



OWASP

Application Security

Different approach to Information Security

Cristi Pascariu



Goals

- Vulnerability audit for large amount of code
- Priority based on risk
- “Spend as little time as possible to find as many vulnerabilities as possible”

Inspiration

- Finding o-day vulnerabilities in the ghetto
 - <http://trillworks.com/nick/2011/07/04/finding-o-day-vulnerabilities-in-the-ghetto/>
- The Art of Software Security Assessment (Identifying and Preventing Software Vulnerabilities)

Problems

- Large amounts of large code bases
- Reviewing in depth is costly
 - Is going to take a lifetime
 - Will miss half of the vulnerabilities
- What is the cost per vulnerability found ?
- What vulnerabilities have the higher risk ?
 - $\text{Risk}(\$) = \text{Impact}(\$) * \text{Likelihood of exploitation}$

Vulnerability classes and mitigation

Systematic vulnerabilities

- Framework
- Code review
- Manual/Automatic testing

Business/ design logic

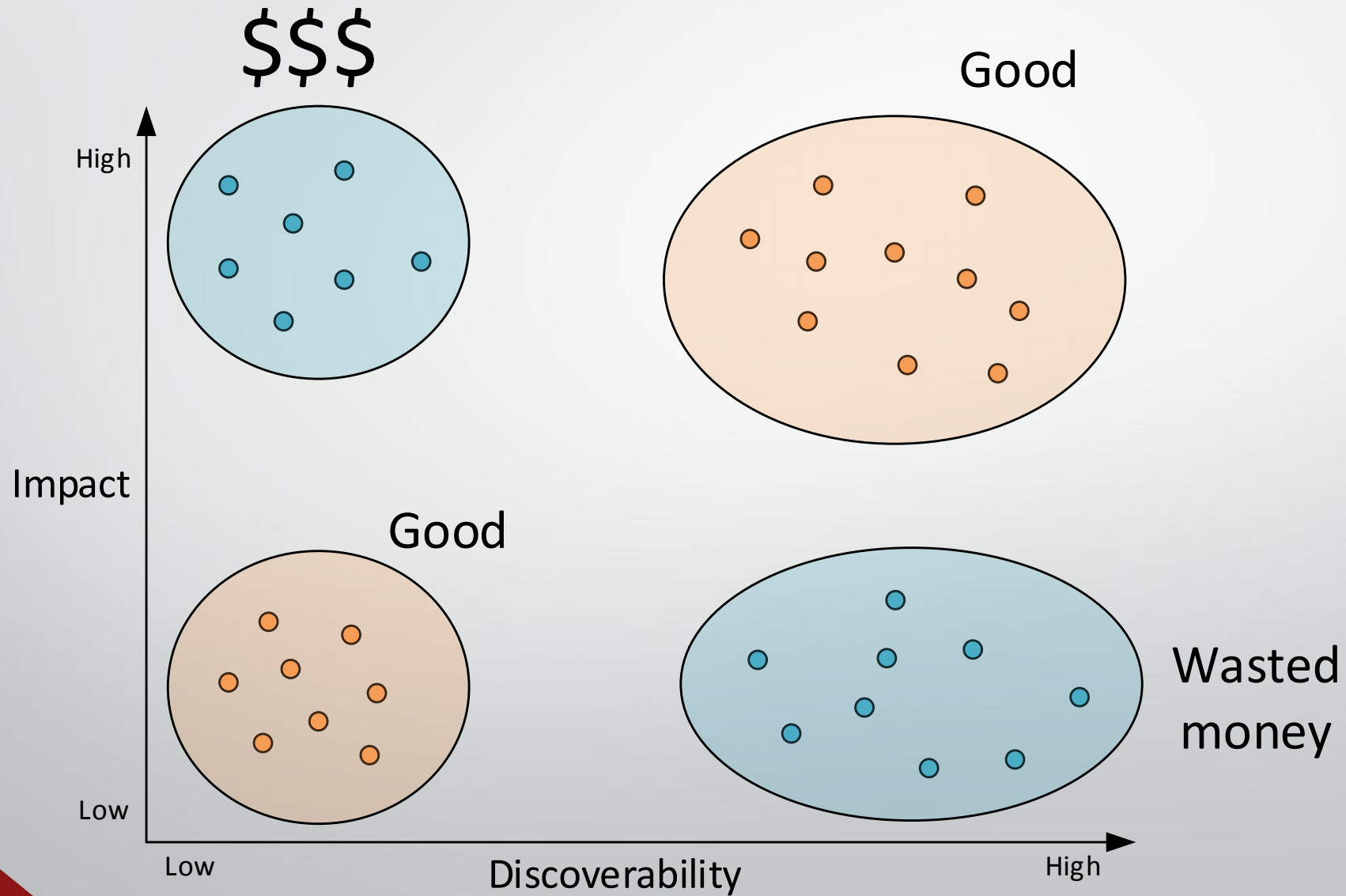
- Code review
- Manual testing



Out of scope

- Review business logic
- Pin-point accuracy of vulnerabilities

Audit

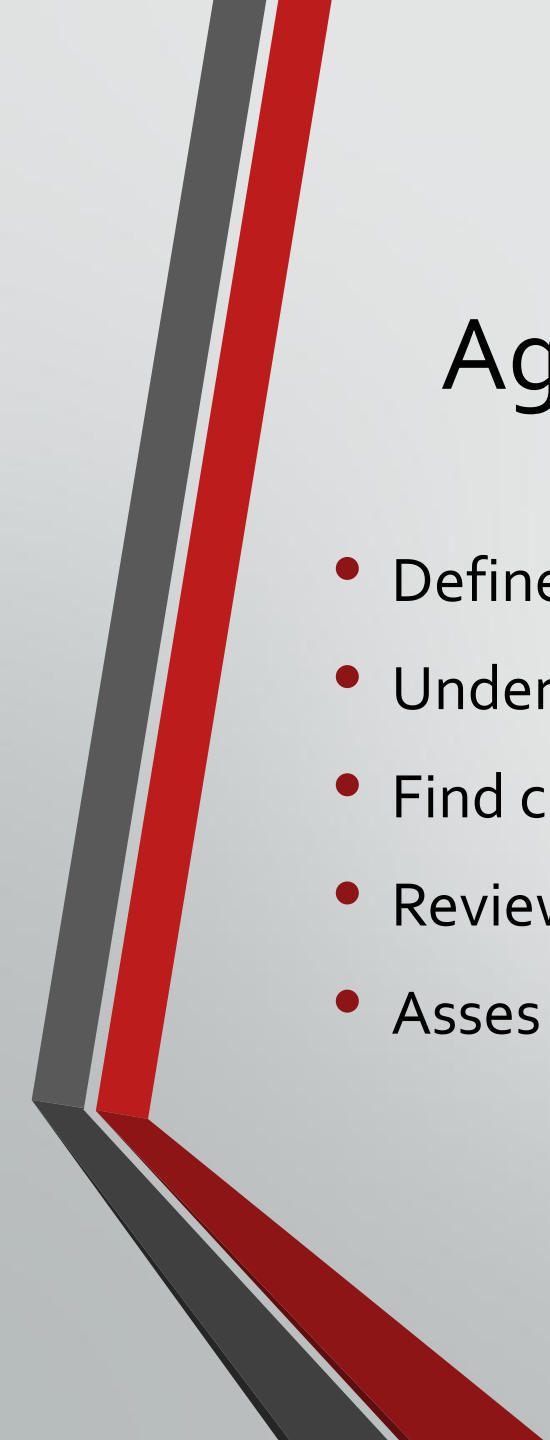


Candidate point strategy

- Fastest way to identify a lot of vulnerabilities
- Starts with identifying:
 - Points with side effects
 - Known vulnerable patterns
- Then back-tracing

Methodology

- All vulnerabilities found using this method
- All vulnerabilities submitted to Secunia SVCRP
 - <http://secunia.com/community/research/>



Agenda

- Define the target
- Understand the target
- Find critical/high-risk code paths to review
- Review and test
- Assess findings and iterate

Agenda

- Define the target
 - Find a target
 - In-house?
 - Public projects?
 - Obtain source code
- Understand the target
- Find critical/high-risk code paths to review
- Review and test
- Assess findings and iterate

Agenda

- Define the target
- Understand the target
 - Learn the language
 - Internalize OWASP top 10
 - Observe the framework and language
 - Dangerous functions
 - Mitigation techniques
 - Find commonly vulnerable code patterns
- Find critical/high-risk code paths to review
- Review and test
- Assess findings and iterate

Agenda

- Define the target
- Understand the target
- Find critical/high-risk code paths to review
 - Higher risk code paths is where you'll want to spend more time
 - Determine your critical functionality and assets
 - Examples might be:
 - System calls, DB access, File system access, Encoding, Cryptographic usage
 - These are the candidate points
- Review and test
- Assess findings and iterate

Agenda

- Define the target
- Understand the target
- Find critical/high-risk code paths to review
- Review and test
 - Start by casting the net wide on a project
 - As you learn more about it, start being more specific and reduce noise
 - Learn to review code at a glance
 - High risk vulnerabilities are usually easily seen at a glance
- Assess findings and iterate

Agenda

- Define the target
- Understand the target
- Find critical/high-risk code paths to review
- Review and test
- Assess findings and iterate
 - Steps to take
 - Assess risk
 - Do root cause analysis
 - Consider if there is likely to be more vulnerabilities of this type
 - Find out if there are steps that can be taken to mitigate the class of vulnerability at large
 - Find next steps to improve
 - Are you coming close to your risk tolerance ?
 - Are there still unknowns ?
 - Are there other higher-risk areas ?
 - Have you addressed the most discoverable bugs ?



Future challenges...

...Secure by default (but can't be assumed)

Example 1:

```
//this is the part that handles the actual recovery
if (isset($_GET['submitted']) && isset($_GET['loginName']) && isset($_GET['key'])) {
    //get the login name and key and verify if they match the ones in the database
    $query = $wpdb->get_results( "SELECT * FROM $wpdb->users WHERE user_login='".$_GET['loginName']."'");
    $dbValue = $query[0]->user_activation_key;
    $id = $query[0]->ID;
    $localHashValue = md5($_GET['loginName'].'RMPBP'.$id.'PWRCVR');
    if ($localHashValue == $_GET['key']) {
```

Example 1: Ctyptography

```
//this is the part that handles the actual recovery
if (isset($_GET['submitted']) && isset($_GET['loginName']) && isset($_GET['key'])) {
    //get the login name and key and verify if they match the ones in the database
    $query = $wpdb->get_results( "SELECT * FROM $wpdb->users WHERE user_login='".$_GET['loginName']."'");
    $dbValue = $query[0]->user_activation_key;
    $id = $query[0]->ID;
    $localHashValue = md5($_GET['loginName'].'RMPBP'.$id.'PWRCVR');
    if ($localHashValue == $_GET['key']) {
```

md5\s?\.(\.*\\$_(GET|POST|REQUEST)

Example 2:

```
require_once('../crayon_wp.class.php');  
crayon_die_if_not_php($_GET['wp_load'], 'wp-load');  
require_once($_GET['wp_load']);
```

Example 2: File inclusion

```
require_once('../crayon_wp.class.php');  
crayon_die_if_not_php($_GET['wp_load'], 'wp-load');  
require_once($_GET['wp_load']);
```

(include|require).*\\$_(POST|GET|REQUEST)

Example 2:

```
$video = $wpdb->get_row("SELECT * FROM " . $wpdb->prefix . "allvideogallery_videos WHERE id=" . $_GET['vid']);
```

Example 2: SQL Injection

```
$video = $wpdb->get_row("SELECT * FROM " . $wpdb->prefix . "allvideogallery_videos WHERE id=" . $_GET['vid']);
```

SELECT .* FROM .* \\$_(POST|GET|REQUEST)



The End

???