诸子流
这短短的一生，我们最终都会失去；你不妨大胆一些，爱一个人，掌一座山，追一个梦。

昵称：诸子流
园龄：5年8个月
粉丝：394
关注：2
+加关注

< 2022年11月 >
日 一 二 三 四 五 六
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 1 2 3
4 5 6 7 8 9 10

搜索
找找看

积分与排名
积分 - 1006324
排名 - 304

随笔分类
Android(8)
Crazy Talk(4)
Eclipse(7)
Hadoop(2)
Java(12)
Kali(15)
Linux(99)
MySQL(12)
Oracle(11)
Pentest(44)
Python(94)
Reverse(8)
Tomcat(2)
Version Control(7)
VMware(5)
更多

随笔档案
2022年2月(1)
2022年1月(1)
2021年11月(1)
2021年10月(1)
2021年8月(1)
2021年4月(3)
2021年3月(2)
2020年12月(1)
2020年11月(1)
2020年10月(1)
2020年9月(3)
2020年8月(1)
2020年7月(3)
2020年6月(4)
2020年5月(1)
更多

阅读排行榜
1. Android Studio打包生成APK教程(177705)
2. PowerShell使用教程(157208)
3. Python3+WebSockets实现WebSocket通信(149616)
4. Wireshark使用教程(界面说明、捕获过滤器表达式、显示过滤器表达式)(121958)
5. zookeeper安装教程(zookeeper3.4.5为例)(83568)

推荐排行榜
1. PowerShell使用教程(30)
2. Wireshark使用教程(界面说明、捕获过滤器表达式、显示过滤器表达式)(18)
3. curl和wget的区别和使用(13)
4. Python3+WebSockets实现WebSocket通信(10)
5. PyCharm+QTDesigner+PyUIC使用教程(10)

最新评论
1. Re:Python3回撤通信编程实现教程(发送者+接收者)
感谢说明！
--ZHider
2. Re:PyCharm+Qt Designer+PyUIC安装配置教程
PyUIC 当参数主写的不对，就是argument 那项正确参数 -m PyQt5.uic.pyuic (FileName) -o (FileNameWithoutExtension).py 这个是...
--流煤兔125
3. Re:PyCharm+Qt Designer+PyUIC安装配置教程
@是星没别离 PyUIC 当参数主写的不对，就是argument 那项正确参数 -m PyQt5.uic.pyuic \(FileName\) -o \(FileNameWithoutExtensi...
--流煤兔125
4. Re:Python3+Robot Framework+RIDE安装使用教程
大神，能请教个问题吗，python3.7+robot framework-ride1.7.4.2+wxpython4.1.1, 使用if和else 经常多行输入，保存重新打开后所有else 消掉...
--ruanyang2019
5. Re:Python3列表(list)比较操作教程
大佬你好，请问单个变量怎么和列表比较 (确认列表有没有包含这个变量)，试了下issubset和issuperset貌似不行 = =
--Rainlane

HTTP Basic和Digest认证介绍与计算

一、说明

web用户认证，最开始是get提交+把用户名密码存放在客户端的cookie中的形式；在意识到这样不安全之后逐渐演变成了post提交+把用户凭证放到了服务端的session中的形式（当然sessionid还在cookie中）。

不过其实最初始http设计的认证方式，既不是“get+cookie”也不是“post+session”，而是Basic和Digest。但Basic和Digest并不流行我想主要是因为麻烦，一是说Basic和Digest使用的Authorization头并不会被浏览器自动发往服务器，二是说对于Digest计算很麻烦。

二、Basic认证形式

2.1 Basic认证请求示例

请求示例如下，主要是Authorization头（位置不重要，http头一般都不分先后）

```
GET /GetDeviceInfo HTTP/1.1
Host: 192.168.220.128
Authorization: Basic YWRtaW46MTIzNDU2
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept-Encoding: gzip, deflate
Accept: */*
Cache-Control: no-cache
Cookie: Secure
Connection: close
```

2.2 Basic认证计算方法

前边请求Authorization头的**YWRtaW46MTIzNDU2**，实际上是用户名admin密码123456使用以下计算方法得到：

```
base64(username:password)
```

Python计算代码如下：

```
import base64

def get_basic_authorization_header_value(username,password):
    # base64编码前后都(要)是字节码形式
    authorization_value = base64.b64encode((f"{username}:{password}").encode()).decode()
    authorization_header_value = f"Basic {authorization_value}"
    return authorization_header_value
```

三、Digest认证形式

3.1 Digest认证请求示例

```
GET /GetDeviceInfo HTTP/1.1
Host: 192.168.220.128
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:60.0) Gecko/20100101 Firefox/60.0
Authorization: Digest username="admin",realm="TVT API Test Tool",nonce="d4f95e85dc5a39a4914db61b67878f5b",uri="GetDeviceInfo",algorithm="MD5",cnonce="d4f95e85dc5a39a4914db61b67878f5b",nc=00000001,qop="auth",response="1cc4cf126d3c4a70d2de34c5d8c2943c"
Accept-Encoding: gzip, deflate
Accept: */*
Cache-Control: no-cache
Cookie: Secure
Connection: close
```

username----系统用户名；客户端自行填充

realm----领域：服务端通过WWW-Authenticate头返回内容可以自己随便定，但其目的是用于提示客户端当前是什么系统，所以规范来说应类似于“myhost@testrealm.com”的形式。

nonce----服务端通过WWW-Authenticate头返回的随机数

uri----请求接口或资源（似乎规范来说应用GET或POST后的一样，上边例子中少了/是因为服务端没按规范实现）

algorithm----后边response用的计算方法

cnonce----client nonce，客户端生成的随机数

nc----nonce count，用于标识进行请求的次数。（但你一直不变服务端也不会管你对不对）

qop----quality of protection，进一步限定response的计算方法，服务端通过WWW-Authenticate头返回。

response----认证最主要的值，前面各字段除algorithm外全要参与该值的计算。

3.2 Digest认证计算方法

在最开始的RFC 2069中规定response计算方法如下：

```
HA1 = MD5(username:realm:password)
HA2 = MD5(method:uri)
response = MD5(HA1:nonce:HA2)
```

随后的RFC 2617对计算方法进行了增强，规定计算方法如下（当algorithm值为MD5或未指定、qop未指定时等同RFC 2069）：

```
# HA1部分
# 当algorithm值为"MD5"或未指定时，HA1计算方法如下
HA1 = MD5(username:realm:password)
# 当algorithm值为"MD5-sess"时，HA1计算方法如下
HA1 = MD5(MD5(username:realm:password):nonce:cnonce)

# HA2部分
# 当qop值为"auth"或未指定时，HA2计算方法如下
HA2 = MD5(method:uri)
# 当qop值为"auth-int"时，HA2计算方法如下:entityBody是指整个body(?)
HA2 = MD5(method:uri:MD5(entityBody))

# response部分
# 当qop值为"auth"或"auth-int"时，response计算方法如下
response = MD5(HA1:nonce:nonceCount:cnonce:qop:HA2)
# 当qop未指定时，response计算方法如下
response = MD5(HA1:nonce:HA2)
```

Python计算代码如下：

```
import hashlib

# body初始值不要是None，不然下边encode时会报错
def get_basic_authorization_header_value(username, password, uri, method, realm, nonce, nc, cnonce, algorithm=None, qop=None, body=""):
    response_value = calc_digest_response_value(username, password, uri, method, realm, nonce, nc, cnonce, algorithm, qop, body)
    authorization_header_value = f'Digest username="{username}",realm="{realm}",nonce="{nonce}",uri="{uri}",algorithm="{algorithm}",cnonce="{cnonce}",nc={nc},qop="{qop}",response="{response_value}"'
    return authorization_header_value

def calc_digest_response_value(username, password, uri, method, realm, nonce, nc, cnonce, algorithm=None, qop=None, body=""):
    # HA1部分
    # 当algorithm值为"MD5"或未指定时，HA1计算方法如下
    if algorithm == "MD5" or algorithm == "" or algorithm is None:
        HA1 = hashlib.md5((f"{username}:{realm}:{password}").encode()).hexdigest()
    # 当algorithm值为"MD5-sess"时，HA1计算方法如下
    elif algorithm == "MD5-sess":
        HA1 = hashlib.md5((f"{username}:{realm}:{password}").encode()).hexdigest()
        HA1 = hashlib.md5((f"{HA1}:{nonce}:{cnonce}").encode()).hexdigest()
    else:
        response_value = '"the value of algorithm must be one of "MD5"/"MD5-sess"/""/None'
        return response_value

    # HA2部分
    # 当qop值为"auth"或未指定时，HA2计算方法如下
    if qop == "auth" or qop == "" or qop is None:
        HA2 = hashlib.md5((f"{method}:{uri}").encode()).hexdigest()
    # 当qop值为"auth-int"时，HA2计算方法如下:entityBody是指整个body我其实不太确定
    elif qop == "auth-int":
        HA2 = hashlib.md5((f"{body}").encode()).hexdigest()
        HA2 = hashlib.md5((f"{method}:{uri}:{HA2}").encode()).hexdigest()
    else:
        response_value = '"the value of qop must be one of "auth"/"auth-int"/""/None'
        return response_value

    # response部分
    # 当qop值为"auth"或"auth-int"时，response计算方法如下
    if qop == "auth" or qop == "auth-int":
        response_value = hashlib.md5((f"{HA1}:{nonce}:{nc}:{cnonce}:{qop}:{HA2}").encode()).hexdigest()
    # 当qop未指定时，response计算方法如下
    elif qop == "" or qop is None:
        response_value = hashlib.md5((f"{HA1}:{nonce}:{HA2}").encode()).hexdigest()
    else:
        response_value = "unknown error"
    return response_value
```

参考：

https://en.wikipedia.org/wiki/Digest_access_authentication

https://tools.ietf.org/html/rfc2069

https://tools.ietf.org/html/rfc2617

分类: Python

好文要顶 关注我 收藏该文

诸子流
粉丝 - 394 关注 - 2
+加关注

« 上一篇：Python3浮点型（float）运算结果不正确处理办法
» 下一篇：华为S5700配置端口镜像和华三S5120配置802.1X认证记录

0 推荐 0 反对

posted on 2019-03-29 17:52 诸子流 阅读(2191) 评论(0) 编辑 收藏 举报

刷新评论 刷新页面 返回顶部