二、文本处理

Linux黑客基础-27-文本处理-01-文本的查看和过滤

Tex t Manipulat ion 文本文件

一切皆文件

snort，NIDS（网络入侵检测系统）https://www.snort.org/

现在被思科（Cisco）收购

案例：snort的配置文件的处理

安装snort：apt-get install snort snort-doc

验证：dpkg -l | grep snort

找到配置文件：/etc/snort/snort.conf

查看文件

1、（cat）

```
┌──(root💀kali)-[/etc/snort]
└─# cat snort.conf
```

2、（head）查看文件的头部

默认显示文件的前10行

```
┌──(root💀kali)-[~/桌面/work/doc]
└─# head snort.conf            查看前10行
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                   Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-sigs@lists.sourceforge.net
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
┌──(root💀kali)-[~/桌面/work/doc]
└─# head -200 snort.conf      显示文件的指定行数
```

3、（tail）显示文件的尾部

```
┌──(root💀kali)-[~/桌面/work/doc]
└─# tail snort.conf
# include $SO_RULE_PATH/smtp.rules
# include $SO_RULE_PATH/snmp.rules
# include $SO_RULE_PATH/specific-threats.rules
# include $SO_RULE_PATH/web-activex.rules
# include $SO_RULE_PATH/web-client.rules
# include $SO_RULE_PATH/web-iis.rules
# include $SO_RULE_PATH/web-misc.rules

# Event thresholding or suppression commands. See threshold.conf
include threshold.conf
```

```
-f, --follow[={name|descriptor}]          随文件增长即时输出新增数据
tail +20 number.txt | more      从第20行开始显示
tail -20 number.txt | more      显示尾部的后20行
```

4、（nl）显示文件的行号

Plain Text

```
┌──(root💀 kali)-[~/桌面/work/doc]
└─# nl /etc/snort/snort.conf
                          148 × 1 ⚙
     1  #--------------------------------------------------
     2  #   VRT Rule Packages Snort.conf
     3  #
     4  #   For more information visit us at:
     5  #     http://www.snort.org                   Snort Website
     6  #     http://vrt-blog.snort.org/    Sourcefire VRT Blog
     7  #
     8  #     Mailing list Contact:      snort-sigs@lists.sourceforge.net
     9  #     False Positive reports:    fp@sourcefire.com
    10  #     Snort bugs:                bugs@snort.org
    11  #
    12  #     Compatible with Snort Versions:
    13  #     VERSIONS : 2.9.7.0
    14  #
    15  #     Snort build options:
    16  #     OPTIONS : --enable-gre --enable-mpls --enable-targetbased -
--enable-ppm --enable-perfprofiling --enable-zlib --enable-active-respons
e --enable-normalizer --enable-reload --enable-react --enable-flexresp3
    17  #
    18  #     Additional information:
    19  #     This configuration file enables active response, to run sno
rt in
    20  #     test mode -T you are required to supply an interface -i <in
terface>
    21  #     or test mode will fail to fully validate the configuration
and
    22  #     exit with a FATAL error
    23  #--------------------------------------------------

    24  ##################################################
    25  # This file contains a sample snort configuration.
    26  # You should take the following steps to create your own custom c
onfiguration:
    27  #
    28  #  1) Set the network variables.
    29  #  2) Configure the decoder
    30  #  3) Configure the base detection engine
    31  #  4) Configure dynamic loaded libraries
    32  #  5) Configure preprocessors
    33  #  6) Configure output plugins
    34  #  7) Customize your rule set
```

```
38      35  #  8) Customize preprocessor and decoder rule set
39      36  #  9) Customize shared object rule set
40      37  ##################################################
41
42      38  ##################################################
43      39  # Step #1: Set the network variables.  For more information, see
    README.variables
44      40  ##################################################
45
46      41  # Setup the network addresses you are protecting
47      42  #
48      43  # Note to Debian users: this value is overriden when starting
49      44  # up the Snort daemon through the init.d script by the
50      45  # value of DEBIAN_SNORT_HOME_NET s defined in the
51      46  # /etc/snort/snort.debian.conf configuration file
52      47  #
53      48  ipvar HOME_NET any
54
55      49  # Set up the external network addresses. Leave as "any" in most s
    ituations
56      50  ipvar EXTERNAL_NET any
57      51  # If HOME_NET is defined as something other than "any", alternati
    ve, you can
58      52  # use this definition if you do not want to detect attacks from y
    our internal
59      53  # IP addresses:
60      54  #ipvar EXTERNAL_NET !$HOME_NET
61
62      55  # List of DNS servers on your network
63      56  ipvar DNS_SERVERS $HOME_NET
64
65      57  # List of SMTP servers on your network
66      58  ipvar SMTP_SERVERS $HOME_NET
67
68      59  # List of web servers on your network
69      60  ipvar HTTP_SERVERS $HOME_NET
70
71      61  # List of sql servers on your network
72      62  ipvar SQL_SERVERS $HOME_NET
73
74      63  # List of telnet servers on your network
75      64  ipvar TELNET_SERVERS $HOME_NET
76
77      65  # List of ssh servers on your network
78      66  ipvar SSH_SERVERS $HOME_NET
79
80      67  # List of ftp servers on your network
```

```
81      68  ipvar FTP_SERVERS $HOME_NET
82
83      69  # List of sip servers on your network
84      70  ipvar SIP_SERVERS $HOME_NET
85
86      71  # List of ports you run web servers on
87      72  portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,183
    0,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7
    510,7777,7779,8000,8008,8014,8028,8080,8085,8088,8090,8118,8123,8180,818
    1,8243,8280,8300,8800,8888,8899,9000,9060,9080,9090,9091,9443,9999,11371,
    34443,34444,41080,50002,55555]
88
89      73  # List of ports you want to look for SHELLCODE on.
90      74  portvar SHELLCODE_PORTS !80
91
92      75  # List of ports you might see oracle attacks on
93      76  portvar ORACLE_PORTS 1024:
94
95      77  # List of ports you want to look for SSH connections on:
96      78  portvar SSH_PORTS 22
97
98      79  # List of ports you run ftp servers on
99      80  portvar FTP_PORTS [21,2100,3535]
100
101     81  # List of ports you run SIP servers on
102     82  portvar SIP_PORTS [5060,5061,5600]
103
104     83  # List of file data ports for file inspection
105     84  portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]
106
107     85  # List of GTP ports for GTP preprocessor
108     86  portvar GTP_PORTS [2123,2152,3386]
109
110     87  # other variables, these should not be modified
111     88  ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.
    12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,2
    05.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]
112
113     89  # Path to your rules files (this can be a relative path)
114     90  # Note for Windows users:  You are advised to make this an absolu
    te path,
115     91  # such as:  c:\snort\rules
116     92  var RULE_PATH /etc/snort/rules
117     93  var SO_RULE_PATH /etc/snort/so_rules
118     94  var PREPROC_RULE_PATH /etc/snort/preproc_rules
119
120     95  # If you are using reputation preprocessor set these
```

```
121      96  # Currently there is a bug with relative paths, they are relativ
     e to where snort is
122      97  # not relative to snort.conf like the above variables
123      98  # This is completely inconsistent with how other vars work, BUG 8
     9986
124      99  # Set the absolute path appropriately
125     100  var WHITE_LIST_PATH /etc/snort/rules
126     101  var BLACK_LIST_PATH /etc/snort/rules
127
128     102  ##################################################
129     103  # Step #2: Configure the decoder.  For more information, see READ
     ME.decode
130     104  ##################################################
131
132     105  # Stop generic decode events:
133     106  config disable_decode_alerts
134
135     107  # Stop Alerts on experimental TCP options
136     108  config disable_tcpopt_experimental_alerts
137
138     109  # Stop Alerts on obsolete TCP options
139     110  config disable_tcpopt_obsolete_alerts
140
141     111  # Stop Alerts on T/TCP alerts
142     112  config disable_tcpopt_ttcp_alerts
143
144     113  # Stop Alerts on all other TCPOption type events:
145     114  config disable_tcpopt_alerts
146
147     115  # Stop Alerts on invalid ip options
148     116  config disable_ipopt_alerts
149
150     117  # Alert if value in length field (IP, TCP, UDP) is greater th ele
     ngth of the packet
151     118  # config enable_decode_oversized_alerts
152
153     119  # Same as above, but drop packet if in Inline mode (requires enab
     le_decode_oversized_alerts)
154     120  # config enable_decode_oversized_drops
155
156     121  # Configure IP / TCP checksum mode
157     122  config checksum_mode: all
158
159     123  # Configure maximum number of flowbit references.  For more infor
     mation, see README.flowbits
160     124  # config flowbits_size: 64
161
```

```
162      125  # Configure ports to ignore
163      126  # config ignore_ports: tcp 21 6667:6671 1356
164      127  # config ignore_ports: udp 1:17 53
165
166      128  # Configure active response for non inline operation. For more in
      formation, see REAMDE.active
167      129  # config response: eth0 attempts 2
```

5、（wc）统计文件行数

Plain Text

```
1  ┌──(root💀kali)-[~/桌面/work/doc]
2  └─# wc -l /etc/snort/snort.conf
                                    1 ⚙
3  735 /etc/snort/snort.conf
```

6、grep做文本过滤（Filtering Text with grep ）

Plain Text

```
1  ┌──(root💀kali)-[~/桌面/work/doc]
2  └─# cat /etc/snort/snort.conf | grep output      过滤output
                                              1 ⚙
3  #  6) Configure output plugins
4  # Step #6: Configure output plugins
5  # output unified2: filename merged.log, limit 128, nostamp, mpls_event_typ
   es, vlan_event_types
6  output unified2: filename snort.log, limit 128, nostamp, mpls_event_type
   s, vlan_event_types
7  # output alert_unified2: filename snort.alert, limit 128, nostamp
8  # output log_unified2: filename snort.log, limit 128, nostamp
9  # output alert_syslog: LOG_AUTH LOG_ALERT
10 # output log_tcpdump: tcpdump.log
```

7、Hacker Challenge: Using grep, nl, tail, and head　挑战: 使用grep，nl，tail，and head

```
┌──(root💀kali)-[~/桌面/work/doc]
└─# nl /etc/snort/snort.conf | grep output
                              1 × 1 ⚙
     33  #  6) Configure output plugins
    445  # Step #6: Configure output plugins
    450  # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
    451  output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types
    453  # output alert_unified2: filename snort.alert, limit 128, nostamp
    454  # output log_unified2: filename snort.log, limit 128, nostamp
    456  # output alert_syslog: LOG_AUTH LOG_ALERT
    458  # output log_tcpdump: tcpdump.log
┌──(root💀kali)-[~/桌面/work/doc]   取544行前面的7行
└─# cat -n /etc/snort/snort.conf | head -544 | tail -7
                                   1 ⚙
    538  # Additional configuration for specific types of installs
    539  # output alert_unified2: filename snort.alert, limit 128, nostamp
    540  # output log_unified2: filename snort.log, limit 128, nostamp
    541
    542  # syslog
    543  # output alert_syslog: LOG_AUTH LOG_ALERT
    544

```

8、只查看文件number.txt（共100行）取第20行到第30行的内容

```
┌──(root💀kali)-[~/桌面/work/doc]
└─# seq 1 100 > number.txt     创建文件
┌──(root💀kali)-[~/桌面/work/doc]
└─# head -30 number.txt | tail -11   方法一
┌──(root💀kali)-[~/桌面/work/doc]
└─# tail +20 number.txt | head -11    方法二
┌──(root💀kali)-[~/桌面/work/doc]
└─# tail -81 number.txt | head -11    方法三
┌──(root💀kali)-[~/桌面/work/doc]
└─# sed -n '20,30p' number.txt        方法四
```

Linux黑客基础-28-文本处理-02-使用sed查找和替换

文本三剑客

grep：过滤

sed：编辑

awk：截取

sed：非交互式、流编辑器

```
cat /etc/snort/snort.conf | grep mysql | sed 's/mysql/MYSQL/'
s----替换
s/old/new/  用new替换成old
echo "Hi,hy,I am hacking" | sed 's/hy/heyuan/g'
g----全局替换
echo "Hi,hy,I am hacking,you not hy" | sed 's/hy/heyuan/2'
加上数字n---只替换第n个出现的字符
针对的是一行中出现的多个要替换的字符
```

Linux黑客基础-29-文本处理-03-使用more和less查看文件

分屏查看；分页显示

结合管道符使用

1、more

回车--向下显示一行

回车--向下显示一屏

pageup--向上翻页

pagedown--向下翻页

q--退出

/关键字--按关键字查找（找到之后继续查找。可以使用n）

！命令----调用命令执行

！/bin/bash----调用了一个shell，（会用这个方法进行shell逃逸）

v---进入vi模式对文件进行编辑

2、less

光标的快速移动

gg---第一行

G----最后一行

N---打开文件时直接显示行号

回车--向下显示一行

回车--向下显示一屏

pageup--向上翻页

pagedown--向下翻页

q--退出

/关键字--按关键字查找（找到之后继续查找。可以使用n）

！命令----调用命令执行

！/bin/bash----调用了一个shell，（会用这个方法进行shell逃逸）

Linux黑客基础-30-文本处理-04-小练习

/usr/share/wordlists　kali中常用的字典目录

┌──(root💀kali)-[/usr/share/wordlists]

└─# ls

dirb　　fasttrack.txt　metasploit　rockyou.txt.gz

dirbuster　fern-wifi　　nmap.lst　　wfuzz

1. Navigate to /usr/share/wordlists/metasploit. This is a directory of multiple wordlists that can be used to brute force passwords in various password protected devices using Metasploit, the most popular pentesting and hack ing framework.（ 进入/usr/share/wordlists/metasploit. 这个目录包含了多个使用 metasploit 暴 力破解受密码保护设备时使用的字典。metasploit 是最受欢迎的渗透测试和 黑客框架。 ）

┌──(root💀 kali)-[~/桌面/work/doc]

└─# cd /usr/share/wordlists/metasploit

/usr/share/wordlists/metasploit

2. Use the cat command to view the contents of the file passwords.lst.（ 使用 cat 命令查看文件 passwords.lst 的内容。 ）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└─# cat password.lst

3. Use the more command to display the file passwords.lst.（ 使用 more 命令查看文件 passwords.lst。 ）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└─# more password.lst

4. Use the less command to view the file passwords.lst.（使用less查看password.lst文件）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└─# less password.lst

5. Now use the nl command to place line numbers on the passwords in passwords.lst. There should be 88,396 passwords.（ 使用 nl 命令让文件 passwords.lst 显示行数。这里应该有 88,396 个密码。 ）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# wc -l password.lst 2 ⚙

88397 password.lst

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# nl password.lst

6. Use the tail command to see the last 20 passwords in passwords.lst.（使用 tail 命令查看文件 passwords.lst 中最后 20 个密码。 ）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# tail -20 password.lst

7. Use the cat command to display passwords.lst and pipe it to find all the passwords that contain 123.（使用 cat 命令查看文件 passwords.lst 然后使用管道查找所有含有 123 的密 码。 ）

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# cat password.lst | grep 123

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# cat password.lst | grep 123 | wc -l 并统计有多少个

30

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# cat password.lst | grep ^123 以123开头的密码

┌──(root💀 kali)-[/usr/share/wordlists/metasploit]

└──# cat password.lst | grep 123$ 以123结尾的密码

Linux黑客基础-31-文本处理-05-sort

sort---排序

sort命令将输入文件看作由多条记录组成的数据流，而记录由课变宽的字段组成，记录之间以换行作为定界符，sort命令与awk一样，可将记录分成多个域（字段（field））及逆行处理，默认的域分隔符是空格，也可以由用户自行定义

sort比较原则：

1、从首字符向后依次比较

2、空字符串<数值<字母

3、字母是按照字母表的顺序排序

4、小写字母要排在大写字母的前面（<aA.....<z<Z）

5、如果都是同一个字母（不考虑大小写），则比较下一级（a1<A2）

6、最后按升序输出

常用选项

-n：按数字大小排序

-u：删除所有重复行

-r：逆序排序

-k：指定排序的域（指定分类是域上的数字分类项）

-t：域分隔符：用非空格或tab键分割域，类似于awk中的-F选项

-urn：

例：sort -t : -k 3 -n -r /etc/passwd

ls -l /etc | sort -k5 -n　　对etc的文件从大到小排序

ls -l /etc | grep -E -v "^d|^l"

grep

–E：扩展正则表达式　–v：反向匹配　"^d|^l" ：表示以d开头或以l开头

ls –l /etc | grep –E –v "^d|^l" | sort –k5 –rn | head　在/etc/目录中找出文件大小排在前10位的文件

Linux黑客基础–31–文本处理–05–uniq

从一个文本文件中去除或禁止重复行，注：相邻的行

语法：

uniq [选项]... [文件]

–u, ––unique　只输出不重复（内容唯一）的行

–f, ––skip–fields=N　不要比较前 N 个域

–c：显示重复的次数

–d：打印重复的行，每种行只显示其中的一行

Linux黑客基础–32–文本截取工具–06–cut

功能：是一个从指定数据中抽取指定列并将其它内容抛弃的过滤器（与colrm刚好相反，colrm从数据中删除指定列并保存其他内容）

cut程序可以方便的配合管道符使用，通常把cut命令用作管道中的过滤器。

用法：cut [选项]... [文件]...

从每个输入<文件>中输出指定部分到标准输出。

常用选项：

–b, ––bytes=列表　只选中指定的这些字节

-c, --characters=列表　　只选中指定的这些字符

-d, --delimiter=分界符　　使用指定分界符代替制表符作为区域分界

-f, --fields=列表　　只选中指定的这些域；并打印所有不包含分界符的行，除非-s 选项被指定

tail -2 /etc/passwd | cut -d: -f1

N　从第1个开始数的第N个字节、字符或域

N-　从第N个开始到所在行结束的所有字符、字节或域

N-M　从第N个开始到第M个之间(包括第M个)的所有字符、字节或域

-M　从第1个开始到第M个之间(包括第M个)的所有字符、字节或域

例;每种参数格式表示举例如下

```
echo "hello,lsl" | cut -c2
echo "hello,lsl" | cut -c2-
echo "hello,lsl" | cut -c2-5
echo "hello,lsl" | cut -c-2
echo "hello,lsl" | cut -c2,7
```

Linux黑客基础-33-文本处理-07-awk

简介：

awk是一门处理文本文件的语言，他把问价你看做一串记录（record），缺省情况下一行即为一个记录。每一行又被拆成若干个域。我们可以把每一行的第一个词看作第一域，依次内推，一个awk程序就是一连串“模式--动作”语句，awk一次读入一行，然后对照程序中的各个模式进行扫描。一旦匹配成功就执行相应的操作

功能：

1、awk是一种用于处理文本的编程语言工具，是Unix/Linux功能最强大的数据处理引擎，可以在匹配的行上执行特殊的操作，shell中最常用于截取文本种某一段数据

2、awk是由Alfred Aho、peter weinberger、brian kernighan于1977年开发的编程语言，主要用于处理文本数据和生成格式化的报告（表）是一个格式化的报告生成工具

3、awk是上述三位创建者姓的首字母。90年代unix对awk做了扩展，成为new awk，简称nawk

4、目前在linux中常用的awk版本有nawk，gawk，其中以unnutu为代表的是nawk，以redhat为代表的是gawk

```
Plain Text
```

```
┌──(root💀 kali)-[~/桌面/work/doc]
└─# ls -l /usr/bin/gawk

-rwxr-xr-x 1 root root 694624  2月 10  2021 /usr/bin/gawk
```

gawk----patern scanning and processing language

gawk是一个模式扫描及处理语言。缺省情况下它以标准输入读入并写至标准输出

5、可以在匹配的行上执行特殊的操作，shell中最常用于截取文本中的某一段数据

6、awk基本结构包括模式匹配（用于找到要处理的行）和处理过程（即处理动作）

awk　pattern　　{acion}

匹配模式----执行动作

7、awk可从文件或字符串中基于指定规则浏览和抽取信息，能够简化位版本的处理工作。是一种自解释的编程语言

8、awk读取命令行上所指定的各个文件（若无，则为标准输入），一次读取一条记录，进行分段处理。在针对每一行，应用程序所指定的命令

参数：

用法：awk [POSIX 或 GNU 风格选项] -f 脚本文件 [--] 文件 ...

用法：awk [POSIX 或 GNU 风格选项] [--] '程序' 文件 ...

-f 脚本文件　　--file=脚本文件

-F fs　　--field-separator=fs

-v var=val　　--assign=var=val

练习: Using /etc/passwd, extract the user and home directory fields for all users on your Kali machine for which the shell is set to /bin/false. Make sure you use a Bash one-liner to print the output to the screen

```
┌──(root💀kali)-[~/桌面/work/doc]
└─# grep false$ passwd | awk -F : '{print "the user " $1 "home directory is " $6}'
the user mysqlhome directory is /nonexistent
the user tsshome directory is /var/lib/tpm
the user Debian-snmphome directory is /var/lib/snmp
the user lightdmhome directory is /var/lib/lightdm
the user speech-dispatcherhome directory is /run/speech-dispatcher
┌──(root💀kali)-[~/桌面/work/doc]
└─# awk -F : '/false$/ {print "the user " $1 "home directory is " $6}' /etc/passwd
the user mysqlhome directory is /nonexistent
the user tsshome directory is /var/lib/tpm
the user Debian-snmphome directory is /var/lib/snmp
the user lightdmhome directory is /var/lib/lightdm
the user speech-dispatcherhome directory is /run/speech-dispatcher
┌──(root💀kali)-[~/桌面/work/doc]
└─# awk -F : '$7=="/bin/false" {print "the user " $1 "home directory is " $6}' /etc/passwd
the user mysqlhome directory is /nonexistent
the user tsshome directory is /var/lib/tpm
the user Debian-snmphome directory is /var/lib/snmp
the user lightdmhome directory is /var/lib/lightdm
the user speech-dispatcherhome directory is /run/speech-dispatcher
┌──(root💀kali)-[~/桌面/work/doc]
└─# awk -v FS=: '$7=="/bin/false" {print "the user " $1 "home directory is " $6}' /etc/passwd
the user mysqlhome directory is /nonexistent
the user tsshome directory is /var/lib/tpm
the user Debian-snmphome directory is /var/lib/snmp
the user lightdmhome directory is /var/lib/lightdm
the user speech-dispatcherhome directory is /run/speech-dispatcher
┌──(root💀kali)-[~/桌面/work/doc]
└─# awk -v FS=: '$NF=="/bin/false" {print "the user " $1 "home directory is " $6}' /etc/passwd
the user mysqlhome directory is /nonexistent
the user tsshome directory is /var/lib/tpm
the user Debian-snmphome directory is /var/lib/snmp
the user lightdmhome directory is /var/lib/lightdm
the user speech-dispatcherhome directory is /run/speech-dispatcher
```

Linux黑客基础-34-文本比较-01-comm

Comparing Files 文件的比较

1、comm

The comm command66 compares two text files, displaying the lines that are unique to each one, as well as the lines they have in common. It outputs three space-offset columns: the first contains lines that are unique to the first file or argument; the second contains lines that are unique to the second file or argument; and the third column contains lines that are shared by both files. The -n switch, where "n" is either 1, 2, or 3, can be used to suppress one or more columns, depending on the need.

┌──(root💀kali)-[~/桌面/work/doc/day1]

└─# comm scan-a.txt scan-b.txt

192.168.1.1

192.168.1.2

192.168.1.3

192.168.1.4

192.168.1.5

┌──(root💀kali)-[~/桌面/work/doc/day1]

└─# comm -12 scan-a.txt scan-b.txt

192.168.1.1

192.168.1.3

192.168.1.4

192.168.1.5

In the first example, comm displayed the unique lines in scan–a.txt, the unique lines in scan–b.txt and the lines found in both files respectively. In the second example, comm –12 displayed only the lines that were found in both files since we suppressed the first and second columns.

Linux黑客基础–35–文本比较–02–diff

The diff command is used to detect differences between files, similar to the comm command. However, diff is much more complex and supports many output formats. Two of the most popular formats include the context format (–c) and the unified format (–u). Listing 57 demonstrates the difference between the two formats

```
-c, -C NUM, --context[=NUM]   同时输出 NUM 行（默认为 3 行）的复制上下文内容
┌──(root💀kali)-[~/桌面/work/doc/day1]
└─# diff -c scan-a.txt scan-b.txt
*** scan-a.txt  2023-04-03 15:35:21.872675992 +0800
--- scan-b.txt  2023-04-03 15:36:46.156679386 +0800
***************
*** 1,5 ****
  192.168.1.1
- 192.168.1.2
  192.168.1.3
  192.168.1.4
  192.168.1.5
--- 1,6 ----
  192.168.1.1
  192.168.1.3
  192.168.1.4
  192.168.1.5
+ 192.1168.1.6
+
-u, -U 数量, --unified[=数量] 输出 <数量>（默认为 3）行一致化上下文
┌──(root💀kali)-[~/桌面/work/doc/day1]
└─# diff -u scan-a.txt scan-b.txt
    1 × 2 ⚙
--- scan-a.txt  2023-04-03 15:35:21.872675992 +0800
+++ scan-b.txt  2023-04-03 15:36:46.156679386 +0800
@@ -1,5 +1,6 @@
 192.168.1.1
-192.168.1.2
 192.168.1.3
 192.168.1.4
 192.168.1.5
+192.1168.1.6
+
```

Linux黑客基础-35-文本比较工具-02-vimdiff

vimdiff opens vim68 with multiple files, one in each window. The differences between files are

highlighted, which makes it easier to visually inspect them. There are a few shortcuts that may be

useful. For example:

- do: gets changes from the other window into the current one
- dp: puts the changes from the current window into the other one
-]c: jumps to the next change
- [c: jumps to the previous change
- C w: switches to the other split window

```
┌──(root💀kali)-[~/桌面/work/doc/day1]
└─# vimdiff scan-a.txt scan-b.txt
```