# 周报

本周项目任务过于繁重，所以我将以前两周内容重新复习了一遍，那么这次周报以复习内容为主，包含渗透

## 1.复习Linux课程以及python课程

**Linux：**

搭建操作环境，将centos远程连接Linux系统，学习了虚拟机快照，Linux目录结构以及Linux命令基础。

快照的作用：快照可以保存虚拟机的状态，当虚拟机出现问题的时候，可以通过预先制作的快照恢复到制作时候的状态，用作备份用。

Linux操作系统的目录结构：

Linux只有一个顶级目录，称之为：根目录（用/表示）

Windows系统有多个顶级目录，即各个盘符。

ls命令的作用：在命令行当中，以平埔的形式，展示当前工作目录的内容。

什么是当前工作目录?

Linux命令行在执行命令的时候需要一个工作目录，打开命令行程序默认设置工作目录在用户的HOME目录。

**Python方面：**

学习了变量命名规则、基本数据类型（int、float、str、bool）。

条件语句（if-elif-else）、逻辑运算符（and/or/not）。

学习了-ls命令的参数和选项，-cd-pwd命令，以及相对路径绝对路径和特殊路径符。

ls命令的参数的作用：可以指定要查看的文件夹的内容

ls命令的选项：-a选项：可以展示出隐藏的内容

-l选项：以列表的形式展示内容，并展示更多细节

-h选项：与-l选项搭配使用，显示文件的大小单位。表示上一级目录

cd命令的作用：cd命令可以切换当前工作目录（没有选项，只有参数，表示目标路径）。

pwd命令的作用：输出当前所在的工作目录（没有选项，没有参数，直接使用即可）。

相对路径和绝对路径

特殊路径符：.表示当前目录

..表示上一级目录

~表示用户的HOME目录

for循环与while循环的区别与应用场景。

列表（增删改查）、字典（键值对操作）、元组（不可变性）。

函数定义与参数传递（位置参数、默认参数）。

**IP地址的详解：**

| | |
|---|---|
| IPv4地址段: | 42.186.0.0 – 42.186.255.255 |
| 网络名称: | Netease-Network |
| 单位描述: | Guangzhou NetEase Computer System Co., Ltd |
| 管理联系人: | ZX3316-AP |
| 技术联系人: | ZX3316-AP |
| 国家代码: | CN |
| 地址状态: | ALLOCATED PORTABLE |
| 维护账号: | MAINT-CNNIC-AP |
| 次级维护帐号: | MAINT-CNNIC-AP |
| 事件响应账号: | IRT-CNNIC-CN |
| 路由维护帐号: | MAINT-CNNIC-AP |
| 最后修改记录: | 2016-06-20T05:52:01Z |
| 数据来源: | APNIC |
| | |
| IPv4地址段: | 123.58.160.0 – 123.58.191.255 |
| 网络名称: | Netease-Network |
| 单位描述: | Guangzhou NetEase Computer System Co., Ltd. |
| 单位描述: | NetEase Building No.16 Ke Yun Road, Zhong Shan Avenue, |
| 单位描述: | Guangzhou, P.R. China |
| 国家代码: | CN |
| 管理联系人: | AUTO1-FW |
| 技术联系人: | AUTO1-FW |
| 维护账号: | MAINT-AP-CNISP |
| 事件响应账号: | IRT-CNISP-CN |
| 地址状态: | ALLOCATED NON-PORTABLE |
| 最后修改记录: | 2015-09-08T01:35:27Z |
| 数据来源: | APNIC |
| | |
| IPv4地址段: | 114.113.216.0 – 114.113.219.255 |

比如说这个IP地址是通过扫IP的结果

我们需要做的工作是想办法及时发现我们当前扫描的IP存在waf，记录并跳过此IP，继续下一个IP的扫描。
如果我们等待masscan扫描这个IP全端口结束，再去判断端口开放数量是否异常是需要比较久的时间，这里我们可以设定首先一个异常数值，并使用subprocess监视masscan运行时打印出来的当前开放端口数，当监视到的当前开放端口数超过我们设置的异常数值时，也就意味着该IP服务器应该存在waf，直接break进入下一个循环。
这里演示这个 59.111.14.159 这个IP，，sudo masscan 59.111.14.159 -p1-65535 --rate2000，我们可以发现开放端口数量异常，那么这里就是我们需要舍弃的IP，在python中大概代码如下所示：

```
import re
import os
import sys
import click
import subprocess
import threading

limitNumber = 80

lock = threading.Lock()

command = 'masscan 59.111.14.159 -p1-65535 --rate 2000'
child = subprocess.Popen(command,stdout=subprocess.PIPE,stderr=subprocess.STDOUT,shell=True)
while child.poll() is None:
    output = child.stdout.readline()
    line = str(output, encoding='utf-8').strip()
    if 'found=' in line:
        lock.acquire()
        print(line)
        lock.release()
        foundNumber = re.findall(r'found=(\d{1,5})', line)
        if int(foundNumber[-1]) > int(limitNumber):
            os.kill(child.pid, 9)
            print('疑似有WAF!存活端口'+ str(foundNumber[-1]) +'个')
```

这里还有个坑，如果masscan加上-oX-oJ等输出参数的话，是捕获不到打印的内容，因此我这里没有使用输出参数保存扫描结果，而是保存masscan打印的所有内容，最后使用正则提取开放的端口。当然这里也可以通过修改masscan的源码，重新编译使用，
masscan结束后就到了nmap，在nmap中，我会使用-sV-sT-Pn--version-all --open 参数
-sV 是用来识别端口服务的。
-sT 其实在速度上并不如半开放扫描-sS，因为他需要完成整个握手包的，而且使用-sT目标主机会记录大量日志，-sT唯一的优势就是使用这个参数不需要root权限，普通用户权限就可以使用
而-sS是需要root权限。-Pn就是跳过发现主机过程，nmap在扫描之前会首先探测主机是否开放，发现方式就是ping，因为我们之前已经使用了masscan识别到了开放端口，也就意味着主机是存活的，我们也就没有必要再次去nmap去探测主机是否开放。
--open 是限定只探测状态为open的端口--version-all 会对每个端口尝试每个探测报文，在测试时候经常会碰到端口服务识别不出来，可以尝试使用这个参数说不定可以解决，因为他会对每个端口进行每一种类型，保证最大准确率。
端口识别出来后，当然就是针对每一种开放服务进行相应的安全检测，这些大同小异。

以上内容包括渗透内容，主要针对于Linux和PYthon的渗透