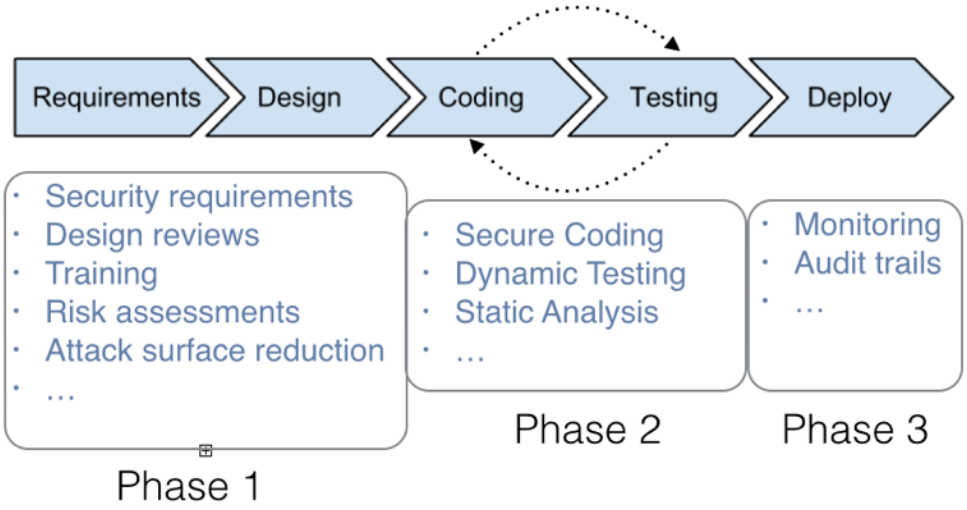
Tuesday, June 3, 2014

Leverage OpenSource to improve your security

Luca Carettoni - luca@addepar.com - @_ikki

Secure Software Development Lifecycle



Phase 1 - Requirements/Design

- Free On-Demand Training Courses
  - SAFECode - https://training.safecode.org/
- Insecure J2EE app
  - WebGoat - https://code.google.com/p/webgoat/
- Insecure Python app
- Gruyere - http://google-gruyere.appspot.com/
- Insecure PHP app
  - DVWA - http://www.dvwa.co.uk/
- Insecure Ruby app
  - Railsgoad - http://railsgoat.cktricky.com/
- Insecure Mobile app
  - ExploitMe Mobile - http://labs.securitycompass.com/exploit-me/
- Built-in security mechanisms and common pitfalls
  - http://blog.nibblesec.org/2014/04/on-web-frameworks-built-in-security.html
- How to choose a (secure) framework?
  - Evaluate all security features and default settings
  - Maturity of the project
  - Project popularity
  - Look for security advisories in Google, osvdb.org, etc

- Time To Patch statistics

- Community-based security initiatives
  - Google patch reward program - https://www.google.com/about/appsecurity/patch-rewards/
  - NodeSecurity - https://nodesecurity.io/
  - Mustache-Security - https://code.google.com/p/mustache-security/
  - TrueCrypt Audit - http://istruecryptauditedyet.com/

Phase 2 - Code/Testing

Don't reinvent the wheel, especially around security and crypto
- Validation APIs
  - E.g. AntiSamy - https://code.google.com/p/owaspantisamy/
- Secure functions
  - E.g. SafeCURL - https://github.com/fin1te/safecurl
- Logging
  - E.g. GELF Appenders - http://graylog2.org/gelf#libraries

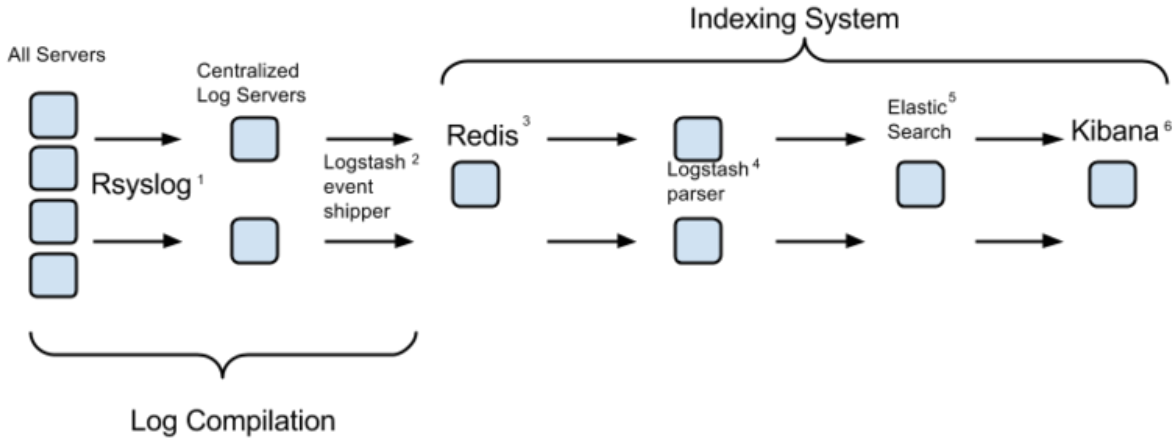
Maximize the home-field advantage
- Mix source code + dynamic testing
  - E.g. "Strategic" code reviews
    - Define critical areas of your codebase, setup automatic alerts, review all changes
  - E.g. "Continuous" semi-automatic security testing
    - Setup a dynamic web scanner to run for each candidate
    - Customize the scanner to detect previously discovered vulnerabilities
    - Evaluate the results
- Testing methodology
  - OWASP Testing Guide - https://www.owasp.org/index.php/OWASP_Testing_Project
- Web App Scanner
  - OWASP ZED Attack Proxy - https://code.google.com/p/zaproxy
  - Subgraph Vega - https://subgraph.com/vega/
  - Skipfish - https://code.google.com/p/skipfish/
  - IronWASP - https://ironwasp.org/
  - Burp Suite AppStore (OSS plugins) - https://pro.portswigger.net/bappstore/
- Libraries checker
  - OWASP Dependency-Check - https://github.com/jeremylong/DependencyCheck
  - Retire.js - http://bekk.github.io/retire.js/

Phase 3 - Deploy (and Maintain)

- Logging
  - Prevent #fails by:
    - Having a centralized logging mechanism
      - *syslog-ng/rsyslog, GELF plugins*

- Having backups
  - *tar, rsync, ssh, ...*
- Using the same timezone for all servers
  - *ntp*
- Aggregating system and application logs
  - *syslog-ng/rsyslog, GELF plugins*

- Full logging stack
  - Logstash, ElasticSearch, Kibana - http://www.elasticsearch.org/



- Attack surface monitoring
  - Continuous deployment requires continuous security
    1. Collect all public IPs for your infrastructure
       For AWS: boto, cli53
    2. Perform an Internet-facing portscan
       Nmap - http://nmap.org/
    3. Perform services enumeration
       Nmap NSE script - http://nmap.org/nsedoc/categories/default.html

http-title.nse

Script Output

```
Nmap scan report for scanme.nmap.org (74.207.244.221)
PORT   STATE SERVICE
80/tcp open  http
|_http-title: Go ahead and ScanMe!
```

ssl-cert.nse

Script Output

```
443/tcp open  https
| ssl-cert: Subject: commonName=www.paypal.com/organizationName=PayPal, Inc.\
/stateOrProvinceName=California/countryName=US
| Not valid before: 2011-03-23 00:00:00
|_Not valid after:  2013-04-01 23:59:59
```

sslv2.nse

Script Output

```
443/tcp open   https   syn-ack
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_IDEA_128_CBC_WITH_MD5
|     SSL2_RC2_CBC_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_CBC_128_CBC_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
```

4. Send to InfoSec, DevOps, ...
5. Sleep 10
6. Goto 1

- ModSecurity - https://www.modsecurity.org/
  - OWASP Core Rules - https://github.com/SpiderLabs/owasp-modsecurity-crs