AnimateDead:
Debloating Web Applications via Concolic Execution

Babak AminAzad Rasoul Jahanshahi Chris Tsoukaladelis

Manuel Egele Nick Nikiforakis

Stony Brook University BOSTON UNIVERSITY

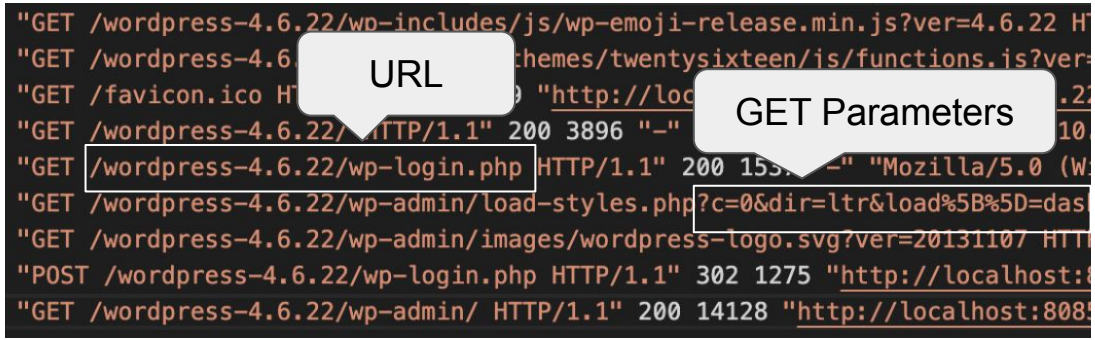
Motivation

- Previous debloating schemes rely on dynamic code coverage.
- This information is costly to collect
  - Runtime overhead of tracing.
  - Under-representation of error-inducing code paths and less popular features.
  - Dynamic code coverage is affected by database and network state

Build a PHP Emulator capable of concolic execution named *AnimateDead*

Use Existing Web Server Logs as Entry Points

- No instrumentation overhead using web server logs



Missing from the logs

- Post parameters
- File uploads
- Cookies
- Session variables

- Database
- Network

Mark them as Symbolic variables.

Concrete vs Symbolic execution

```
1.  $user_name = $_POST['user'];
2.  if (!isset($user_name)) {
3.      $redirect_to = login_url('Username not provided.');
4.  }
5.  else {
6.      $user = get_user_by_login($user_name);
7.      if (!$user && strpos($user_name, '@')) {
8.          $user = get_user_by_email($user_name);
9.      }
10.     if ($user) {
11.         $redirect_to = get_dashboard_url($user->ID);
12.     }
13.     else {
14.         $redirect_to = login_url('Invalid username.');
15.     }
16. }
17. wp_safe_redirect($redirect_to);
18. exit();
```

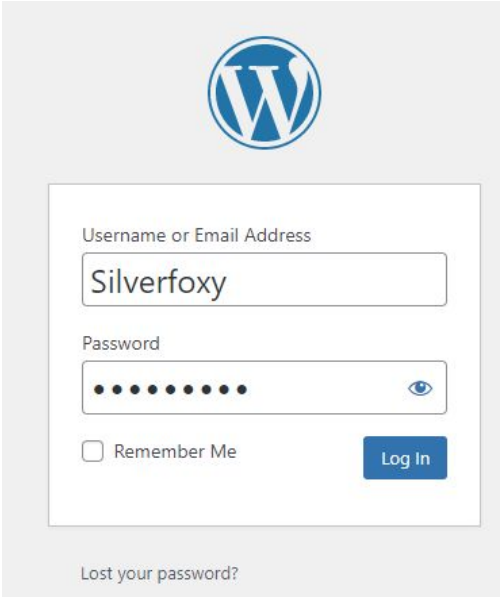
No username

Login w/ Username

Login w/ Email

Success

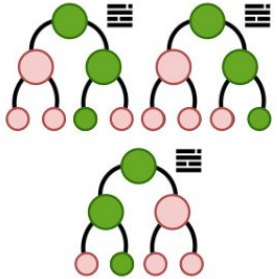
Username not found.



AnimateDead Overview

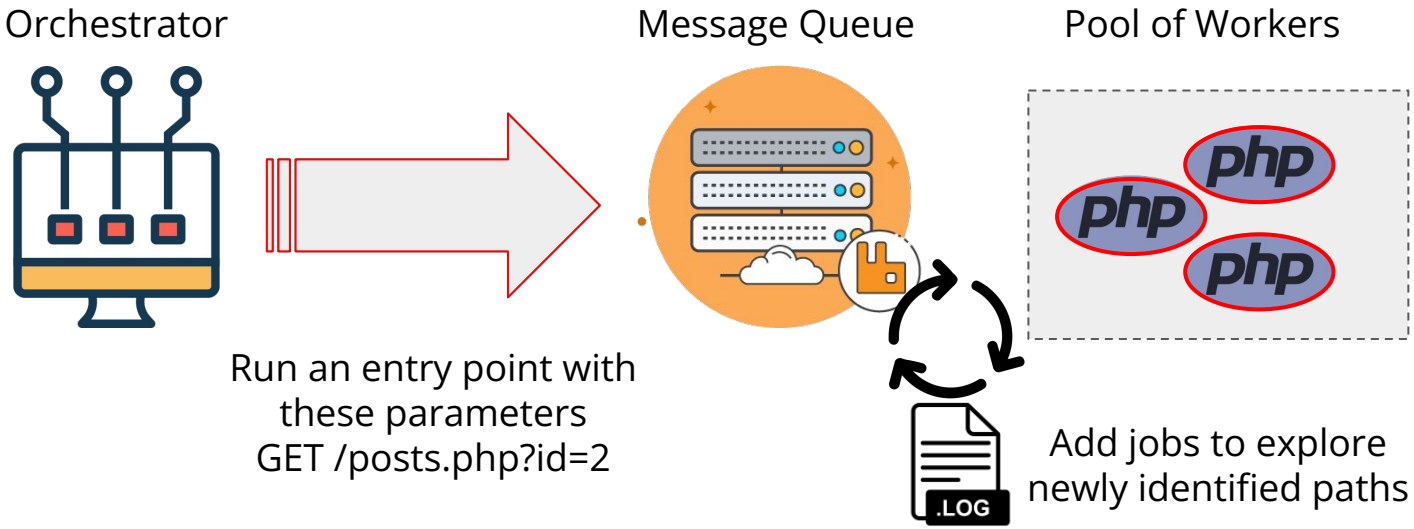
Translate log files to web application entry points

Reachability analysis from each entry point
<Using Concolic execution>



Debloat web applications by removing unreachable modules

Distributed Concolic Analysis



Emulation Replay

Reanimation logs

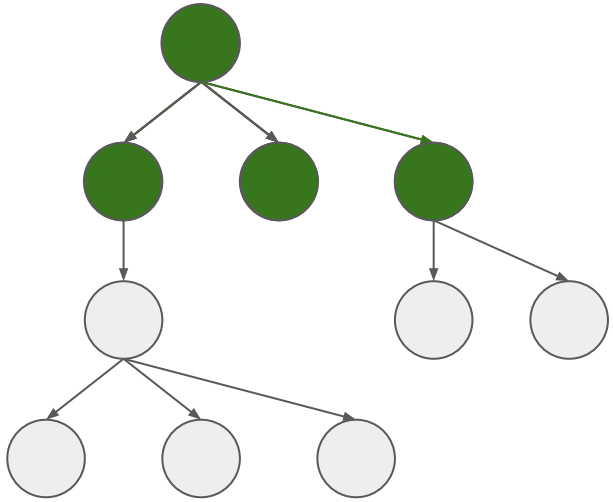
- Skip Line 1

[Explore Line 3]

- Skip Line 1
- Skip Line 3

[Explore Line 5]

```
1. if (isset($_POST['action']))
2.   …
3. elseif (isset($_POST['taskid']))
4.   …
5. else
6.   …
```



AnimateDead's Concolic Execution

- Concolic execution: Transition from Symbolic variables to Concrete ones.
- Required for instructions that change the structure of the call graph.
  - File inclusion (include, require, include_once, require_once)
  - Autoloader (new, static call, static property fetch)
  - Dynamic function call
  - Callbacks (*call_user_func*($cb), *preg_replace_callback*(/regex/, $cb))

Enabling the Concolic Transition

- **Type Tracking**
  - PHP APIs return types: *substr* ⇨ string, *isset* ⇨ boolean
  - $var *instanceof* ClassName
- **Value Set Analysis**
  - String operations to Regex:
    strncmp('pma', $cookie_name, 3) = 0 ⇨ /pma.*/
  - Set membership:
    in_array($_REQUEST['export_type'], *array*('sql', 'codegen', 'csv', ...))
- **Execution Environment (e.g., File system)**
  - *include* 'vendor/phpmyadmin/export/export_' . **$export_type** . '.php';

State Space Explosion

- State space explosion
  - Eliminate unsatisfiable paths to limit the total number of explored paths
  - Prioritize paths based on their likelihood to explore unique parts of the codebase
- 72% of symbolic conditions in PMA only check for presence of a variable
  - Include the presence or absence of HTTP parameters including POST, Cookie, File uploads.
- Efficient path selection
- Branch-coverage guided path prioritization
  - Must not get stuck in loops.
  - Maximize unique coverage.

Security metrics

Size Reduction

Web Application	*AnimateDead*	LIM
phpMyAdmin	▾69%	▾77%
WordPress	▾46%	▾50%
HotCRP	▾25%	▾40%
FluxBB	▾47%	▾53%

CVE Reduction

Web Application	*AnimateDead*	LIM
phpMyAdmin	▾65%	▾80%
WordPress	▾35%	▾35%

Conclusion

- Using concolic execution, we can perform reachability analysis for debloating.
- Debloating metrics for AnimateDead are comparable to dynamic debloating of Less is More.
- Source code and documents are available at https://debloating.com