



Java Web Application Security

Develop. Penetrate. Protect. Relax.

Matt Raible

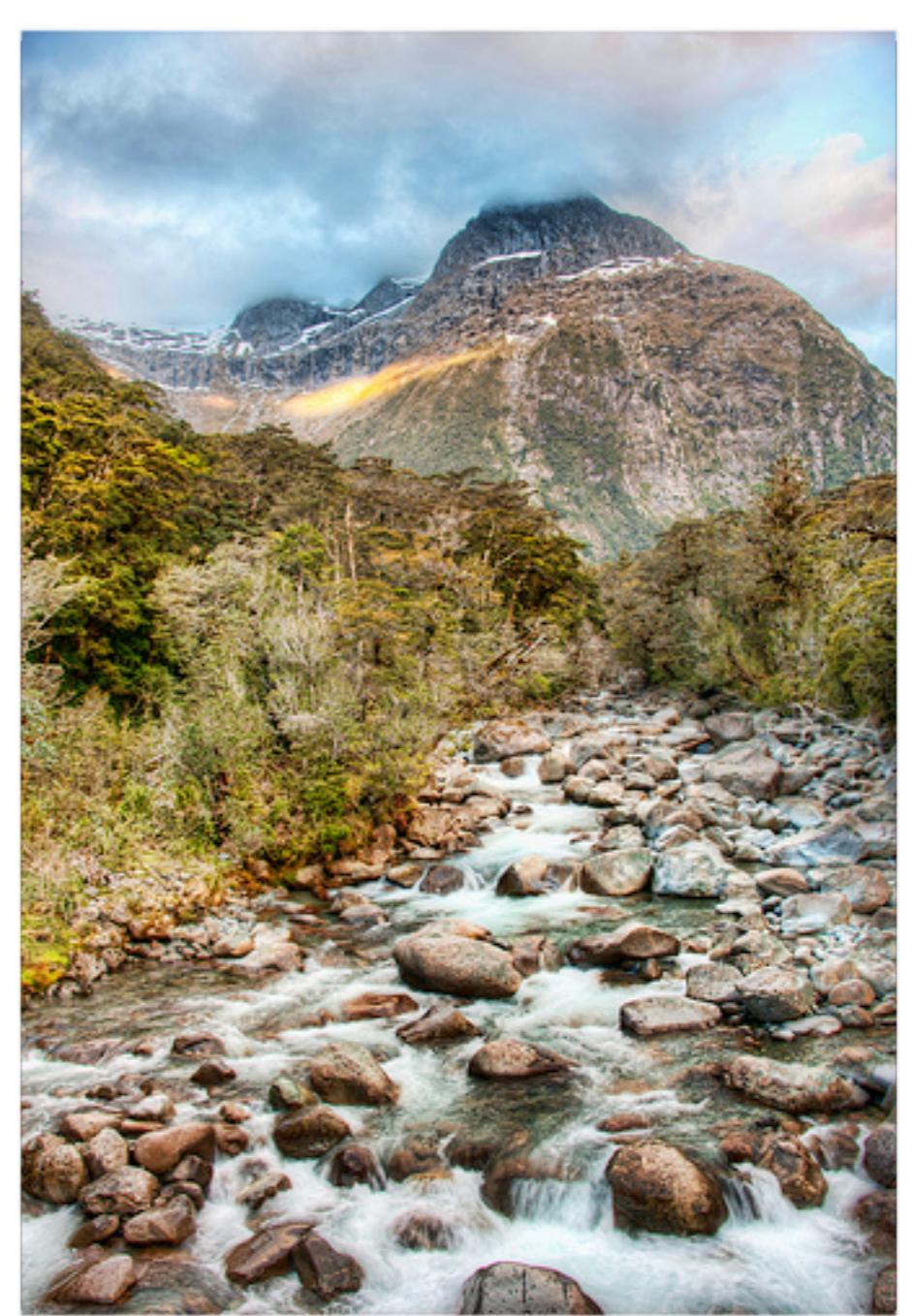
<http://raibledesigns.com>

@mraible

Images by Stuck in Customs - <http://www.flickr.com/photos/stuckincustoms>

Introductions

- Your experience with web development?
- Your experience with implementing security?
- Have you used Java EE 6, Spring Security or Apache Shiro?
- What do you want to get from this talk?



The background of the slide is a photograph of a coastal landscape at sunset. In the foreground, dark silhouettes of trees and a steep, green-covered hillside are visible. Beyond them, the ocean stretches to the horizon under a sky filled with soft, pastel-colored clouds transitioning from orange and yellow near the sun to blue and purple higher up.

Blogger on
raibledesigns.com

Father, Skier,
Cyclist

Founder of [AppFuse](http://appfuse.org)

**Web Framework
Connoisseur**

Who is **Matt Raible**?

Session Agenda

- Security Development
 - Java EE 6, Spring Security, Apache Shiro
 - SSL and Testing
- Verifying Security
 - OWASP Top 10
- Commercial Tools and Services
- Q and A

Develop

Penetrate

Protect

Relax

Why am I here?

- ❖ **Purpose**

- ❖ To learn more about Java webapp security and transform myself into a security expert.

- ❖ **Goals**

- ❖ Show how to implement Java webapp security.
- ❖ Show how to penetrate a Java webapp.
- ❖ Show how to fix vulnerabilities.

Develop



Dynamic Language Support?

- If it deploys on Tomcat, it has a web.xml.
- Grails
- JRuby on Rails
- Lift
- Play! Framework



Java EE 6

- Security constraints defined in web.xml
 - web resource collection - URLs and methods
 - authorization constraints - role names
 - user data constraint - HTTP or HTTPS
- User Realm defined by App Server
- Declarative or *Programmatic* Authentication
- Annotations Support





Java EE 6 Demo

<http://www.youtube.com/watch?v=8bXBGU7uo4o>

Servlet 3.0

- HttpServletRequest
 - authenticate(response)
 - login(user, pass)
 - logout()
 - getRemoteUser()
 - isUserInRole(name)



Servlet 3.0 and JSR 250

- Annotations
 - @ServletSecurity
 - @HttpMethodConstraint
 - @HttpConstraint
 - @RolesAllowed
 - @PermitAll
 - @DenyAll



Java EE Security Limitations

- No error messages for failed logins
- No Remember Me
- Container has to be configured
- Doesn't support regular expressions for URLs



Spring Security



- Filter defined in web.xml
- Separate security context file loaded by Spring
 - Defines URLs, Roles and Authentication Providers
 - Defines UserService (provided or custom)
- Password Encoding
- Remember Me



Spring Security Demo

<http://www.youtube.com/watch?v=poc5dylmbig>

Securing Methods

```
<global-method-security secured-annotations="enabled"/>
```

```
@Secured("IS_AUTHENTICATED_ANONYMOUSLY")
public Account readAccount(Long id);
```

```
@Secured("IS_AUTHENTICATED_ANONYMOUSLY")
public Account[] findAccounts();
```

```
@Secured("ROLE_TELLER")
public Account post(Account account, double amount);
```

```
<global-method-security jsr250-annotations="enabled"/>
```

Securing Methods 3.0

```
<global-method-security pre-post-annotations="enabled"/>
```

```
@PreAuthorize("isAnonymous()")  
public Account readAccount(Long id);
```

```
@PreAuthorize("isAnonymous()")  
public Account[] findAccounts();
```

```
@PreAuthorize("hasAuthority('ROLE_TELLER')")  
public Account post(Account account, double amount);
```

Spring Security Limitations

- Authentication mechanism in WAR
- Securing methods only works on Spring beans
- My remember me example doesn't work



Apache Shiro

- Filter defined in web.xml
- shiro.ini loaded from classpath
 - [main], [urls], [roles]
- Cryptography
- Session Management



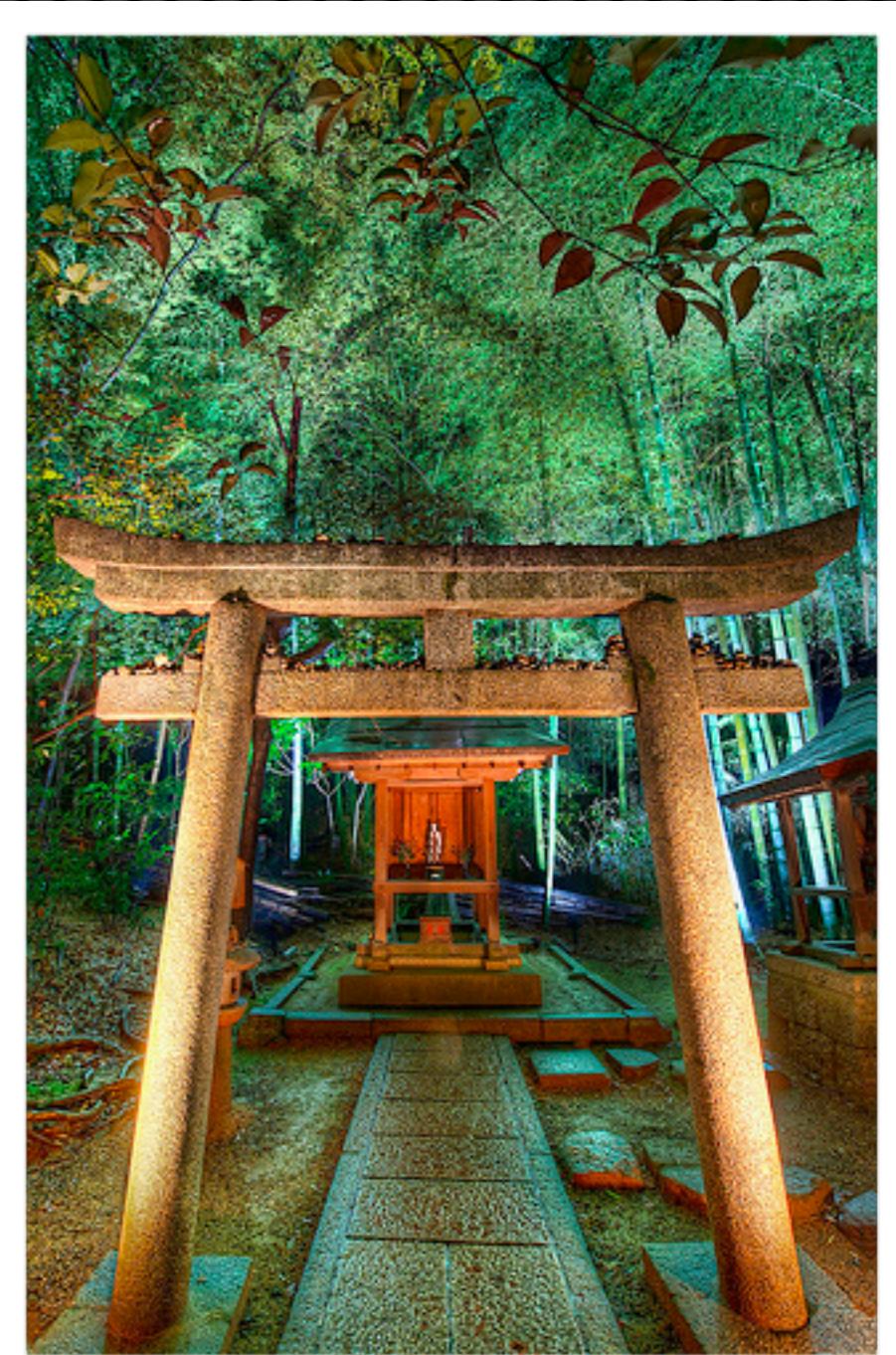


Apache Shiro Demo

<http://www.youtube.com/watch?v=YJByiDvOhsc>

Testing with SSL

- Cargo doesn't support http and https at same time
- Jetty and Tomcat plugins work for both
- Pass javax.net.ssl.trustStore & javax.net.ssl.trustStorePassword to maven-failsafe-plugin as <systemPropertyVariables>



Ajax Login

```
package org.appfuse.examples.webapp.security;

import javax.servlet.*;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class OptionsHeadersFilter implements Filter {

    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
        throws IOException, ServletException {
        HttpServletResponse response = (HttpServletResponse) res;

        response.setHeader("Access-Control-Allow-Origin", "http://" + req.getServerName());
        response.setHeader("Access-Control-Allow-Methods", "GET,POST");
        response.setHeader("Access-Control-Max-Age", "360");
        response.setHeader("Access-Control-Allow-Headers", "x-requested-with");
        response.setHeader("Access-Control-Allow-Credentials", "true");

        chain.doFilter(req, res);
    }

    public void init(FilterConfig filterConfig) {
    }

    public void destroy() {
    }
}
```

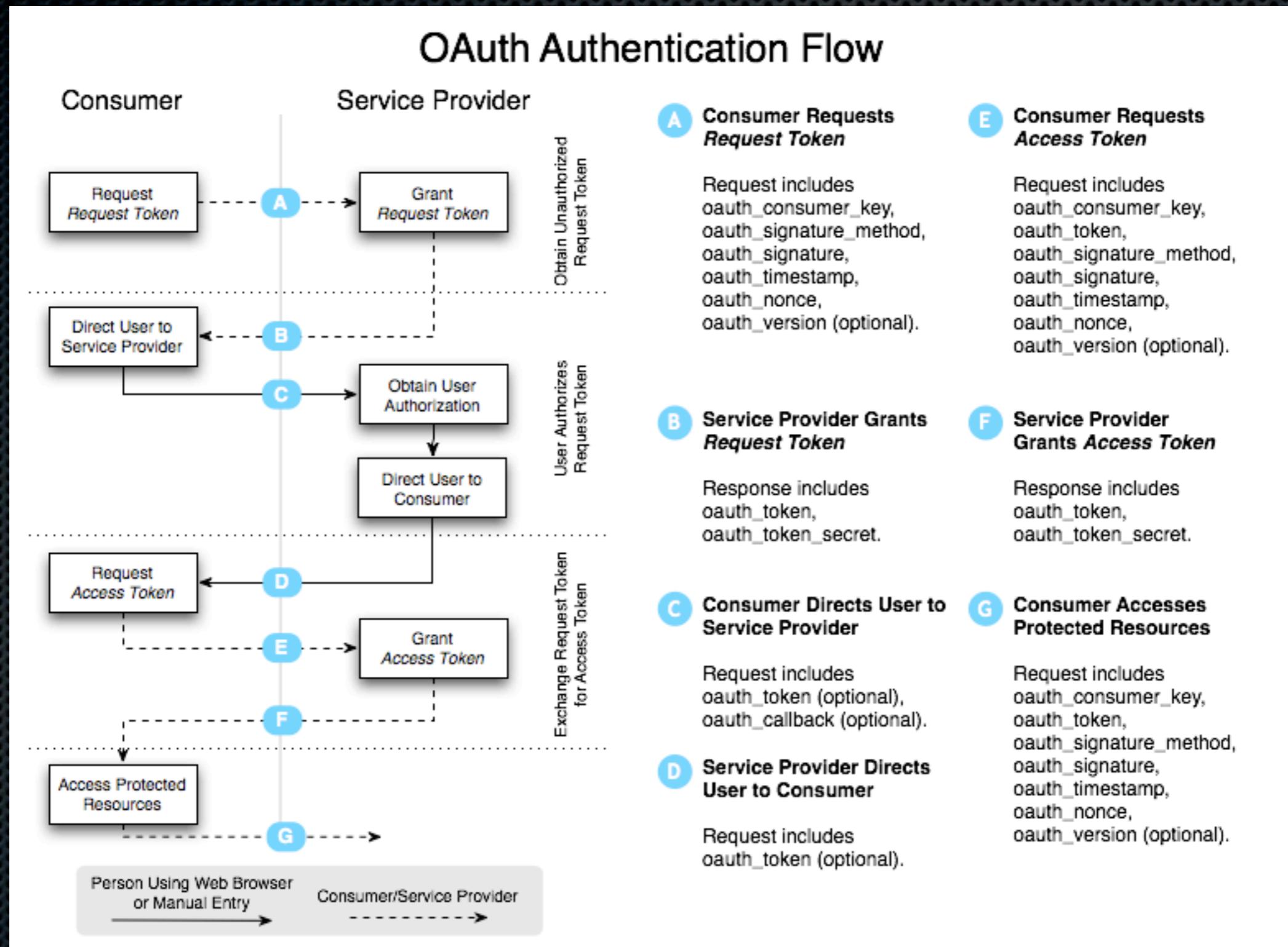
http://raibledesigns.com/rd/entry/implementing_ajax_authentication_using_jquery

Securing a REST API

- Use Basic or Form Authentication
- Use Developer Keys
- Use OAuth



OAuth





REST Security and OAuth Demo

http://raibledesigns.com/rd/entry/implementing_oauth_with_gwt

http://raibledesigns.com/rd/entry/grails_oauth_and_linkedin_apis

Penetrate

- OWASP Testing Guide and Code Review Guide
- OWASP Top 10
- OWASP Zed Attack Proxy
- Burp Suite
- OWASP WebGoat



OWASP

- The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. At OWASP you'll find free and open ...
- Application security tools, complete books, standard security controls and libraries, cutting edge research
- <http://www.owasp.org>



OWASP
The Open Web Application Security Project

OWASP Top 10 for 2010

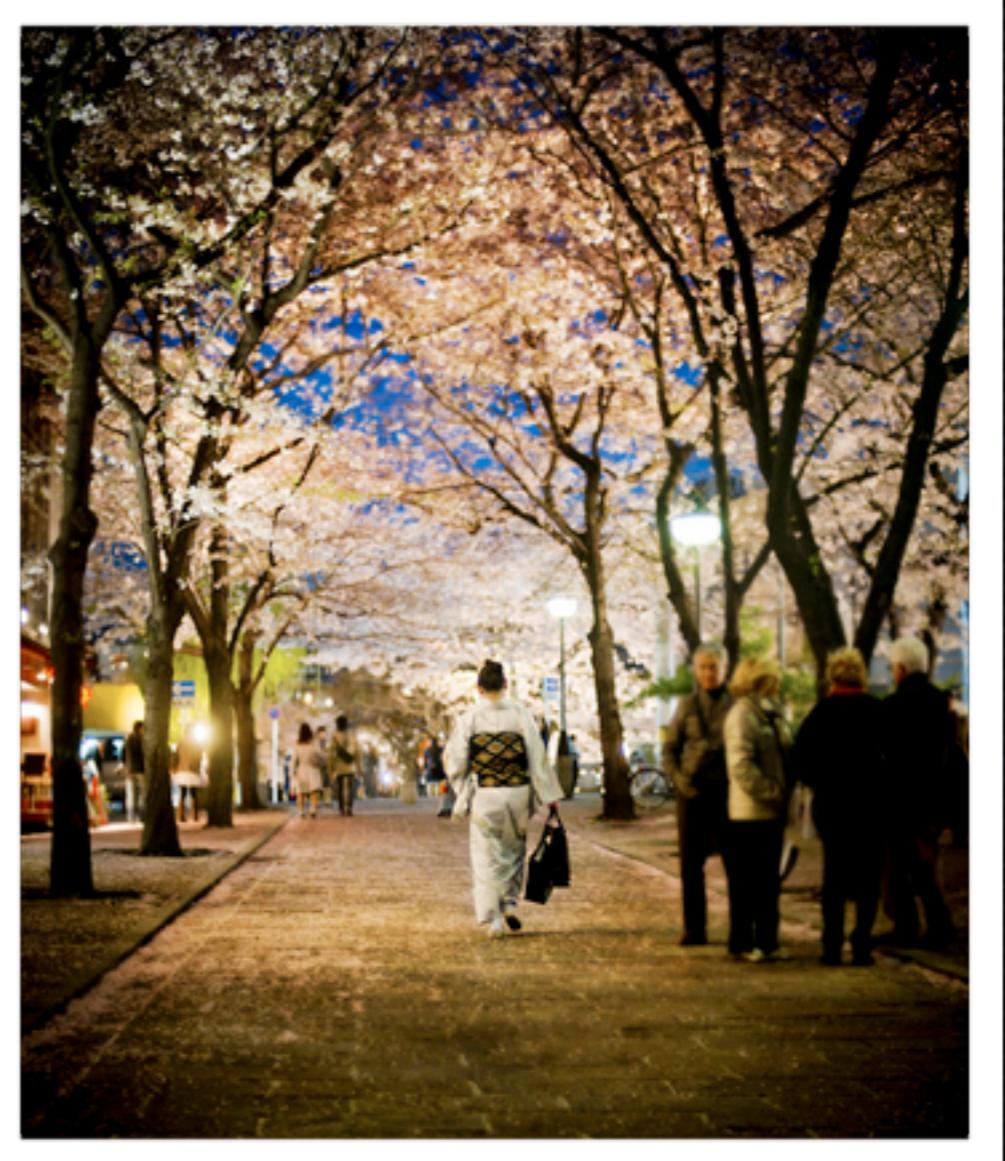
1. Injection
2. Cross-Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross-Site Request Forgery (CSRF)

OWASP Top 10 for 2010

6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

Protect

- Firewalls
- IDS and IDPs
- Audits
- Penetration Tests
- Code Reviews with Static Analysis Tools



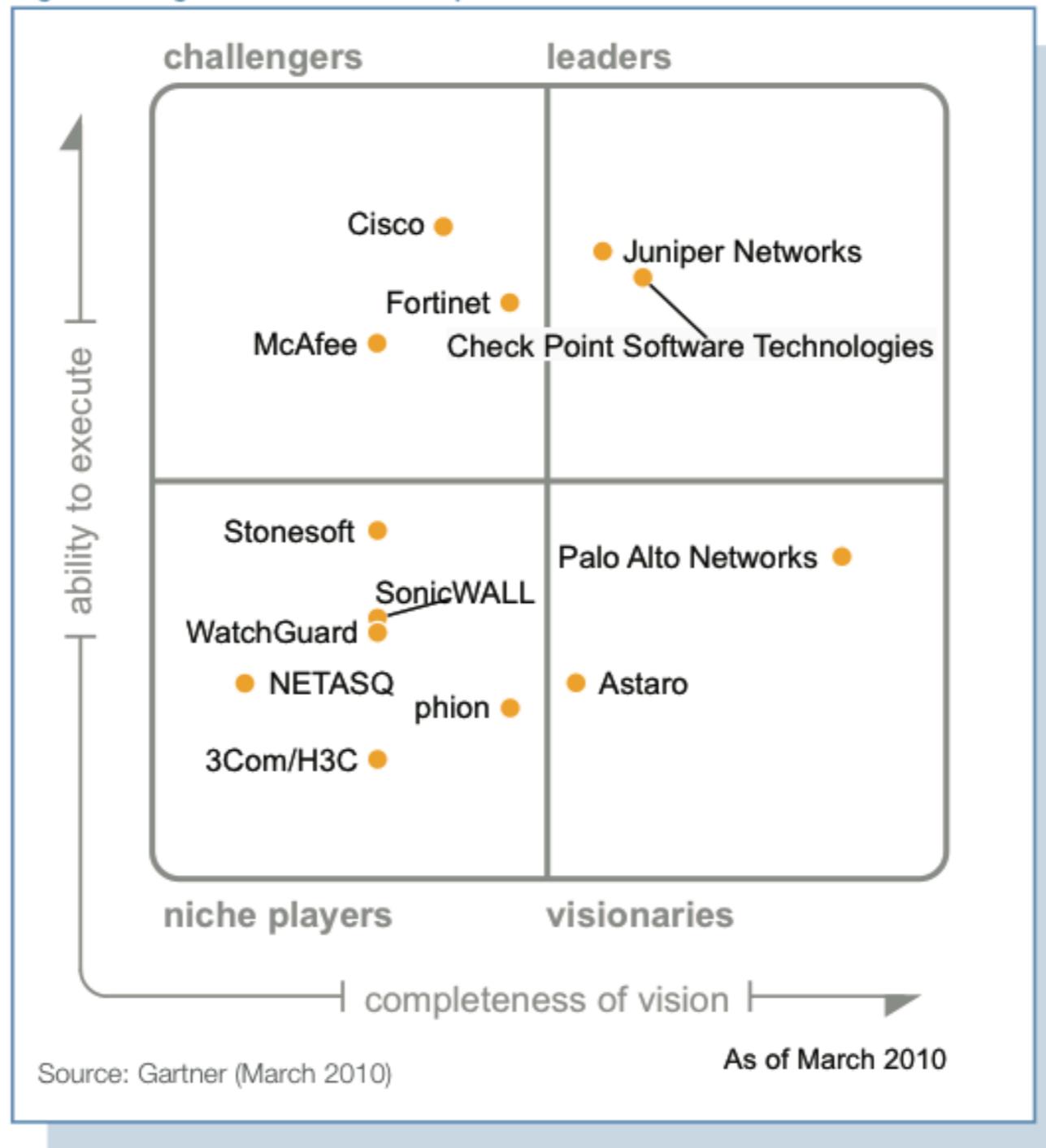
Firewalls

- Stateless Firewalls
- Stateful Firewalls
 - Invented by Nir Zuk at Check Point in the mid-90s
- Web App Firewalls
 - Inspired by the 1996 PHF CGI exploit
 - WAF Market \$234m in 2010



Gartner on Firewalls

Figure 1. Magic Quadrant for Enterprise Network Firewalls



Relax

- **Web App Firewalls:** Imperva, F5, Trustwave
 - **Open Source:** WebNight and ModSecurity
- **Stateful Firewalls:** Palo Alto, Juniper
- **IDP/IDS:** Sourcefire, TippingPoint
- **Audits:** ENY, PWC, Grant Thornton
- **Pen Testing:** Electric Alchemy, FishNet Security, Immunity
- **Static Analysis:** Fortify, Veracode

Questions?

Contact Information

<http://raibledesigns.com>

[@mraible](https://twitter.com/mraible)

My Presentations

<http://slideshare.net/mraible>

