `<iframe>.*</iframe>`

Currently logged in as: Anonymous [switch user](#)

[lesson list](#)  
[login page](#)  
[lesson 1](#)  
[lesson 2](#)  
[lesson 3](#)  
[lesson 4](#)  
[lesson 5](#)  
[lesson 6](#)  
[lesson 7](#)  
[lesson 8](#)

**Lesson:** lesson 2 - attack detection via AppSensor's XSS inspector  
**Objective:** Observe how input is checked by AppSensor for common XSS patterns Invalid input is rejected and a specific AppSensor exception is created.

Please enter your data:  
Input    
Received Input: null

**Detection Point:** IE1  
Count: 3  
Interval: 30  
Response: [log, logout]

**User Record**  
User Record: Anonymous  
Total Intrusions: 0  
Intrusions for IE1: 0

OWASP APPSENSOR PROJECT - [HTTP://OWASP.ORG/APPSENSOR](http://owasp.org/appsensor)

# Lesson 2 - The Code

**Lesson:** lesson 2 - attack detection via AppSensor's XSS inspector  
**Objective:** Observe how input is checked by AppSensor for common XSS patterns Invalid input is rejected and a specific AppSensor exception is created.

```
dataResult = request.getParameter("attackstring");

boolean attackDetected =
    org.owasp.appsensor.AttackDetectorUtils.verifyXSSAttack(dataResult);

if (attackDetected) {
    dataResult = "Exception Caught By ESAPI Validation";
    new AppSensorException(
        appsensorID, "Invalid Input per AppSensor Detection",
        "Attacker is sending input that violates defined whitelists"
        + "("+ request.getRequestURI()+ ")"
        + " user ("
        + ESAPI.authenticator().getCurrentUser().getAccountName() + ")");
    dataResult = "removed";
}
```

lesson2.jsp



# Lesson 2

## appsensor.properties

- Define regex black list of xss patterns
- Black list ok for attack detection
- Define response thresholds as normal

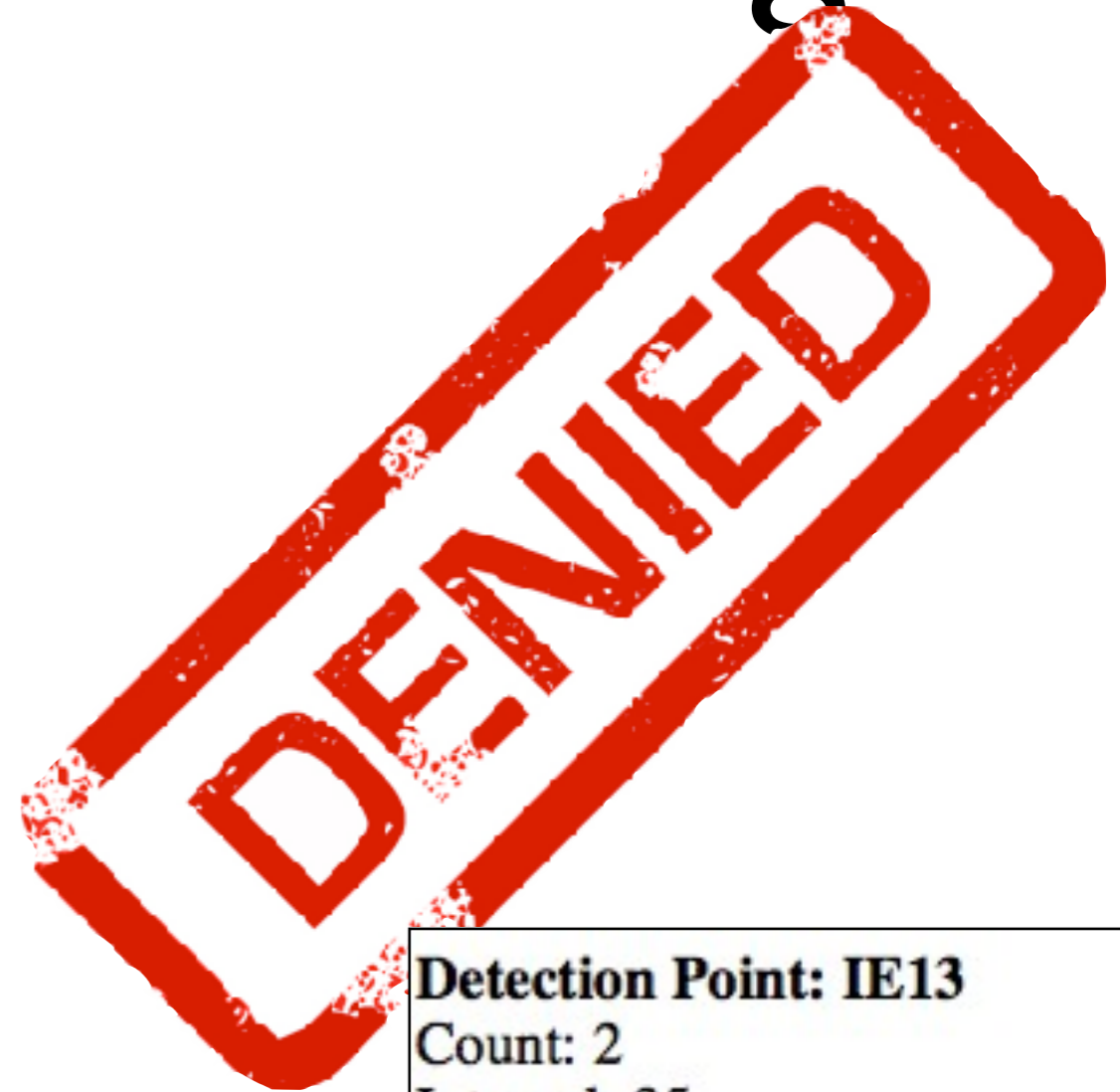
```
xss.attack.patterns=  
\"><script>,  
script.*document\\.cookie,  
<script>,  
<IMG.*SRC.*=.*script,  
<iframe>.*</iframe>  
  
...  
IntrusionDetector.IE1.count=3  
IntrusionDetector.IE1.interval=30  
IntrusionDetector.IE1.actions=log,logout
```

appsensor.properties

# lesson 3

## Per User Page Blocking

- Disable user's access to the page
- Good solution for sensitive operations - transfer funds, update address
- Just affects malicious user
- Simple with AppSensor



**Detection Point: IE13**  
Count: 2  
Interval: 35  
Response:  
[disableComponentForUser]

# lesson 3 - The Code

**Lesson:** lesson 3 - Feature Locking Per User

**Objective:** Observe how input is checked by AppSensor for common XSS patterns After the threshold is reached the page functionality is locked for this user. Log out to see that the anonymous user can still see the page

```
ASUser user = APPSENSOR.asUtilities().getCurrentUser();

boolean isActive =
AppSensorServiceController.isServiceActiveForSpecificUser
(request.getRequestURI(),user);

if (!(isActive)){
    %>This page has been disabled<%
}else{
    //display normal page
```

lesson3.jsp

# Lesson 3

## appsensor.properties

- Define normal thresholds
- Define how long page is disabled for user (30 minutes)

```
IntrusionDetector.IE12.count=2  
IntrusionDetector.IE12.interval=35  
IntrusionDetector.IE12.actions=disableComponentForUser  
IntrusionDetector.IE12.disableComponentForUser.duration=30  
IntrusionDetector.IE12.disableComponentForUser.timeScale=m
```

**appsensor.properties**

# Lesson 4

## Full Feature Blocking

- Block access to all users
- Possible for critical pages
- Better to shutoff page and investigate than risk compromise



# lesson 4 - The Code

**Lesson:** lesson 4 - Feature Locking Global

**Objective:** Observe how input is checked by AppSensor for common XSS patterns After the threshold is reached the page functionality is locked for all authenticated users. Log out to see that the anonymous user can still see the page

```
boolean isActiveForEveryone =  
AppSensorServiceController.isActive  
(request.getRequestURI());  
  
if (!(isActiveForEveryone)){  
    %>Page has been disabled for everyone<%  
}  
lesson4.jsp
```

```
IntrusionDetector.IE12.actions=disableComponent  
IntrusionDetector.IE12.disableComponent.duration=10  
appsensor.properties
```



# Additional Response Capabilities

OWASP AppSensor - Response Actions



## Open Web Application Security Project (OWASP)

### AppSensor - Response Actions

v0.6 **Draft** 27th August 2010

Colin Watson

Based on original ideas by Michael Coates and with contributions from John Melton.

**Feedback welcome**

[http://www.owasp.org/index.php/  
File:Owasp-appsensor-responses.pdf](http://www.owasp.org/index.php/File:Owasp-appsensor-responses.pdf)

# Additional Response Capabilities

**Table 1: AppSensor Responses**

CATEGORY		RESPONSE (ADDED SINCE v1.1)	
TYPE	DESCRIPTION	ID	DESCRIPTION
Silent	User unaware of application's response	ASR-A	Logging Change
		ASR-B	Administrator Notification
		ASR-C	Other Notification
Passive	Changes to user experience but nothing denied	ASR-D	User Status Change
		ASR-E	User Notification
		ASR-F	Timing Change
Active	Application functionality reduced for user(s)	ASR-G	Process Terminated
		ASR-H	Function Amended
		ASR-I	Function Disabled
		ASR-J	Account Logout
		ASR-K	Account Lockout
Intrusive	User's environment altered	ASR-L	Application Disabled
		ASR-M	Collect Data from User



# AppSensor @ Mozilla

# Mozilla Threat Profile

- Lots of users
- Many web apps
- Apps constantly growing & changing
- All code open source



# Mozilla Services

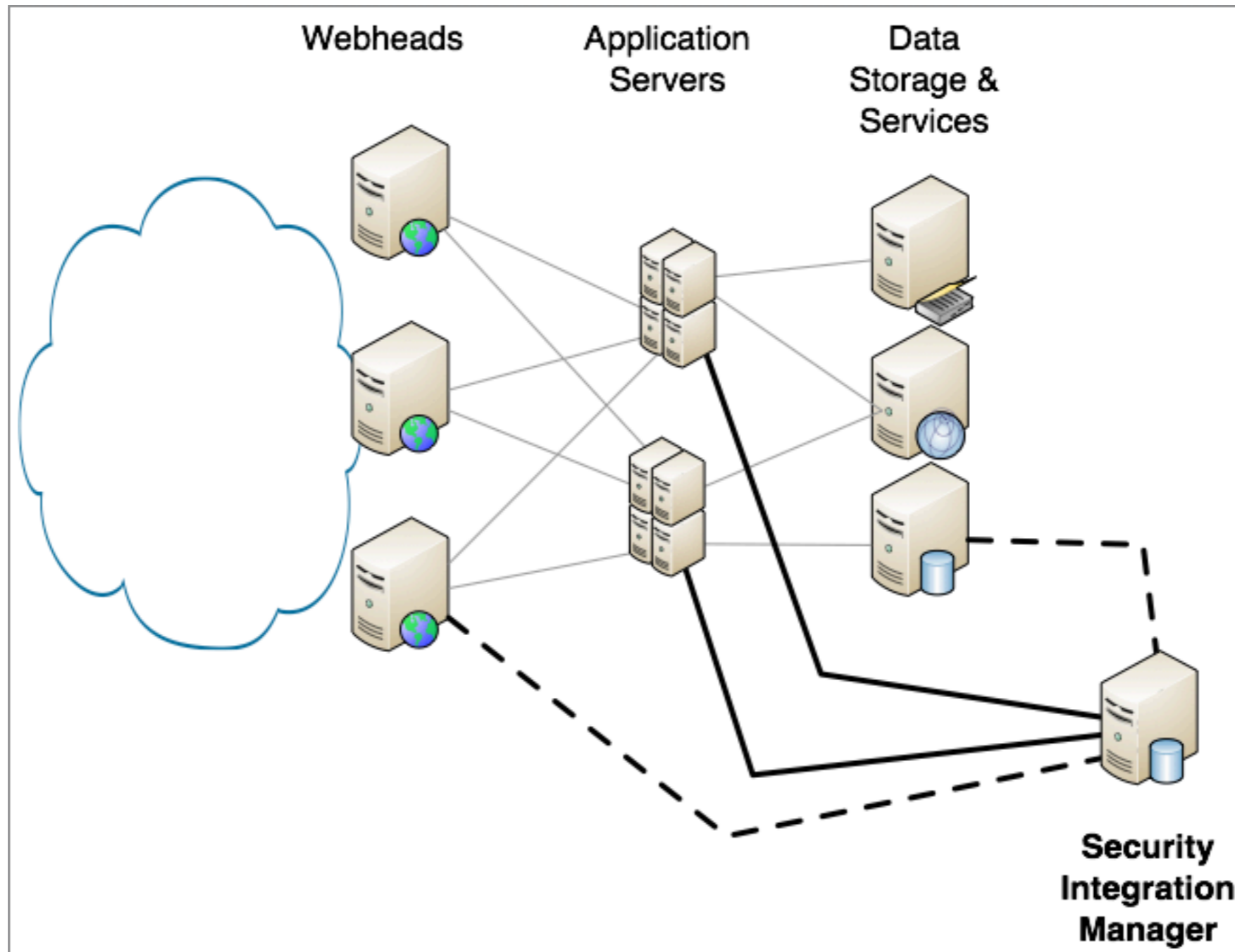
- Firefox Sync
  - Millions of users
  - Service based app
  - Stores encrypted user data
- Example detection points
  - Credential mismatch within URL request
  - Tampering with reset code
  - Account delete attempt without password



# What to Capture

- Threat model attack scenarios
  - Access Control Failures
  - Account lockouts
  - Failed CAPTCHA
- Monitor trends of interesting events
  - New privileged account created
  - Password reset requested
  - Account creations
  - Sensitive bug access
  - New attachment

# SIM Deployment



# Common Event Format (CEF)

- Emerging standard on logging format
- Easily parsed by security integration manager (sim)
- Enables AppSensor Logging

```
CEF:0|Mozilla|MozFooApp|1.0
|ACE0|Access Control Violation|8|rt=01
31 2010 18:30:01 suser=janedoe suid=55
act=Action Denied src=1.2.3.4
dst=2.3.4.5 requestMethod=POST
request=http://foo.mozilla.org/foo/
abc.php?a\=b
cs1Label=requestClientApplication
cs1=Mozilla/5.0 (Macintosh; U; Intel Mac
OS X 10.6; en-US; rv:1.9.2.2) Gecko/
20100316 Firefox/3.6.2
msg=Additional Data here
```

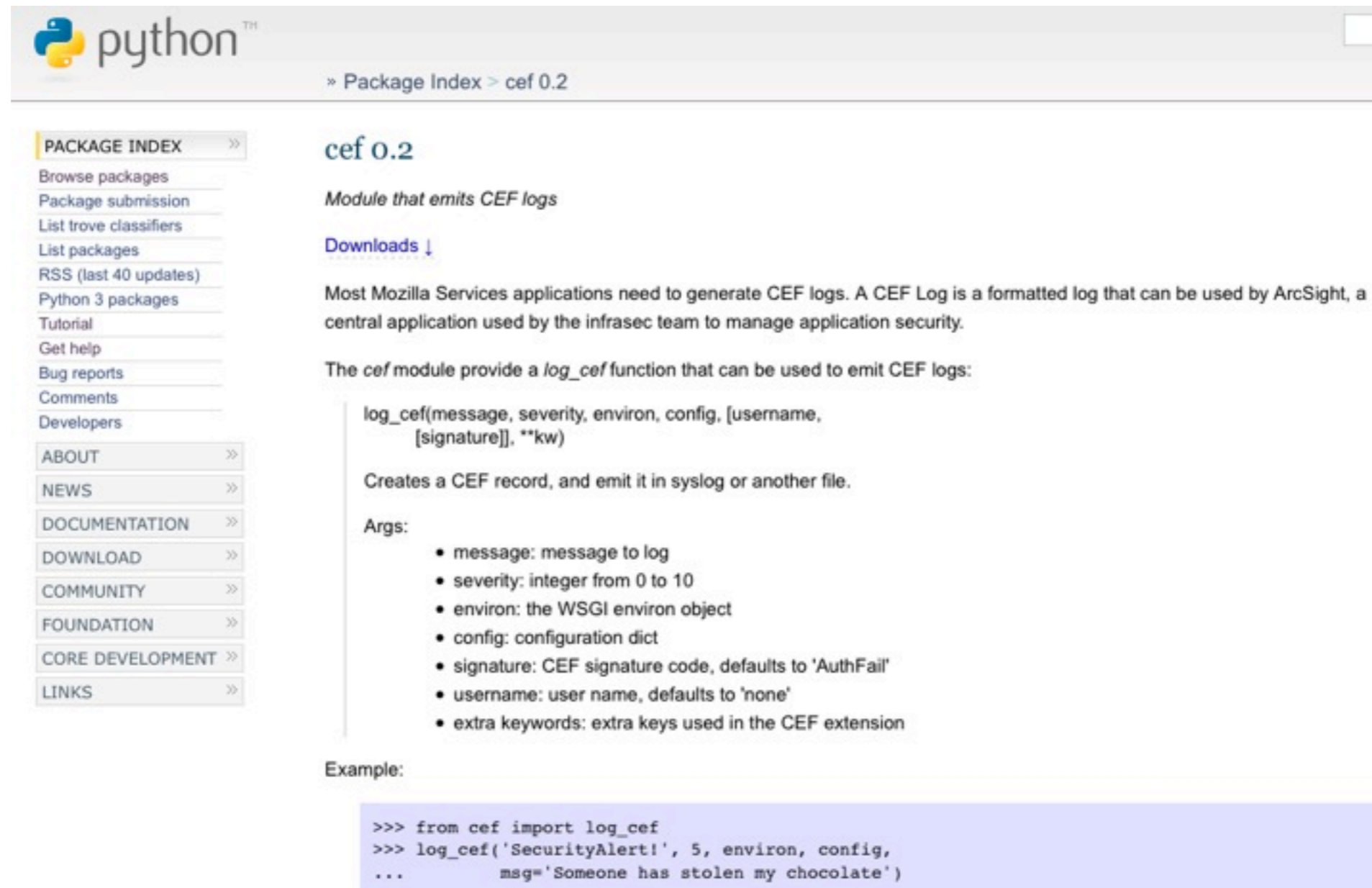
# Detection Point w/CEF

```
if (!$authdb->authenticate_user(fix_utf8_encoding($auth_pw)))  
{  
    if ($cef)  
    {  
        $message = new CommonEventFormatMessage(  
WEAVE_CEF_AUTH_FAILURE,  
        'User Authentication Failed', 3,  
        array('username' => $url_user, 'requestip' =>  
get_source_ip()));  
        $cef->logMessage($message);  
    }  
    report_problem('Authentication failed', '401');  
}
```

<http://hg.mozilla.org/services/>



# Python CEF @ PyPI



The screenshot shows the PyPI page for the 'cef' package. The page header includes the Python logo and the text 'python™'. Below the header, there is a breadcrumb trail: '» Package Index > cef 0.2'. On the left side, there is a navigation menu with links for 'PACKAGE INDEX', 'Browse packages', 'Package submission', 'List trove classifiers', 'List packages', 'RSS (last 40 updates)', 'Python 3 packages', 'Tutorial', 'Get help', 'Bug reports', 'Comments', and 'Developers'. Below the navigation menu, there are links for 'ABOUT', 'NEWS', 'DOCUMENTATION', 'DOWNLOAD', 'COMMUNITY', 'FOUNDATION', 'CORE DEVELOPMENT', and 'LINKS'. The main content area is titled 'cef 0.2' and contains the following text:

*Module that emits CEF logs*

[Downloads ↓](#)

Most Mozilla Services applications need to generate CEF logs. A CEF Log is a formatted log that can be used by ArcSight, a central application used by the infrasec team to manage application security.

The *cef* module provide a *log\_cef* function that can be used to emit CEF logs:

```
log_cef(message, severity, environ, config, [username,
      [signature]], **kw)
```

Creates a CEF record, and emit it in syslog or another file.

Args:

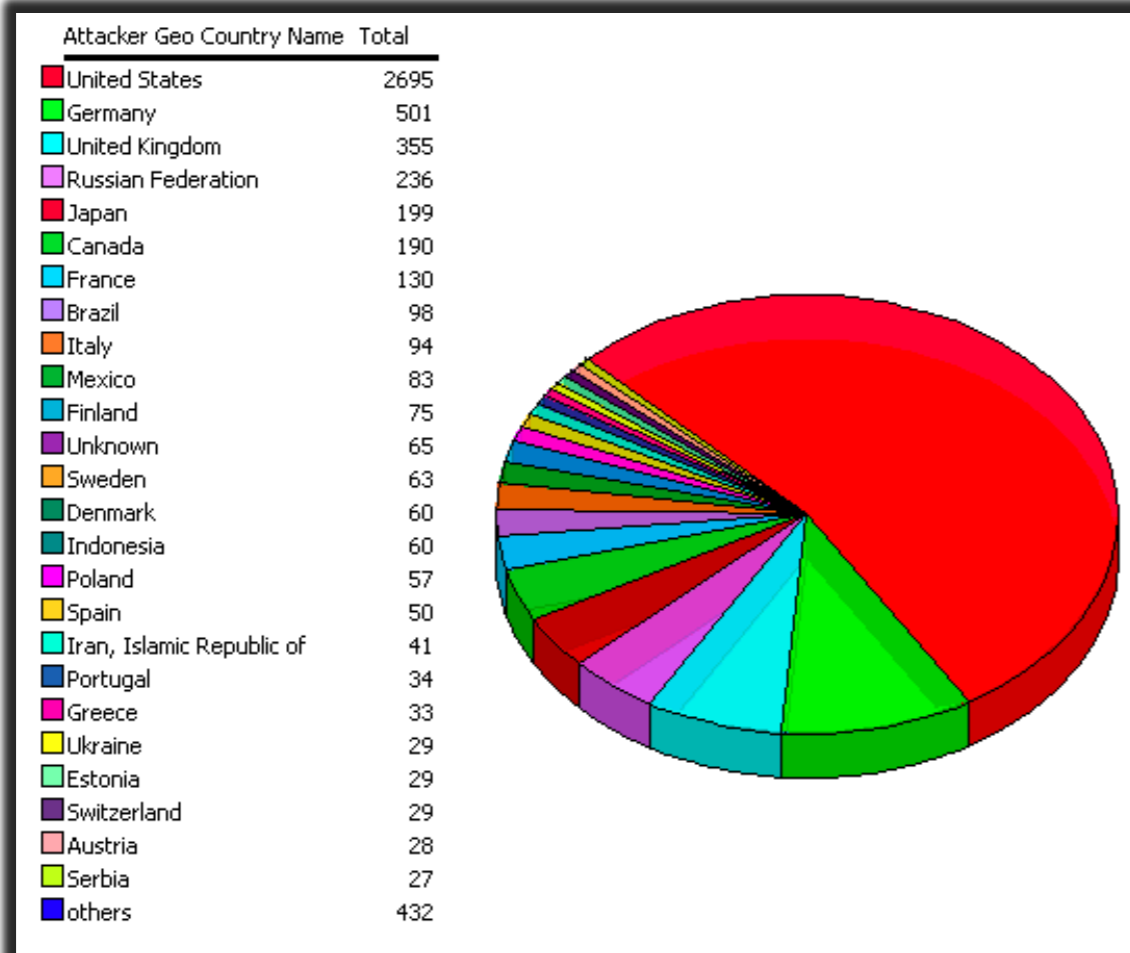
- message: message to log
- severity: integer from 0 to 10
- environ: the WSGI environ object
- config: configuration dict
- signature: CEF signature code, defaults to 'AuthFail'
- username: user name, defaults to 'none'
- extra keywords: extra keys used in the CEF extension

Example:

```
>>> from cef import log_cef
>>> log_cef('SecurityAlert!', 5, environ, config,
...         msg='Someone has stolen my chocolate')
```

<http://pypi.python.org/pypi/cef/>

# Data Analysis



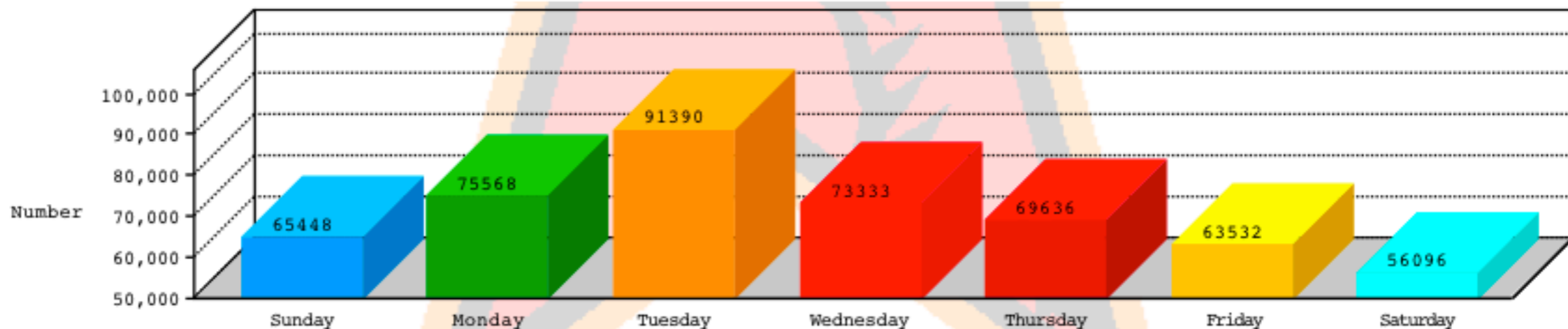
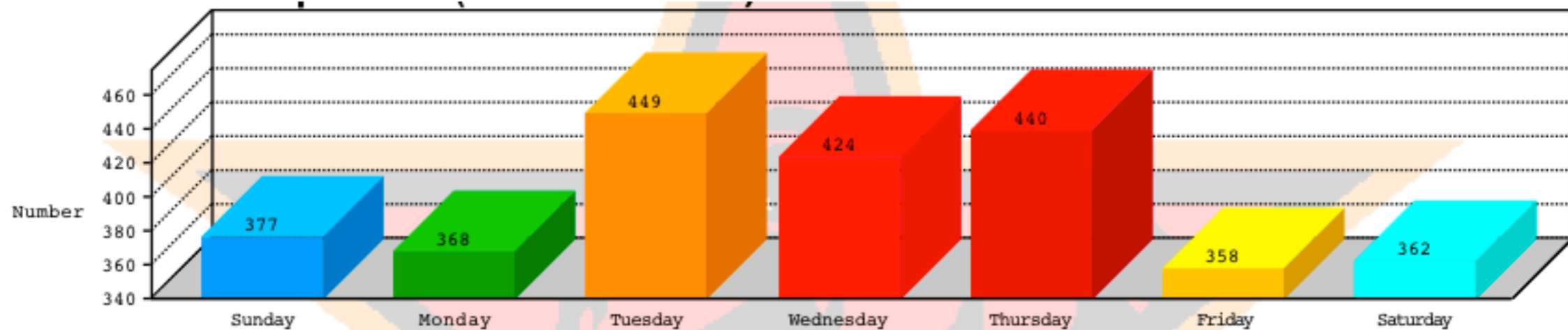
Name	Attacker	Target	Priority	Device Vendor	Device Product	Target Host	Request Url
User Account Created	sh...	76.17...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	cor...	74.99...	10.10...	5	mozilla	weave	wp-we... https://phx-sync29
User Authentication Failed	fjm...	71.16...	10.10...	5	mozilla	weave	wp-we... https://phx-sync29
User Authentication Failed	jco...	76.12...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
Password Reset	coc...	200.4...	10.8...	3	mozilla	weave	https://auth.servic
User Account Created	jej...	81.51...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	kak...	88.18...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
User Authentication Failed	ros...	69.25...	10.10...	5	mozilla	weave	wp-we... https://phx-sync29
User Account Created	jej...	81.51...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	jea...	86.77...	10.10...	5	mozilla	weave	wp-we... https://phx-sync29
User Authentication Failed	he...	72.37...	10.10...	5	mozilla	weave	wp-we... https://phx-sync16
User Authentication Failed	tip...	109.2...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
User Account Created	ba...	72.24...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	gm...	60.24...	10.10...	5	mozilla	weave	wp-we... https://phx-sync26
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
User Account Created	ma...	84.19...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	ds...	99.98...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
User Authentication Failed	vic...	123.1...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
Failed Captcha on create requ...	jej...	81.51...	10.8...	3	mozilla	weave	https://auth.servic
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	efu...	187.1...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
Failed Captcha on create requ...	eid	82.16...	10.8...	3	mozilla	weave	https://auth.servic
User Authentication Failed	ste...	78.22...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
User Authentication Failed	so...	81.57...	10.10...	5	mozilla	weave	wp-we... https://phx-sync28
User Authentication Failed	gm...	60.24...	10.10...	5	mozilla	weave	wp-we... https://phx-sync26
User Account Created	las...	194.1...	10.8...	3	mozilla	weave	https://auth.servic
User Account Created	na...	148.8...	10.8...	3	mozilla	weave	https://auth.servic

# Trend Analysis



## Sync - Weekly Report

7 Day Window 08-29-2010 to 09-04-2010



# AppSensor - More Info

[http://www.owasp.org/index.php/Category:OWASP\\_AppSensor\\_Project](http://www.owasp.org/index.php/Category:OWASP_AppSensor_Project)

<http://code.google.com/p/appsensor/>

[owasp-appsensor-project@lists.owasp.org](mailto:owasp-appsensor-project@lists.owasp.org)

[mcoates@mozilla.com](mailto:mcoates@mozilla.com)

[michael.coates@owasp.org](mailto:michael.coates@owasp.org)

<http://michael-coates.blogspot.com>

@\_mwc