RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Cloud Application Security Assessment, Guerrilla Style

SESSION ID: CSV-F03A

Mark Orlando
Director of Cyber Operations
Foreground Security

Adam Willard
Application Security Analyst
Foreground Security

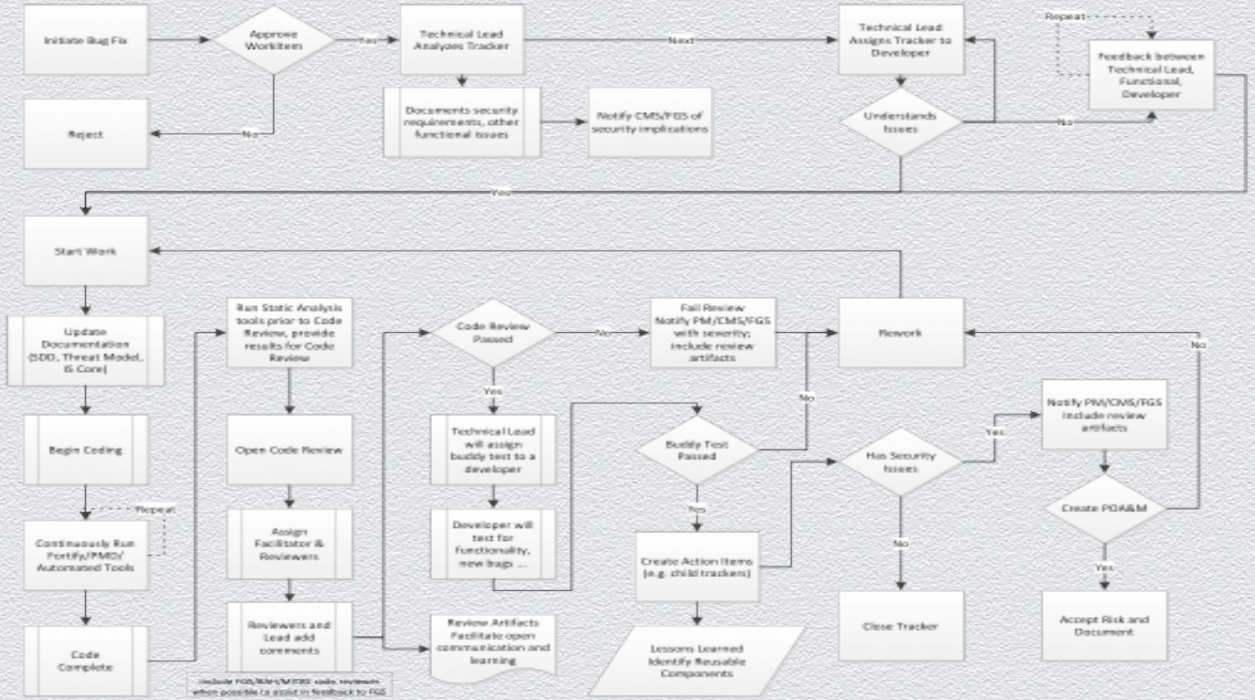
Agenda

- Challenge
- Potential Solutions
- Our Approach
- Results and Lessons Learned

Challenge

- Responsible for “securing” web applications that we:
  - Didn’t build
  - Don’t host
  - Can’t harden
  - Can’t disrupt via active (penetration) testing

Potential Solution: Create great processes



Potential Solution: Recommend standards

Potential Solution: Ask for full spectrum testing

- Many cloud service providers WILL let you conduct active testing
- Just need to understand limitations and try to build testing requirements into service agreement (d'oh!)

Our Goal

- DO NOT want to replace existing tools, code analysis, or automated scanning processes
- DO want to:
  - Facilitate ongoing discussions
  - Make a case for comprehensive security testing
  - Provide historical context
  - Tailor features and functionality to customer
  - Empower the analyst

Reconnaissance

- Understand the environment
  - XML Firewalls and Edge Devices
  - Load Balancers
  - Firewall Configurations and Logs aggregated
  - Security logs (SIEM)
  - Enterprise AV/IDS
- Exploratory testing via site walkthrough, documentation review, and interviews

Reconnaissance

- Gathering Host and IP data
  - What are your targets?
- Basic analysis of your requests
  - Begin mapping the systems based on simple analysis of headers and cookies

Static Analysis

- Involve security teams in the process
  - Every little bit helps
- Be a fly on the wall in developer meetings
  - You'll be amazed at how much information can be gathered regarding the health of a project.
  - Also helps identify breakdowns in communication
    - Developers are in the trenches and are usually vocal about what problems they are having if upper management is not in the meetings

3rd Party Components

- While it is important to review the code that is being developed, there is a reason for patches and zero days
- Read 3rd party CVE and Documentation carefully
  - Much of the documentation is based on simple implementations and are usually presenting live samples or default configurations
  - Plan to test those items
    - You'll be surprised what is installed by default as many teams install everything so they can get systems up and running quickly

Tools

- Burp
- w3af
- sqlmap
- Fiddler with Casaba Plugin
- Nmap/Zenmap
- Kali Linux
- Custom tools developed in house

Response Analysis

- Extract the Contents of the Response
  - Embedded Hyperlinks (review query string parameters)
  - Comments (html and JavaScript) –You may even find SQL
  - Forms
  - Hidden Fields
  - CSS
  - Iframes
  - JavaScript

Response Analysis

- Additional items to place into the tests
  - Test urls with parameters (GET) and switch to a POST
  - Test POST parameters as a GET
  - (simple tools can assist)
  - IP vs Hostname
- Identify shared resources
  - Look for the URLs constructs
  - May be an authentication mechanism

Response Analysis

- Weak Encoding
  - Look at the responses and compare the data that is displayed due to parameters
- Identify Cookies and their scope

Reporting Structure

- Understand reporting requirements (content and presentation)
- Automate as much as possible
- Run analysis as often as possible to continuously validate issues; remediation time is a useful metric

Our Results: One Project, Eight Months

- 30+ issues identified in commercial products
- 116 unique security findings for one major system alone
- More findings than external penetration testing or code analysis

Lessons Learned

- Developers/COTS embed details that can lead you to vulnerabilities
- Quality analysis and demonstration are key
- Keep your ear to the ground. Go to dev meetings!
- Manage expectations (yours and theirs)
- Work with ops
- Findings alone may mean little to business owners
- Test early and often

Lessons Learned

- Don’t reinvent the wheel
  - Obtain or develop a set of passive tools and a sustainable testing framework: continuous, low-intensity, multi-pronged. *Guerrilla style*.
- Demonstrate understanding and value
  - Seek to build trust with developers and system maintainers; this is part of “understanding the environment”
- Manage expectations
  - Testing, like the app itself, should evolve over time; this is why recurring testing and situational awareness is key

Try It Out

- Download Web Analyzer for free at http://foregroundsecurity.com/resources/tools
- Give us feedback!
  - awillard@foregroundsecurity.com
  - morlando@foregroundsecurity.com

RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Cloud Application Security Assessment, Guerilla Style