



From Rivals to BFF: WAF & VA Unite

Brian Contos, Chief Security Strategist
Imperva Inc.

brian.contos@imperva.com

+1 (650) 832.6054

OWASP

07.23.2009

Copyright © The OWASP Foundation

Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation

<http://www.owasp.org>

Agenda

- Background
- Web Application Assessment
- Web Application Firewalls
- WAF + VA Integration
- More Information




Application Security Appetizers

- Functional software flaws are often more devastating than security flaws
 - ▶ Most application development teams have a queue filled with functional prioritized flaws
 - ▶ Not all of them are fixed right away, or even within a year; some never get fixed
 - ▶ Security flaws usually are not & usually should not be treated differently from functional
 - ▶ The difference is WAF can help address most security issues; functional has less options
- It was thought that the skills needed to hack applications were high
 - ▶ Then we discovered it was actually just as if not easier than hacking infrastructure
- It was also thought that we only have to worry about high visibility sites
 - ▶ We found that automation + search engines + opportunistic attackers is prominent
- Most Web apps aren't being protected at even the most minimal levels
 - ▶ Secure code requires extra effort; results can be hard to measure – so it's often not done
 - ▶ It takes more time and money and requires skills that the current team might not have
 - ▶ Consider audit logs, IO, and CPU
- WAF, VA, code review... got dragged into a debate that few still believe in
 - ▶ Finding problems and fixing problems are very different issues
 - ▶ WAF doesn't fix everything – consider a bad implementation of encryption
 - ▶ Addressing a security vulnerability with a code fix or WAF is a question of risk management, resources, time, etc – one isn't always a better solution than the other
 - ▶ WAF gives you more options than fixing/not fixing the code - fix it outside of the code, block, alert



WAF+VA - PCI DSS 6.6 Clarification

PCI 6.6 Supplement states, “Proper implementation of both options (application code review AND Web Application Firewall) would provide the best multi-layered defense.”



Information Supplement: Payment Card Industry Data Security Standard (PCI DSS) Requirement 6.6
Code Reviews and Application Firewalls

General

PCI DSS Requirement 6.6 provides two options that are intended to address common threats to cardholder data and ensure that input to web applications from untrusted environments is inspected “top to bottom.” The details of how to meet this requirement will vary depending on the specific implementation supporting a particular application. Forensic analyses of cardholder data compromises have shown that web applications are frequently the initial point of attack upon cardholder data, through SQL injection in particular.

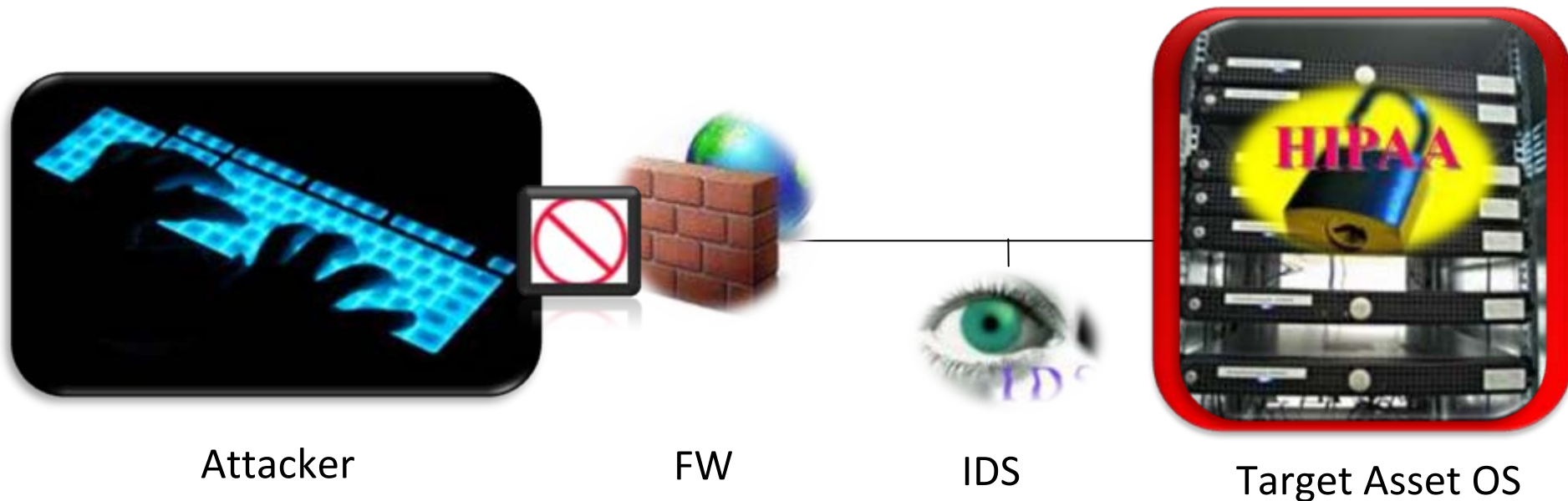
The intent of Requirement 6.6 is to ensure web applications exposed to the public Internet are protected against the most common types of malicious input. There is a great deal of public information available regarding web application vulnerabilities. The minimum vulnerabilities to consider are described in Requirement 6.5. (Refer to the References section for other sources of information on web application testing.)

Proper implementation of both options would provide the best multi-layered defense.

PCI SSC recognizes that the cost and operational complexity of deploying both options may not be feasible. Further, one or the other option may not be possible in some

VA Integration – Not New

FW + IDS + OS VA: Rule #1 with SIEM in '02



1. FW accept packet
2. Same event is detected by the IDS as an attack (CVE)
3. Earlier VA information tells us that the target OS is vulnerable to said attack (CVE)
4. Further risk information tells us business/compliance criticality; attacker/target info
5. Alert, block, TCP-RST to attacker & target; if internal attacker block at layer-2

When Looking at the Stats it Sometimes Feels Like a Losing Battle

■ 92% of Web applications have vulnerabilities¹

- Cross Site Scripting – 80%
- SQL Injection – 62%

- ▶ Identifying and fixing vulnerabilities is time consuming and costly

■ Web attacks are increasing; financially motivated = professional bad guys

- ▶ SQL injection attacks and vulnerabilities increased (2008)²
 - SQL attacks more sophisticated and automated

- ▶ Credit card losses: AT&T, Nikon, AOL, Voxant.com, PortTix
- ▶ XSS exploits: Netscape, Amazon, Google, Facebook, MySpace

■ The average breach costs over \$6M; breaches cost an average of [\\$197 per compromised record](#)³





Web Application Assessment Nut-shelled

OWASP

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Web Application Assessment

Three Most Common Methods

- Source code review



- Manual penetration test



- Automated vulnerability scanning



Post Assessment Panacea

■ Ideally

- ▶ Custom code is **immediately** fixed by programmers and application is redeployed
- ▶ Patches for 3rd party components are **immediately** installed

■ The above is of course a very romantic and unrealistic view of application development



Post Assessment Reality

■ Fix code

- ▶ **Resource allocation** – programmers are on other projects, features, other bugs
 - Written by sub-contractor (now has to be re-hired)
- ▶ **Time span** – even with resources it takes time to identify root cause, patch the code and then test the patch
- ▶ **(un)Availability** – patches for third party apps are dependent on the vendor
 - Zero Day gap is widening
 - Schedule/Coordinate system downtime
 - Introduce instability
- ▶ **New code** – can introduce new risks

■ Virtual patching with WAF

- ▶ **Fast** – policies can be active in a matter of minutes
- ▶ **Cost Effective** – WAF management (especially with VA help) needs less time than coding





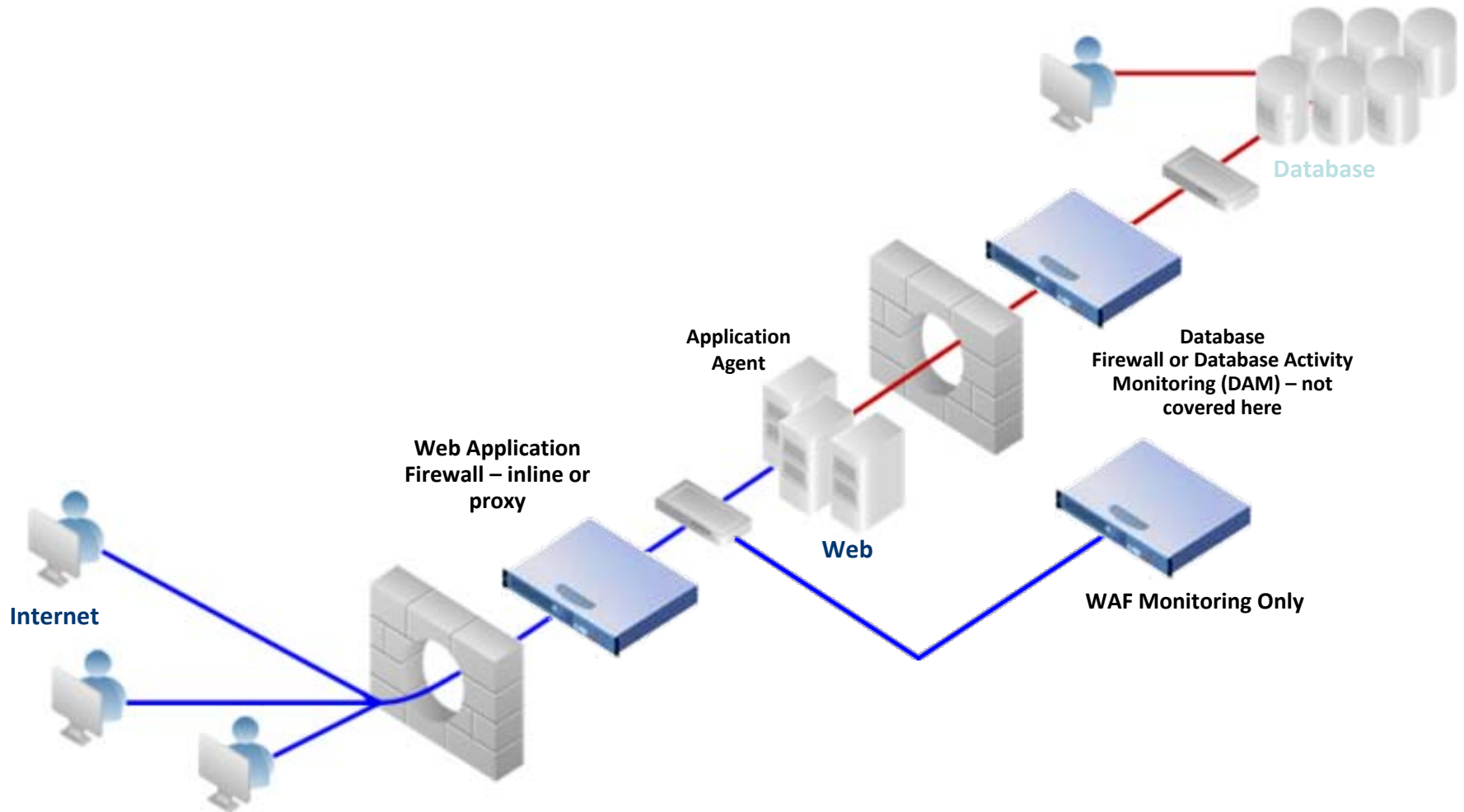
Web Application Firewalls

OWASP

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

WAF Architecture Options



WAF Introduces

■ Better **visibility** into customer environment

- ▶ Enhances accuracy; leverages dynamic profiling
- ▶ Enables customers to monitor attacks attempting to exploit scanned vulnerabilities
- ▶ Allows customers to enforce stricter policies for specific types of attacks

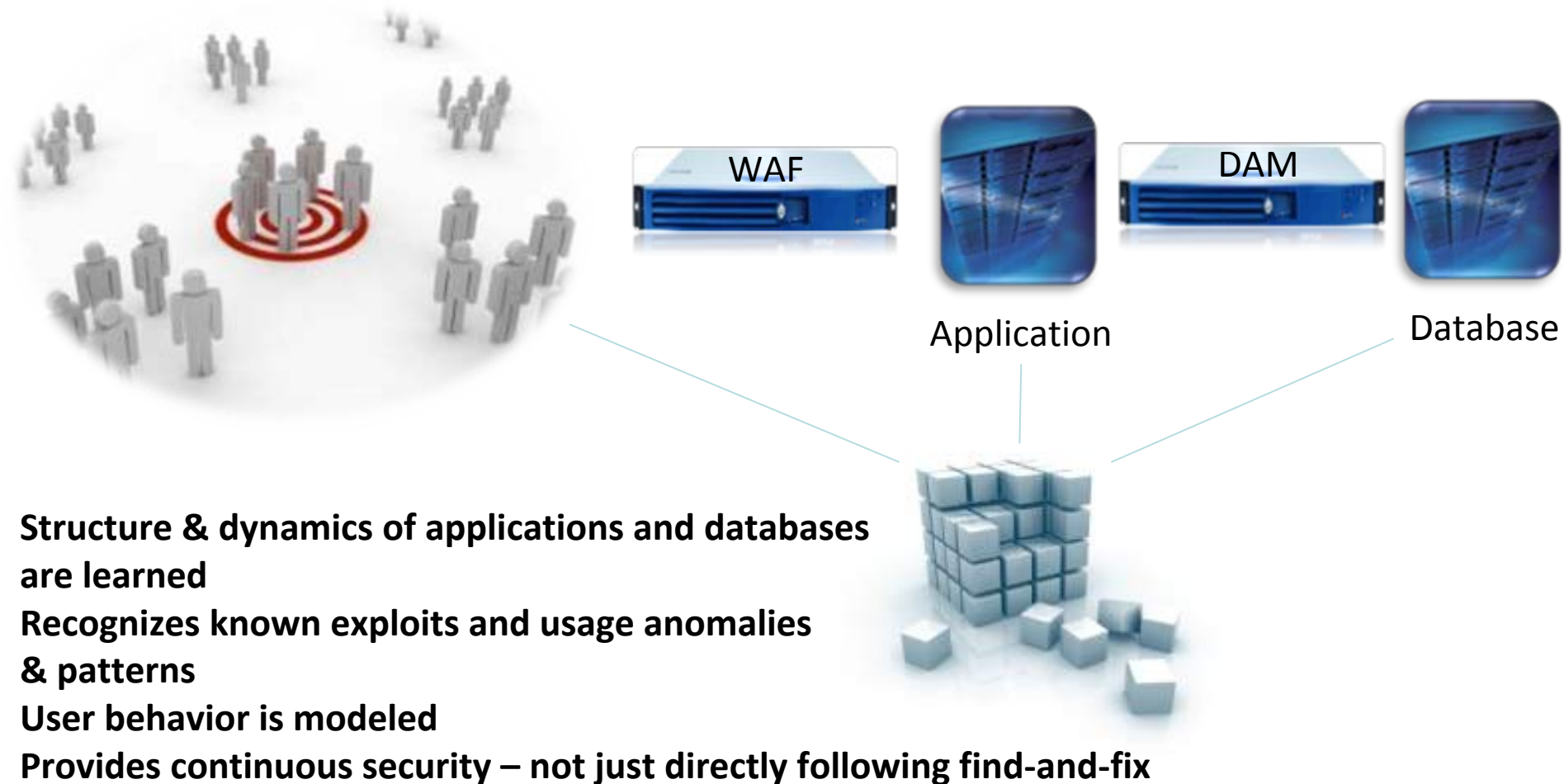


■ Enables **best practice** to proactively block attacks while still scheduling and fixing vulnerabilities in underlying application



Dynamic Profiling


People, Applications, & Databases




Dynamic Profiling Continued

Compare Profiles Against Observed Traffic
Detect Malicious Activity
Detect Usage Policy Exceptions

- Volume
- Location
- Time
- Data Type
- Upload
- Download





- Login
- BCOPY
- BDELETE
- GET
- POST
- Etc.

(SQL Injection) 1 OR 1=1, 1' OR '1'='1, 1'1, 1 EXEC SP_ (or EXEC XP_), 1 AND USER_NAME() = 'dbo'

Select, update, insert, alter, drop, backup, kill, shutdown, **truncate**, create, revoke, deny, restore

Just a Bit on Databases Part I

Discovery, Categorization, & Validation

Where are my applications & databases?



Databases



Applications



Replicated Databases



Unauthorized Systems
& Services Web/SOA

Just a Bit on Databases Part II

Discovery, Categorization, & Validation

What sensitive data resides on them?



ABC	123	Other
ert	654	5546 4857 8138 9872
ffdd	555	8574 2201 1587 1295
ytryj	1265	3571 2252 4467 8849
nnj	98	7145 7585 9872 0002

ABC	123	Credit Card Numbers
ert	654	Cat
ffdd	555	Dog
ytryj	1265	Fish
nnj	98	Bird



Test Server Using Real Customer Data

Leveraging Validation Algorithms Such as the Luhn Algorithm



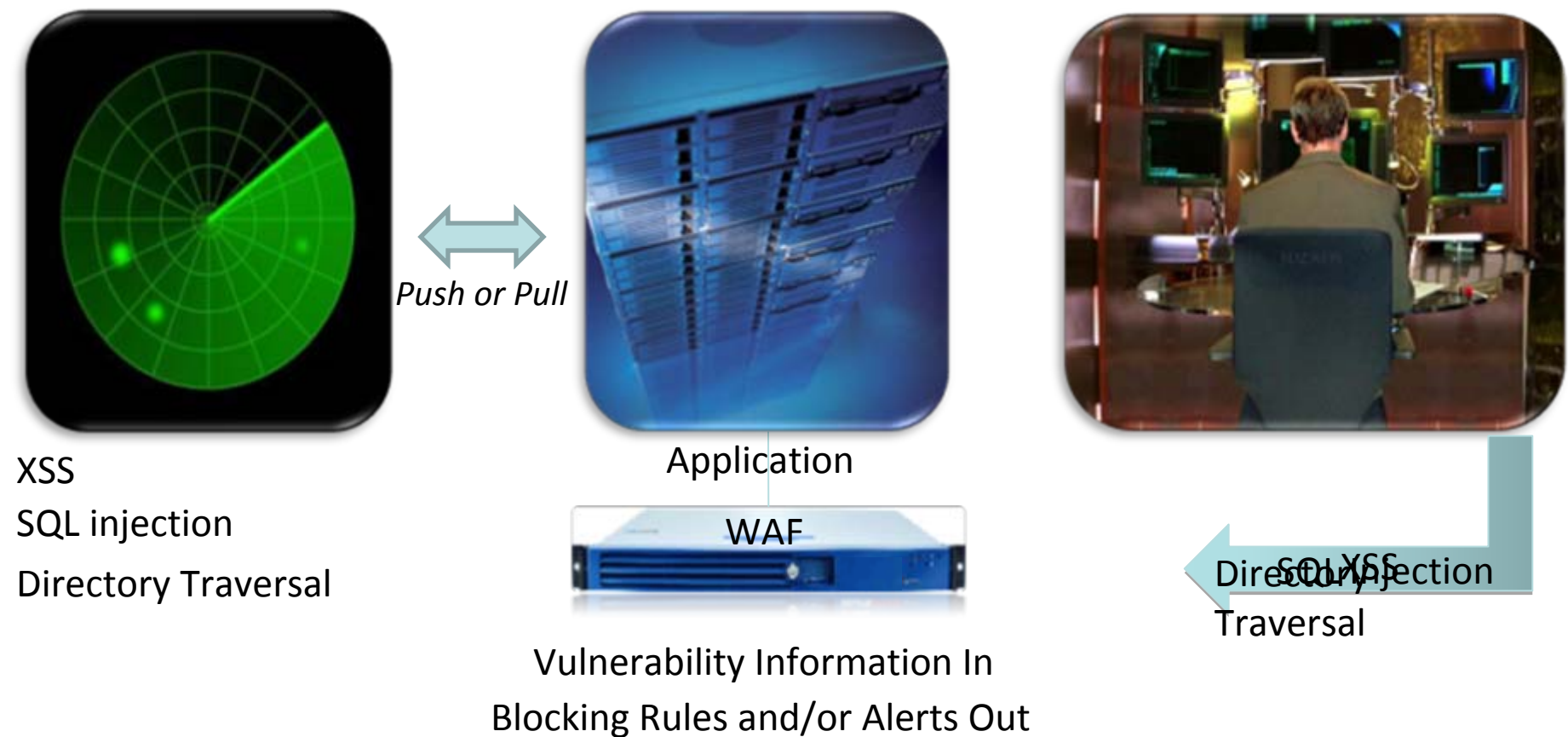
WAF & VA Integration

OWASP

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

WAF Integration with VA Software and Services



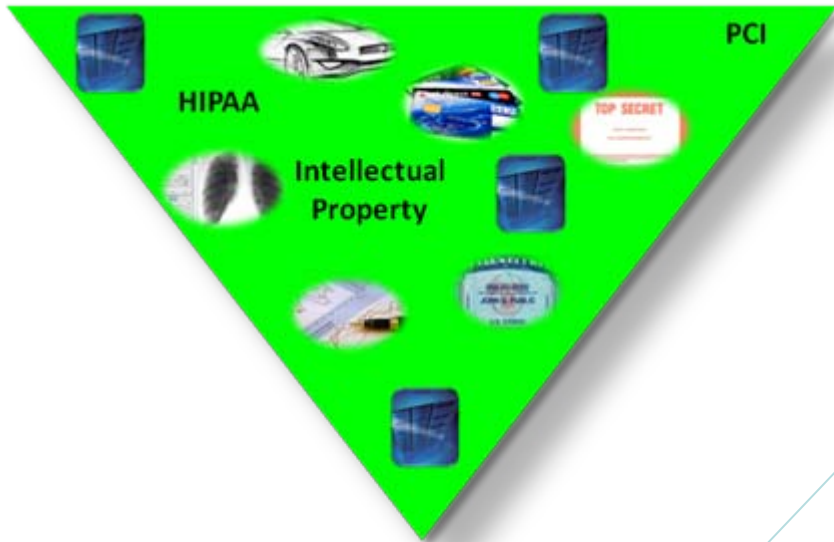
With many solutions you can re-run the scan & if the WAF prevents the exploit it is then categorized as virtually patched; with some implementations this is done automatically from the WAF - it prompts an ad-hoc scan to check the rule's effectiveness

Closing the Loop

- Need to periodically reassess
 - ▶ New types of attacks
 - ▶ Application changes
 - ▶ Configuration changes
- With WAF – Scanner integration
 - ▶ Both sides are kept up-to-date with new attack types
 - ▶ WAF detects application changes and invokes ad-hoc scan for the new / modified modules



WAF Feeds VA



WAF



Assessment
Software/Service



Real World Integration Example Between Imperva SecureSphere & A Leading VA Solution

1. Configure Imperva SecureSphere to access the VA solution using a secure API key
2. Vulnerabilities are detected and verified by VA solution
3. Imperva SecureSphere downloads a website's vulnerability data from the VA solution's API interface
4. The Imperva SecureSphere user selects the vulnerabilities to block or alert on and applies rules to a selected policy
5. The VA solution user uses the VA solution to retest and verify successful blocking with Imperva SecureSphere and close the vulnerabilities



SDLC

- Application Security Life Cycle is not a singular event in time it is a continuous process
- Creating a continuous process requires automation, instrumentation and faster response times
- Application vulnerability scanners automate the assessment phase
- Web Application Firewalls provide for faster response times during policy setting phase and provide for automation of enforcement and measurement phases
- Combining application scanners & WAF provides automation of the policy setting phase & a smooth closure of the life cycle loop

WAF & VA: That was Then

- False Dichotomy
- Technical Issues
- Poor Communication



WAF & VA: This is Now

WAF & VA Work Better Together

