Apache Web Platform Security

Ivan Ristic
Founder, Thinking Stone
ivanr@webkreator.com
+44 7766 508 210

OWASP AppSec Europe
April 2005

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

Talk Overview

1. Introduction

- What is this talk about?
- Defining the Apache Web platform
- About "Apache Security"
- About the speaker

What is this talk about?

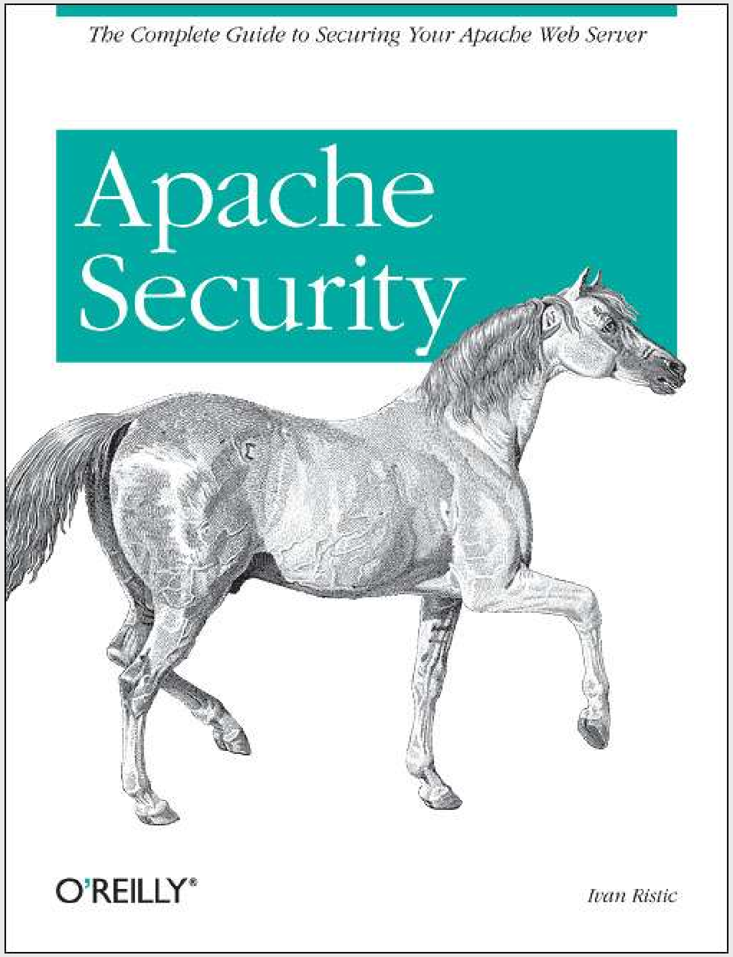
- A high-level overview of everything you need to know if you are deploying Apache
- Loosely based on my book, Apache Security
- A mixture of network security, host security, and web application security, in the combination relevant for the Apache web server

Defining the Apache Web Platform

- Web server
- Application server or application server front-end
- mod_php, mod_perl, Tomcat, etc
- Reverse proxy
  - Performance
  - Load balancing and scalability
  - Architectural flexibility (centralisation, integration, decoupling, access control)
  - Security (web application firewall)
- Probably not the most performant of web servers, but certainly the best choice when all factors (price, performance, flexibility, extensibility, available expertise) are considered

About “Apache Security”

- Everything you need to know to deploy Apache securely
- Discussions on all levels: high-level content followed by technical details
- Published by O’Reilly in March 2005; 420 pages


The Complete Guide to Securing Your Apache Web Server

Apache Security

O'REILLY®

Ivan Ristic

About the Speaker

- Developer / architect / administrator, spent a great deal of time looking at web security issues from different points of view.
- Author of **ModSecurity**, an open source Web firewall/IDS.
- Author of **Apache Security**, published by O'Reilly in March 2005.
- Founder of **Thinking Stone**, a web security company.

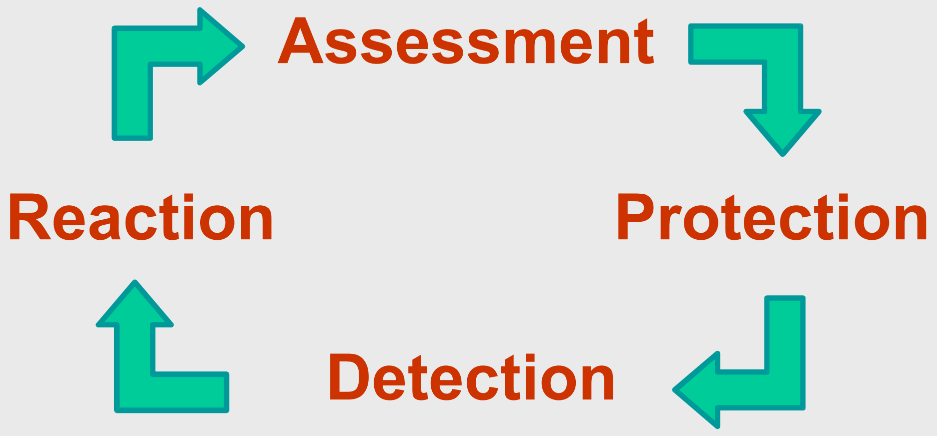
2. Problem Overview

- What is security?
- Three web system views:
  - User view
  - Network view
  - Process view
- What are the threats?

What Is Security?

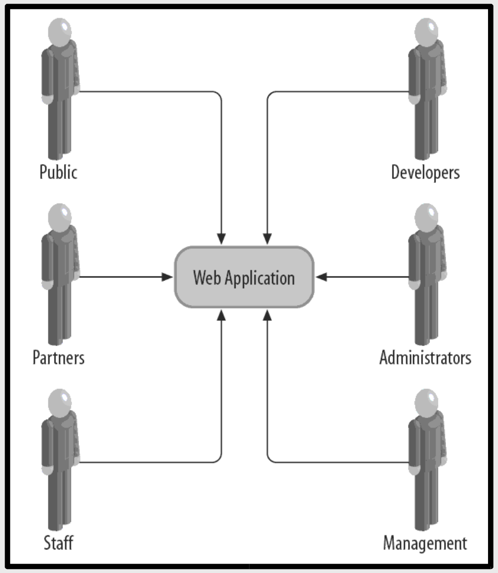
- Static definition
  - Confidentiality
  - Integrity
  - Availability
  - Accountability
- Dynamic definition
  - Assessment
  - Protection
  - Detection
  - Reaction

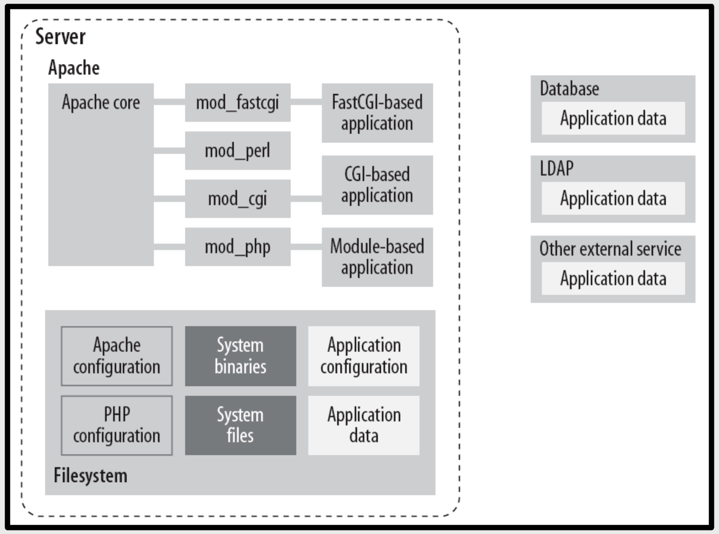
CIA^2

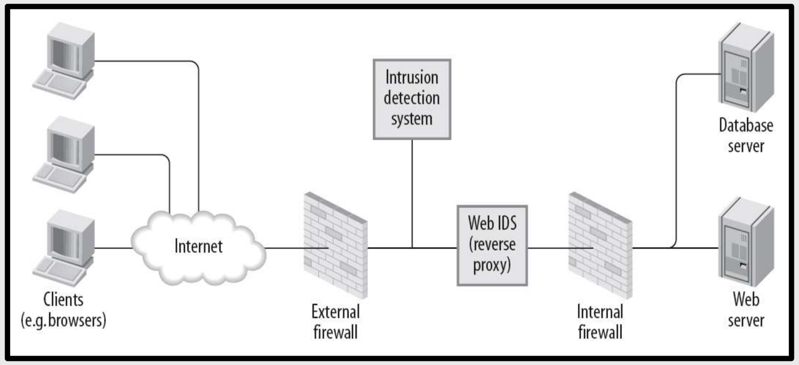


System views

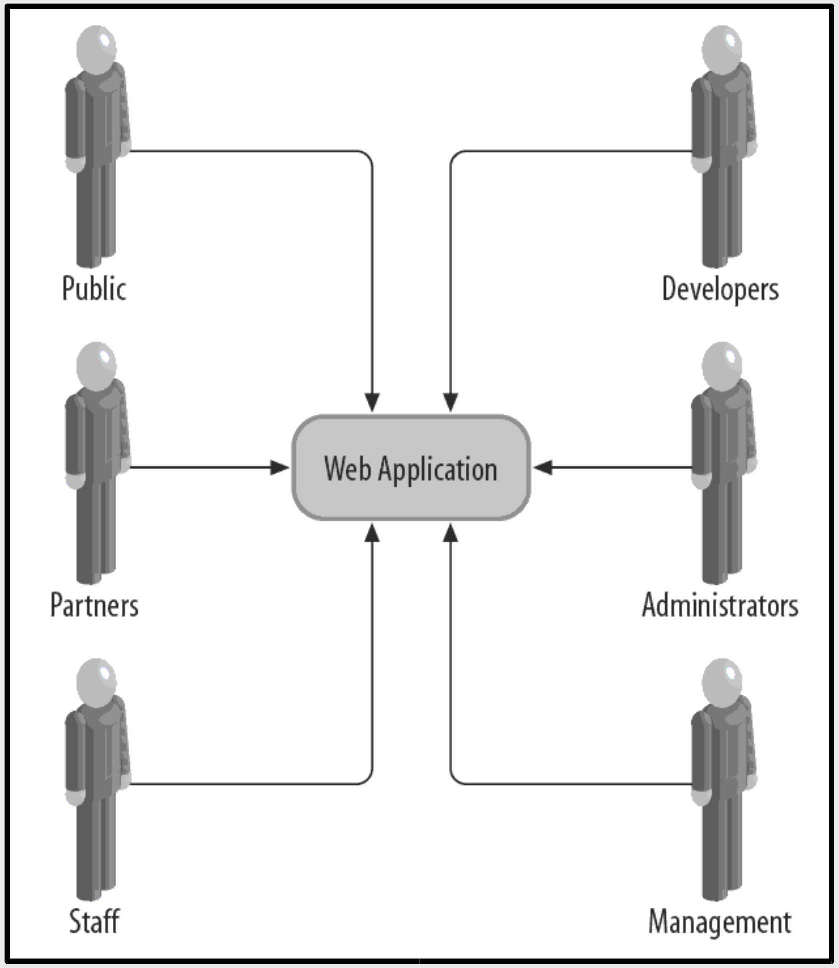
1. User view
2. Network view
3. Process view



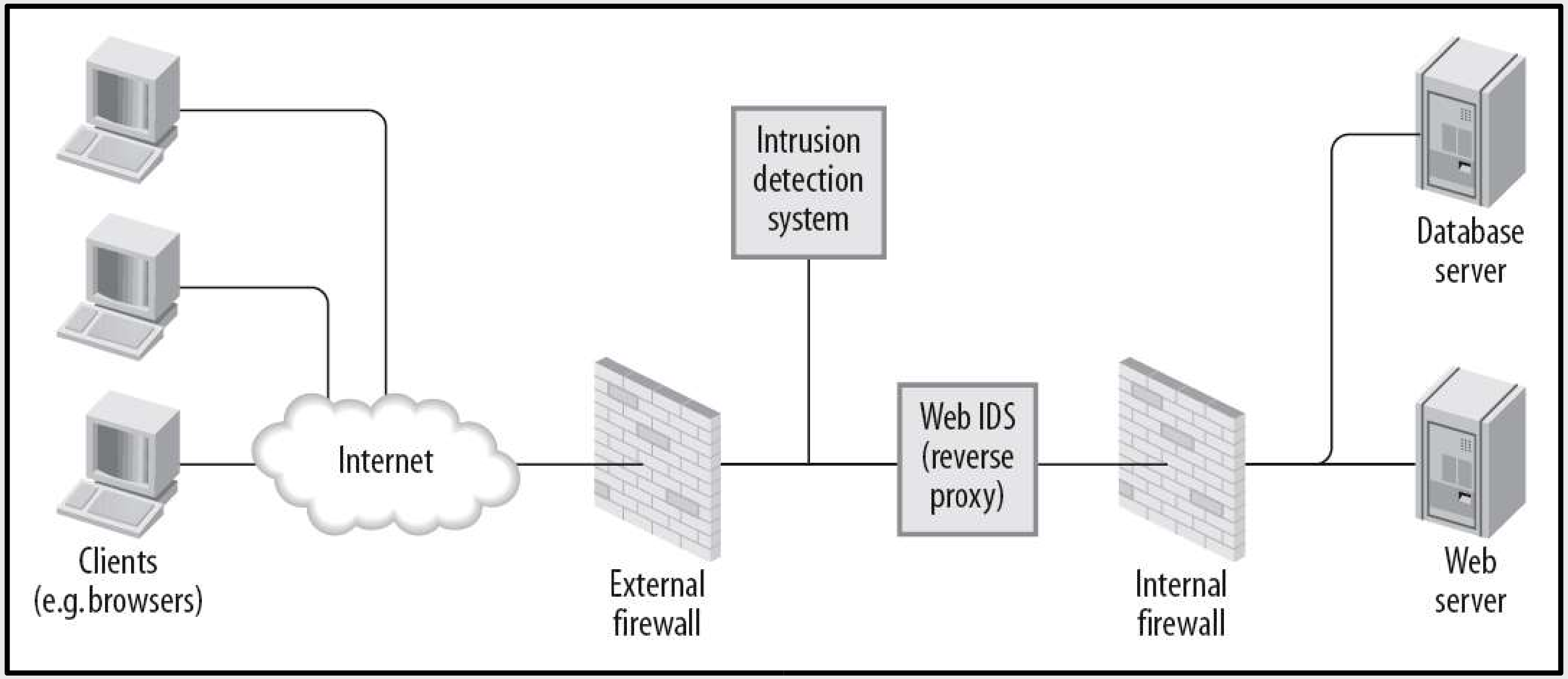




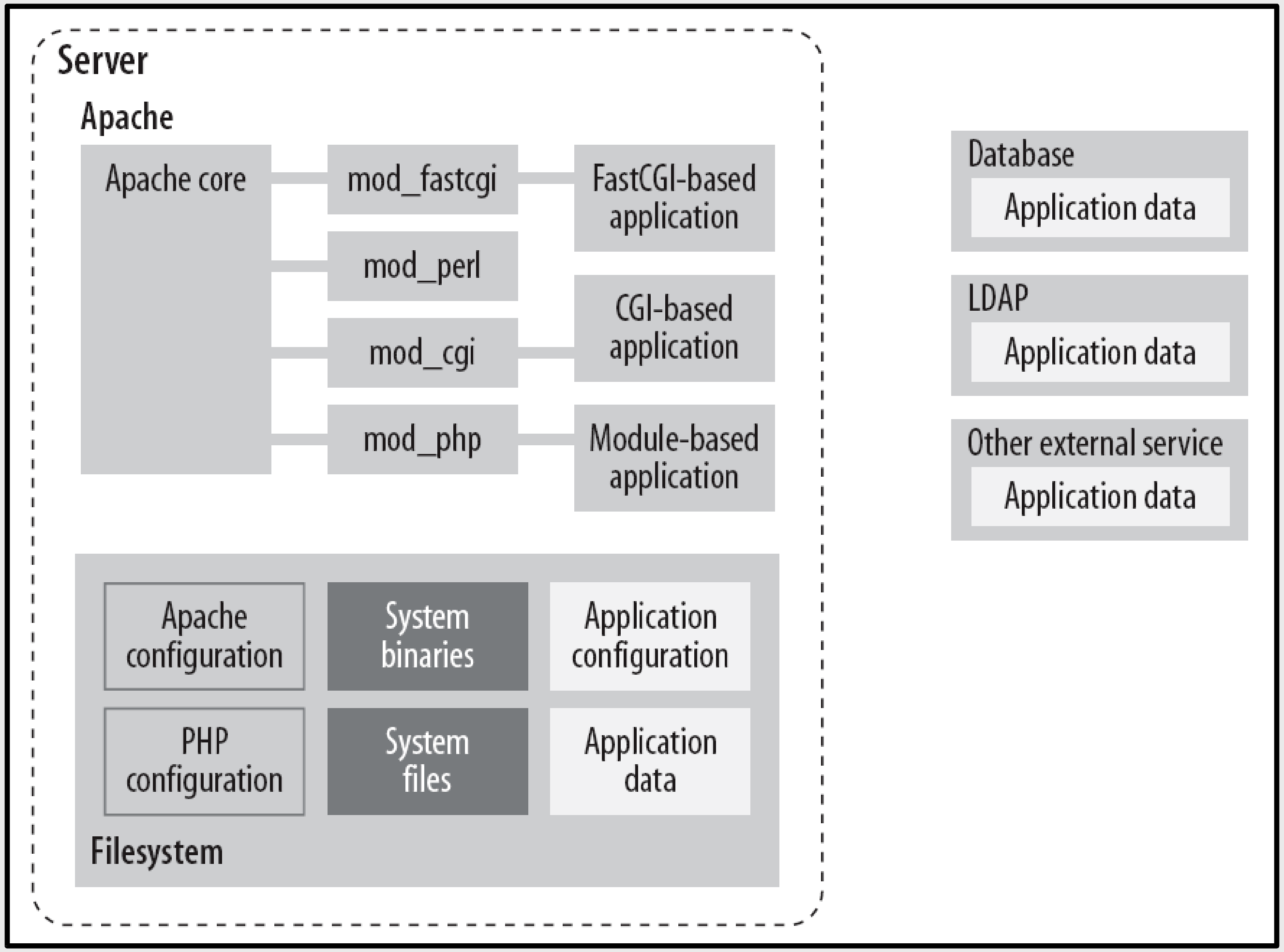
User view (1/3)


Public
Developers
Partners
Web Application
Administrators
Staff
Management

Network view (2/3)



Process view (3/3)



Possible Dangers

You can expect to experience five classes of problem:

- **Apache vulnerabilities**
- **Configuration problems**
- **Denial of Service attacks**
- **Web application security problems**
- **Attacks on users**

3. Choosing the Strategy

- Helper techniques
- Defensible systems
- Formulating the strategy

Your strategy sets up the stage for what happens later.

Helper Techniques

- Threat modelling
- System hardening matrix
  - Hardening techniques on one axis
  - System categories on the other (e.g. test, development, production, mission critical systems)
- Risk assessment
  - Exploitability, damage potential, asset value
- Patching plan
  - Patch immediately
  - Patch the next working day
  - Patch when the vendor patch comes our, or within five working days (when installed from source)

Defensible Systems

- **Defensible networks**, a term coined by Richard Bejtlich in "The TAO of Network Security Monitoring" (highly recommended).
- Four basic principles:
  - **Minimal**
  - **Compartmentalized**
  - **Maintainable**
  - **Observable**

Formulating the Strategy

- Our strategy formulated:
  - Accept you will fail
  - Be realistic about your resources
  - Compartmentalise
  - Start secure (know your stuff or find someone who does)
  - Remain secure (i.e. patch regularly)
  - Know what is happening
  - Be vigilant
  - React quickly

4. Installation and configuration

- **Use Apache 2**
- Keep up-to-date
- Use the latest version, apply the patches, verify the authenticity of the source code
- Construct configuration from scratch
- Use only the modules you need
- Configure limits
- Configure to fail securely
- **Use SSL**

5. Sharing Apache

- Sharing with developers
- Sharing with others (virtual hosting)
- Problems:
  - Shared server resources (CPU, RAM)
  - Ability to execute binaries on the server
  - File permissions
  - Shared web server process
  - Shared domain names
- **Who controls the web server?**

6. Denial of Service Attacks

- Network-based attacks
- HTTP-based attacks
- Real-life problems

Network-based DoS Attacks

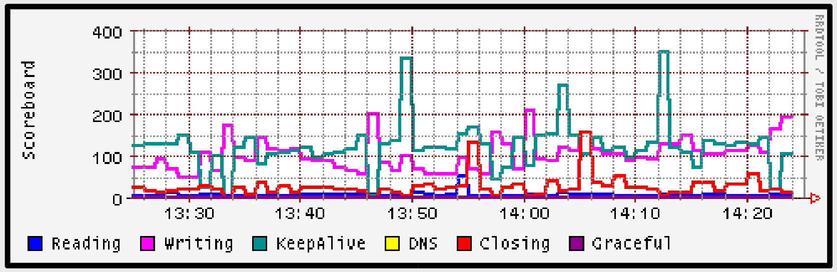
- Very little you can do on the web server level
- Some can be defended from at the network firewall level
- Enable SYN cookies in the operating system
- Be prepared:
  - Know when you are being attacked
  - Have the details of your upstream provider ready

HTTP-based Attacks

- Possible types of attack:
  - Apache vulnerabilities
  - Attacks against the programming model (problem with the limited number of Apache processes)
  - Brute-force attacks.
- Solutions:
  - Patch Apache regularly
  - Configure Apache limits
  - Figure out who is attacking you. Reject such traffic in the firewall. **Often difficult to do.**

Real-life Problems

- What you will encounter:
  - Slow clients and large files (and download accelerators) problems
  - Traffic spikes (e.g. Slashdot, cyber-activism, attacks from competitors)
  - Badly written web applications
- Mitigation:
  - Fix web applications
  - Buy more RAM
  - Tweak the Keep-Alive settings
  - Add response compression (mod_deflate)
  - Add caching (mod_cache)
  - Traffic-shaping modules



7. Logging and Monitoring

- Thinking about log retention
- Increase logging levels
- Include application logs in your plans
- Apache health monitoring
- Event monitoring

Log Retention

- What do you want to keep and for how long?
- Put logs on a separate partition
- Make sure the filesystem does not overflow (log rotation)
- Keep recent logs on the server for easy access and troubleshooting
- Centralise logs for additional security
  - Syslog
  - Many people are using Syslog-NG
  - Spread toolkit (mod_log_spread)

Increase Logging Detail

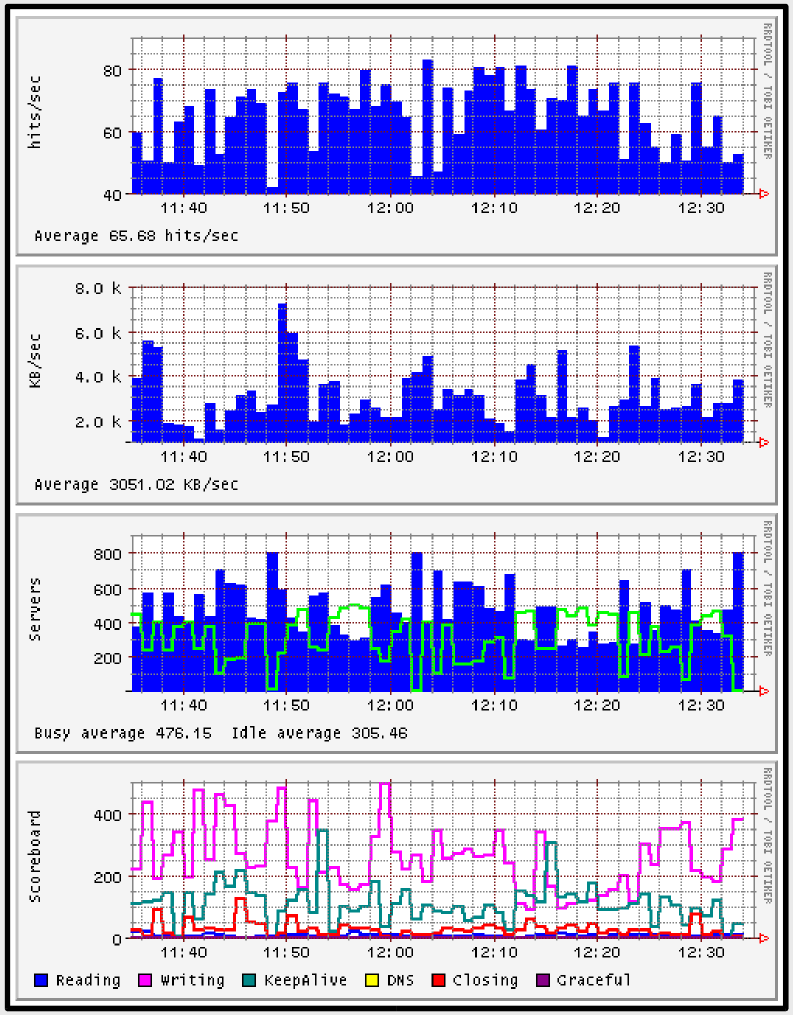
- Add information to the access_log:
  - Referrer
  - User agent
  - Username
  - Session token
  - UNIQUE_ID
  - Transaction duration
- Set error_log level to "**info**"
- Use mod_security:
  - Log POST data
  - Performance measurement

Application Logs

- Treat them equally (rotation, centralisation)
- If you can, get the application to utilise the HTTP codes:
  - Log analysis will be much easier
  - You can configure mod_security to selectively log POST data based on the response code

Apache Health Monitoring

- Performance
- Availability
- mod_status
- mod_watch
- apache-monitor



An hour of activity of the Apache running on www.apache.org. Produced with apache-monitor.

Event Monitoring

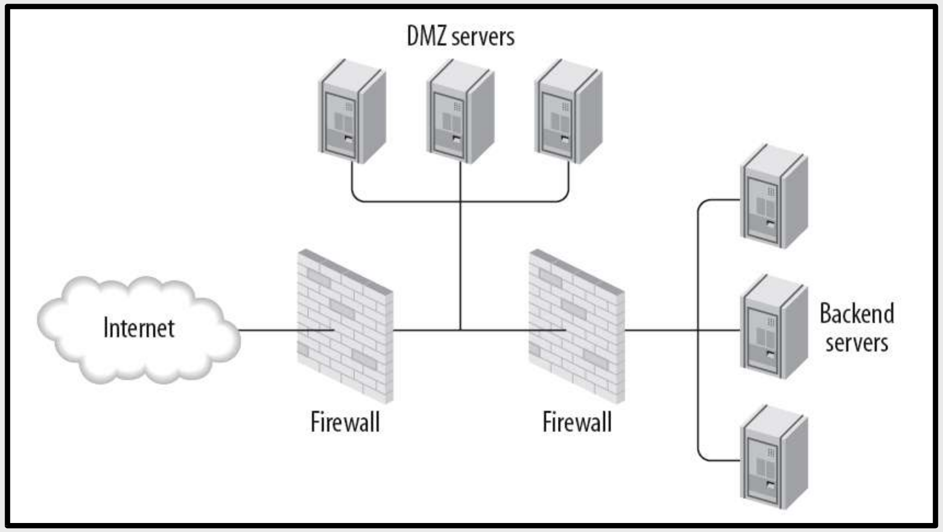
- Funnel all events into log files
- Do not rely on ad-hoc notification
- Have automated scripts inspect the logs on regular basis
  - Artificial Ignorance
- Real-time monitoring is very cool, but difficult to get right.
  - Swatch
  - SEC (Simple Event Correlator)

8. Infrastructure

- Network security
- Host security
- Isolation strategies
- Use of reverse proxies

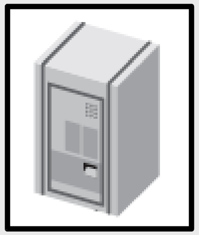
Network Security

- Network firewall
- Demilitarised zones
- Centralized logging
- Network monitoring
- Intrusion detection
- Web intrusion detection
- Independent security assessment



Host Security

- **Timely patching**
- Restricted user access
- Minimal services
- Host-based firewall
- Kernel hardening (grsecurity, SELinux)
- Event monitoring
- Process monitoring
- Integrity validation



Isolation strategies

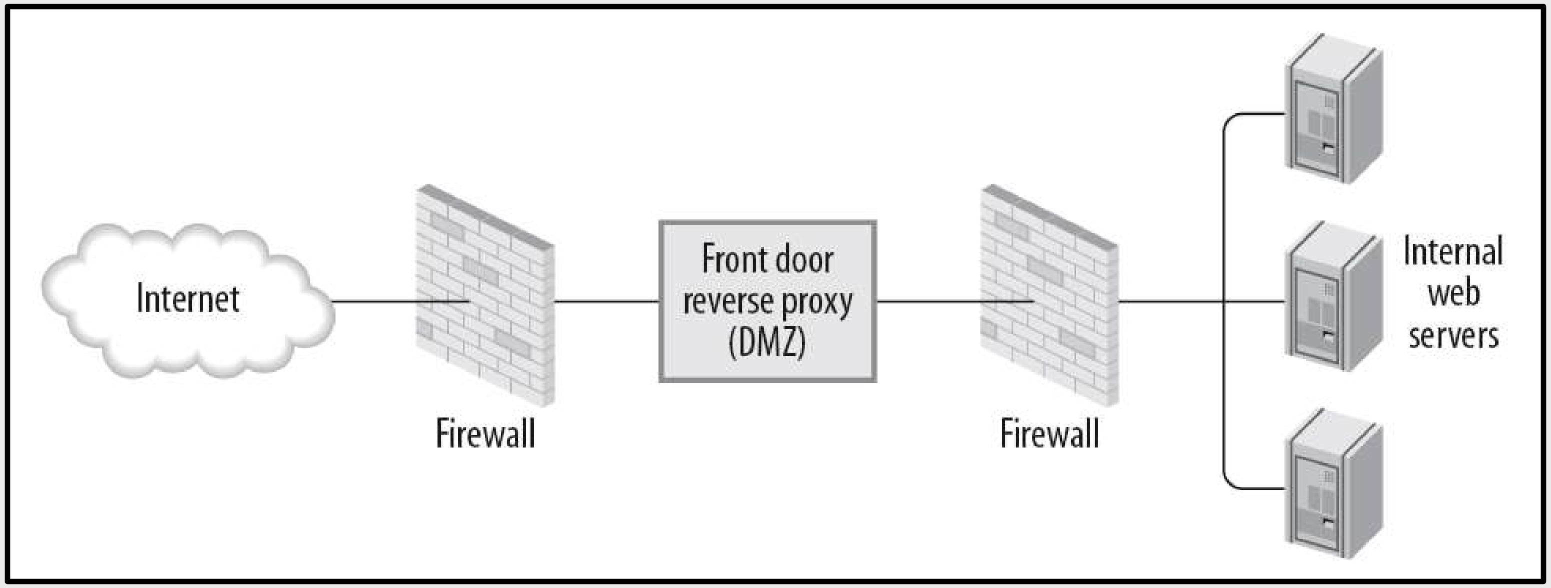
- Techniques:
  - Run as separate user (**suEXEC**, **FastCGI**)
  - Filesystem isolation (**permissions**, **chroot**)
  - Virtual servers
  - Physical servers
- Apache from operating system
- Applications from Apache
- Application modules from each other
- Use separate (restricted) database accounts, or separate database engines

Use of Reverse Proxies

- Reverse proxy patterns
  - Front door
  - Integration reverse proxy
  - Protection reverse proxy
  - Performance reverse proxy
  - Scalability reverse proxy
- Logical patterns, orthogonal to each other
- Often deployed as a single physical reverse proxy

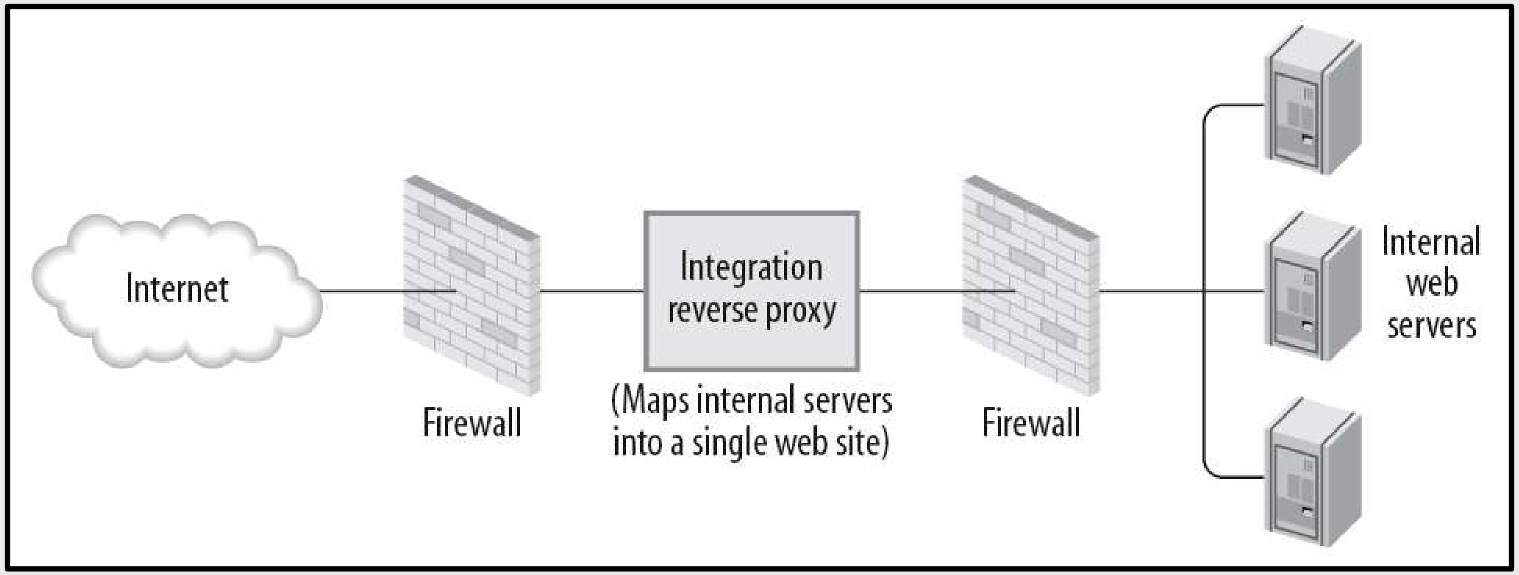
Front Door (1/5)

- Make all HTTP traffic go through the proxy
- Centralisation makes access control, logging, and monitoring easier



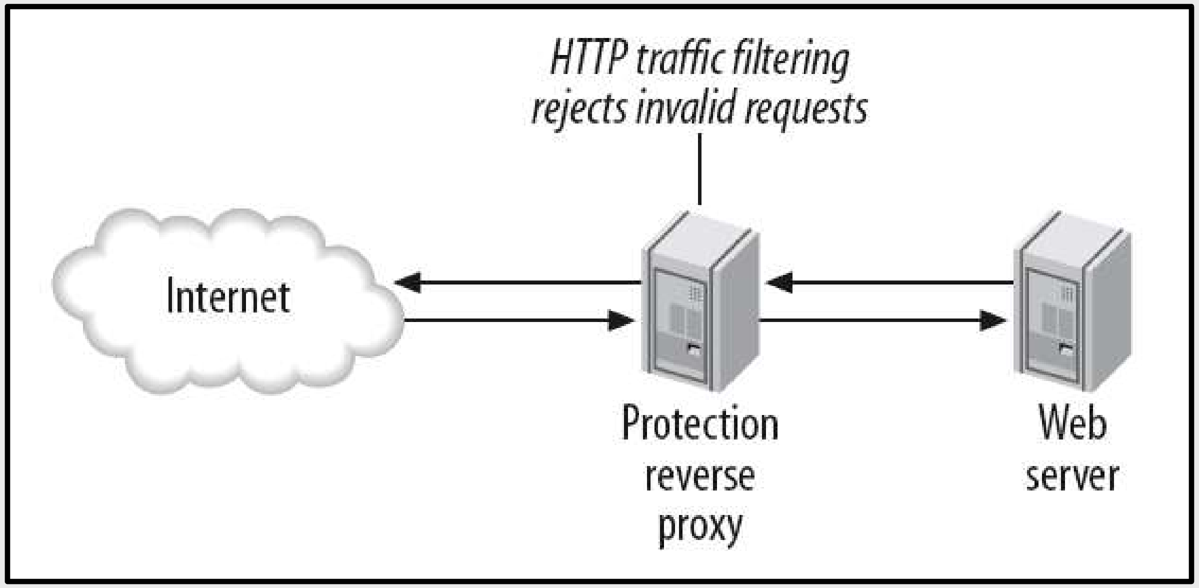
Integration Reverse Proxy (2/5)

- Combine multiple web servers into one
- Hide the internals
- Decouple interface from implementation


Internet
Firewall
Integration reverse proxy
(Maps internal servers into a single web site)
Firewall
Internal web servers

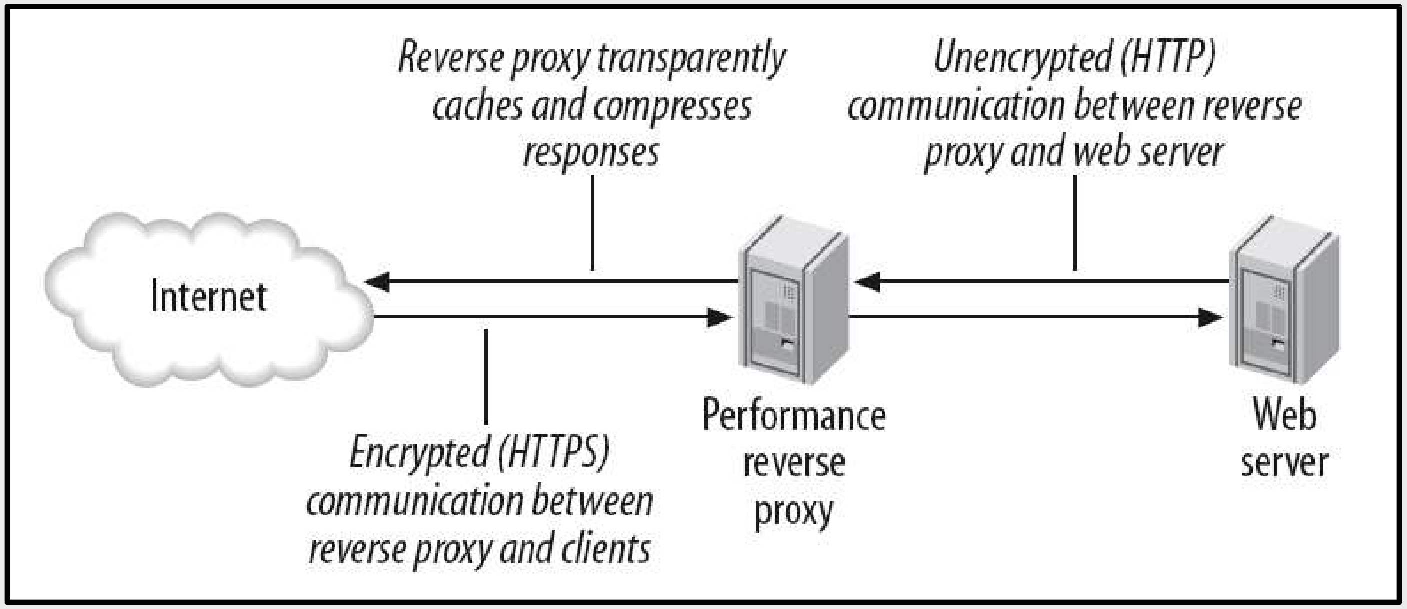
Protection Reverse Proxy (3/5)

- Observes traffic in and out
- Blocks invalid requests and attacks
- Prevents information disclosure



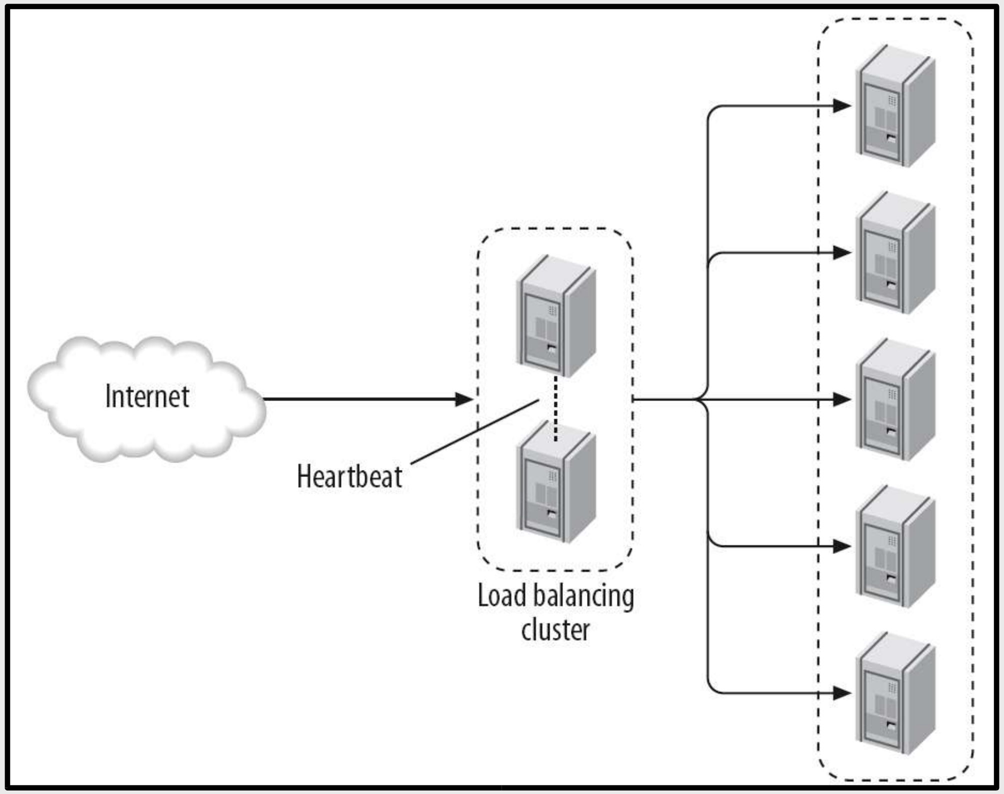
Performance Reverse Proxy (4/5)

- Transparent caching
- Transparent response compression
- SSL termination


Reverse proxy transparently caches and compresses responses
Unencrypted (HTTP) communication between reverse proxy and web server
Internet
Encrypted (HTTPS) communication between reverse proxy and clients
Performance reverse proxy
Web server

Scalability Reverse Proxy (5/5)

- Load balancing
- Fault tolerance
- High availability



Talk Overview

Questions?

Thank you!

Download this presentation from **http://www.thinkingstone.com/talks/**