Python代码审计连载之三：Server Side Request

笑然 / 2016-11-10 05:25:00 / 浏览数 2568 安全技术 漏洞分析 顶(0) 踩(0)

原文地址：https://www.cdxy.me/?p=744

CMS某些需求导致服务器主动向外发起请求，比如从外部URL添加资源：

目前很多成熟cms并不能有效控制该接口风险，我的第一个CVE就由它而来。

技术细节不再叙述，建议先阅读P神的文章，其中给出了非常棒的Python解决方案。

- 谈一谈如何在Python开发中拒绝SSRF漏洞

防御策略

1. 使用计数器确保30x跳转不会进入死循环
2. 归一化IP，防止畸形IP(8/10/16进制、省略格式)绕过防御规则。案例: WordPress <4.5 SSRF 分析
3. 白名单验证协议(http/https)和端口号(80/8080/443)
4. 黑名单验证IP是否属于内网
5. 跳转之后注意要继续进行2,3,4步的验证

Python风险点

Python开发中常用三种http请求方法(pycurl/urllib/requests)，这里简单分析其安全性及注意事项。

重定向

是否默认跟随重定向

- pycurl（不跟随）
- urllib/urllib2/requests（跟随）

默认最大重定向次数

- pycurl（未限制）
- urllib/urllib2（10次）
- requests（30次）

风险点：使用pycurl开启跟随跳转之后，需手动限制最大跳转次数。

```
c = pycurl.Curl()
c.setopt(pycurl.FOLLOWLOCATION, 1)
c.setopt(pycurl.MAXREDIRS, 5)
```

协议支持

urllib/urllib2/requests -> http/https/ftp 同时urllib/urllib2也支持`file:///etc/passwd`

pycurl支持更多：

```
supporting DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMT
```

风险点：未手动限定协议时可导致安全风险(如通过`file://`造成DoS)

运维风险点

一些CMS喜欢使用配置项来控制是否允许"与内网地址通讯"，使用时请注意关闭该配置以保证安全性。

测试方案

白盒

源码中定位到过滤函数，分析其逻辑，要特殊注意正则表达式的使用。

黑盒

1. fuzz协议(端口)
2. 测试全部内网IP地址段(畸形IP)是否被过滤
3. 测试重定向支持(循环重定向)
4. 测试域名解析到内网时是否被过滤
5. 综合2,3,4步，在重定向中测试域名解析到内网

辅助工具

搭建在公网服务器，提供 [循环重定向、延时响应、解析到内网IP、任意跳转] 四种功能，用于测试之前提到的安全风险，用法见代码。

- Github: ssrf_server.py

```
# !/usr/bin/env python
#  -*- coding: utf-8 -*-

"""
<br>  SSRF PoC Server
<br>
<br>  requirement:
<br>  --(Python 2.x + pip)
<br>  --pip install flask
<br>
<br>  usage:
<br>  --python ssrf_server.py
<br>
<br>  functions:
<br>  --infinite loop with time-delay:
<br>  ----/loop/[any-char]?sleep=[seconds]
<br>  --redirect:
<br>  ----/redirect/[count]?url=[destination]
<br>  --domain to ip:
<br>  ----/dns?ip=[IP]
<br>
<br>  example:
<br>  --infinite redirect loop with a 10-second-delay each time
<br>  ----http://yourhost:666/loop/xxx?sleep=10
<br>  --redirect 3 times and go to google.com finally
<br>  ----http://yourhost:666/redirect/3?url=https://www.google.com)
<br>  --redirect to a DOMAIN,and let the domain lead to 10.0.0.1
<br>  ----http://yourhost:666/dns?ip=10.0.0.1
<br>
<br>  author[mail:i@cdxy.me]
"""

import time
import random
import sys
from flask import Flask, request, render_template_string, redirect, session
from string import ascii_lowercase
```

```
SLEEP_ARG = 'sleep'
URL_ARG = 'url'
IP_ARG = 'ip'
JUMP_COUNT = 'count'


class Config():
   SECRET_KEY = '1426b50619e48fc6c558b6da16545d2e'
   debug = True


app = Flask(__name__)
app.config.from_object(Config)


def random_string(length=8):
   return ''.join([random.choice(ascii_lowercase) for _ in range(length)])


@app.route('/')
def index():
   return render_template_string(__doc__)


@app.route('/loop/<string:random>')
def loop(random):
   s = request.args.get(SLEEP_ARG)
   if s:
       time.sleep(int(s))
       return redirect('/loop/%s?%s=%s' % (random_string(), SLEEP_ARG, s))
   return redirect('/loop/%s' % random_string())


@app.route('/redirect/<int:count>')
def redirect_(count):
   c = count
   url = request.args.get(URL_ARG)
   if c:
       session[JUMP_COUNT] = c
       return redirect('/redirect/' + str(c - 1) + '?' + URL_ARG + '=' + url)
   else:
       return redirect(url)


@app.route('/dns')
def dns2ip():
   return redirect('http://www.%s.xip.io' % request.args.get(IP_ARG))


if __name__ == '__main__':
   if '-h' in sys.argv or '--help' in sys.argv:
       print __doc__
       sys.exit(0)
   app.run(host='0.0.0.0', port=666)
```


上一篇：Python代码审计连载之二：SSTI 下一篇：Python代码审计连载之四：Co...

1. 0 条回复

- 动动手指，沙发就是你的了！

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