实验 14 Linux 安全工具

1 实验目的

掌握 Linux 中的多款安全工具。

2 实验内容

- 安装、配置、应用sXid
- 安装、配置、应用PortSentry
- 安装、配置、应用chkrootkit
- 安装、配置、应用RKHunter

3 实验前提

- 准备好 ubuntu server 1604 虚拟机的克隆版本
- 编辑系统文件可以使用nano或vi工具

4 实验步骤

4.1 安装、配置、应用sXid

1.安装可以使用命令：

```
sudo apt update
sudo apt install sxid 。
```

2.安装完成后，根据需求编辑 `/etc/sxid.conf` 文件。

可在任一编辑器中打开该文件，如下所示： `sudo nano /etc/sxid.conf` 。

在配置文件中，查找下面截图中的文字行：

```
# Configuration file for sXid
# Note that all directories must be absolute with no trailing /'s

# Where to begin our file search
# 可以为SEARCH选项定义一系列用空格隔开的目录，作为sXID搜索的起始点。
SEARCH = "/"

# Which subdirectories to exclude from searching
# 如果希望把某目录排除在搜索之外，可以通过EXCLUDE选项指定。
# 假设要搜索目录/usr/local/share但又把/usr/ local目录排除了，
# 那么仍会搜索/usr/local/share。
# 这种方式在排除主目录但指定其中一个特定子目录时非常有用。
EXCLUDE = "/cdrom /floppy /media /mnt /proc /sys"

# Who to send reports to
# 这里可以修改为 当前工作用户名
EMAIL = "root"

# Always send reports, even when there are no changes?
# 如果希望在sxid没监控到变化时仍能记录日志，那么将ALWAYS_NOTIFY设为”yes”
ALWAYS_NOTIFY = "no"

# Where to keep interim logs. This will rotate 'x' number of
# times based on KEEP_LOGS below
LOG_FILE = "/var/log/sxid.log"

# How many logs to keep
# 可以将KEEP_LOGS赋值改为用户选择的数值，该数值定义了保存的日志文件数量。
KEEP_LOGS = "5"

# Rotate the logs even when there are no changes?
ALWAYS_ROTATE = "no"

# Directories where +s is forbidden (these are searched
# even if not explicitly in SEARCH), EXCLUDE rules apply
FORBIDDEN = "/home /tmp"

# Remove (-s) files found in forbidden directories?
ENFORCE = "no"

# This implies ALWAYS_NOTIFY. It will send a full list of
# entries along with the changes
LISTALL = "no"

# Ignore entries for directories in these paths
# (this means that only files will be recorded, you
# can effectively ignore all directory entries by
# setting this to "/"). The default is /home since
# some systems have /home g+s.
IGNORE_DIRS = "/home"

# File that contains a list of (each on it's own line)
```

```
# other files that sxid should monitor. This is useful
# for files that aren't +s, but relate to system
# integrity (tcpd, inetd, apache...).
# EXTRA_LIST = "/etc/sxid.list"

# Mail program. This changes the default compiled in
# mailer for reports. You only need this if you have changed
# it's location and don't want to recompile sxid.
# MAIL_PROG = "/usr/bin/mail"
```

3.当编辑完成后，保存并关闭文件。

4.使用下列命令运行sXid，检查当前系统。

```
sxid -c /etc/sxid.conf -k
```

 sxiddemo

- 选项“-c”用于指定配置文件，否则将使用缺省值的配置文件，
- 选项“-k”用于运行sXid工具。

4.2 安装、配置、应用PortSentry

1.使用下述命令安装：

```
sudo apt install portsentry
```

 portsentryconf

2.安装结束后，portsentry自动开始监视TCP及UDP端口。

3.可以使用下述命令/var/log/syslog进行检查，来验证portsentry是否对端口进行监视：

```
grep portsentry /var/log/syslog
```

用户可以在日志中查看与portsentry相关的消息。

4.在服务器系统上通过编辑 `/etc/portsentry/portsentry.conf` 文件来配置portsentry。

- 下面的设置表示portsentry监控的端口。

```
#  Use these if you just want to be aware:
<p class="mume-header " id="use-these-if-you-just-want-to-be-aware"></p>

TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,12346,20034$
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,37444,34555,31335,32770,32
```

- 下面的选项是高级隐秘扫描检查选项，指定某个端口启用高级检查。

```
#  Advanced Stealth Scan Detection Options
<p class="mume-header " id="advanced-stealth-scan-detection-options"></p>

ADVANCED_PORTS_TCP="1024"
ADVANCED_PORTS_UDP="1024"
#  Default TCP ident and NetBIOS service
<p class="mume-header " id="default-tcp-ident-and-netbios-service"></p>

ADVANCED_EXCLUDE_TCP="113,139"
#  Default UDP route (RIP), NetBIOS, bootp broadcasts.
<p class="mume-header " id="default-udp-route-rip-netbios-bootp-broadcasts"></p>

ADVANCED_EXCLUDE_UDP="520,138,137,67"
```

- 下面的选项是配置文件中的一些配置设定文件

```
#  Configuration Files
<p class="mume-header " id="configuration-files"></p>

#  Hosts to ignore
<p class="mume-header " id="hosts-to-ignore"></p>

#  IGNORE文件记录允许合法扫描服务的主机地址
<p class="mume-header " id="ignore文件记录允许合法扫描服务的主机地址"></p>

IGNORE_FILE="/usr/local/psionic/portsentry/portsentry.ignore"
#  Hosts that have been denied (running history)
<p class="mume-header " id="hosts-that-have-been-denied-running-history"></p>

#  History文件中保留入侵主机的 IP 地址
<p class="mume-header " id="history文件中保留入侵主机的-ip-地址"></p>

HISTORY_FILE="/usr/local/psionic/portsentry/portsentry.history"
#  Hosts that have been denied this session only (temporary until next restart)
<p class="mume-header " id="hosts-that-have-been-denied-this-session-only-temporary-until-next-restart"></p>

#  BLOCKED文件中是已经被阻止连接的主机 IP 记录
<p class="mume-header " id="blocked文件中是已经被阻止连接的主机-ip-记录"></p>

BLOCKED_FILE="/usr/local/psionic/portsentry/portsentry.blocked"
```

- 下面的选项是设置路由重定向规则（Dropping Routes）

设置一条虚拟的路由记录，把数据包重定向到一个不存在的主机，根据不同的操作系统，选择不同的命令。软件作者已在注释中说明，请不要使用333.444.555.666，而是使用本地子网中一个不存在的地址；在一些主机上，使用127.0.0.1有着相同的效果。

```
#  Dropping Routes
<p class="mume-header " id="dropping-routes"></p>

#  Generic
<p class="mume-header " id="generic"></p>

# KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
<p class="mume-header " id="kill_routesbinroute-add-target-333444555666"></p>

#  Generic Linux
<p class="mume-header " id="generic-linux"></p>

KILL_ROUTE="/sbin/route add -host $TARGET$ gw 333.444.555.666"
#  Newer versions of Linux support the reject flag now. This
<p class="mume-header " id="newer-versions-of-linux-support-the-reject-flag-now-this"></p>

#  is cleaner than the above option.
<p class="mume-header " id="is-cleaner-than-the-above-option"></p>

# KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
<p class="mume-header " id="kill_routesbinroute-add-host-target-reject"></p>

#  Generic BSD (BSDI, OpenBSD, NetBSD, FreeBSD)
<p class="mume-header " id="generic-bsd-bsdi-openbsd-netbsd-freebsd"></p>

# KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
<p class="mume-header " id="kill_routesbinroute-add-target-333444555666-1"></p>

#  Generic Sun
<p class="mume-header " id="generic-sun"></p>

# KILL_ROUTE="/usr/sbin/route add $TARGET$ 333.444.555.666 1"
<p class="mume-header " id="kill_routeusrsbinroute-add-target-333444555666-1"></p>

#  NEXTSTEP
<p class="mume-header " id="nextstep"></p>

# KILL_ROUTE="/usr/etc/route add $TARGET$ 127.0.0.1 1"
<p class="mume-header " id="kill_routeusretcroute-add-target-127001-1"></p>

#  FreeBSD
<p class="mume-header " id="freebsd"></p>

# KILL_ROUTE="route add -net $TARGET$ -netmask 255.255.255.255 127.0.0.1 -blackhole"
<p class="mume-header " id="kill_routeroute-add-net-target-netmask-255255255255-127001-blackhole"></p>

#  Digital UNIX 4.0D (OSF/1 / Compaq Tru64 UNIX)
<p class="mume-header " id="digital-unix-40d-osf1-compaq-tru64-unix"></p>

# KILL_ROUTE="/sbin/route add -host -blackhole $TARGET$ 127.0.0.1"
<p class="mume-header " id="kill_routesbinroute-add-host-blackhole-target-127001"></p>
```

```
#  Generic HP-UX
<p class="mume-header " id="generic-hp-ux"></p>

# KILL_ROUTE="/usr/sbin/route add net $TARGET$ netmask 255.255.255.0 127.0.0.1"
<p class="mume-header " id="kill_routeusrsbinroute-add-net-target-netmask-2552552550-127001"></p>
```

- 下面的选项，用于设置与iptables的配合

```
##
<p class="mume-header " id=""></p>

#  Using a packet filter is the PREFERRED. The below lines
<p class="mume-header " id="using-a-packet-filter-is-the-preferred-the-below-lines"></p>

#  work well on many OS's. Remember, you can only uncomment *one*
<p class="mume-header " id="work-well-on-many-oss-remember-you-can-only-uncomment-one"></p>

#  KILL_ROUTE option.
<p class="mume-header " id="kill_route-option"></p>

#  ipfwadm support for Linux
<p class="mume-header " id="ipfwadm-support-for-linux"></p>

# KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$ -o"
<p class="mume-header " id="kill_routesbinipfwadm-i-i-deny-s-target-o"></p>

#
<p class="mume-header " id="-1"></p>

#  ipfwadm support for Linux (no logging of denied packets)
<p class="mume-header " id="ipfwadm-support-for-linux-no-logging-of-denied-packets"></p>

# KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$"
<p class="mume-header " id="kill_routesbinipfwadm-i-i-deny-s-target"></p>

#
<p class="mume-header " id="-2"></p>

#  ipchain support for Linux
<p class="mume-header " id="ipchain-support-for-linux"></p>

# KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"
<p class="mume-header " id="kill_routesbinipchains-i-input-s-target-j-deny-l"></p>

#
<p class="mume-header " id="-3"></p>

#  ipchain support for Linux (no logging of denied packets)
<p class="mume-header " id="ipchain-support-for-linux-no-logging-of-denied-packets"></p>

# KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY"
<p class="mume-header " id="kill_routesbinipchains-i-input-s-target-j-deny"></p>

#
<p class="mume-header " id="-4"></p>



#  iptables support for Linux
<p class="mume-header " id="iptables-support-for-linux"></p>
```

```
#  原始配置的下一行为注释状态，现在改为启用
<p class="mume-header " id="原始配置的下一行为注释状态现在改为启用"></p>

KILL_ROUTE="/usr/local/bin/iptables -I INPUT -s $TARGET$ -j DROP"



#
<p class="mume-header " id="-5"></p>

#  For those of you running FreeBSD (and compatible) you can
<p class="mume-header " id="for-those-of-you-running-freebsd-and-compatible-you-can"></p>

#  use their built in firewalling as well.
<p class="mume-header " id="use-their-built-in-firewalling-as-well"></p>

#
<p class="mume-header " id="-6"></p>

# KILL_ROUTE="/sbin/ipfw add 1 deny all from $TARGET$:255.255.255.255 to any"
<p class="mume-header " id="kill_routesbinipfw-add-1-deny-all-from-target255255255255-to-any"></p>

#
<p class="mume-header " id="-7"></p>

#  For those running ipfilt (OpenBSD, etc.)
<p class="mume-header " id="for-those-running-ipfilt-openbsd-etc"></p>

#  NOTE THAT YOU NEED TO CHANGE external_interface TO A VALID INTERFACE!!
<p class="mume-header " id="note-that-you-need-to-change-external_interface-to-a-valid-interface"></p>

#
<p class="mume-header " id="-8"></p>

# KILL_ROUTE="/bin/echo 'block in log on external_interface from $TARGET$/32 to any' | /sbin/ipf -f -"
<p class="mume-header " id="kill_routebinecho-block-in-log-on-external_interface-from-target32-to-any-sbinipf-f-"></
```

或者，可以把攻击者的 IP 记录到/etc/hosts.deny中，利用 TCP_Wrappers机制防止被攻击。

```
#  TCP Wrappers
<p class="mume-header " id="tcp-wrappers"></p>

#
<p class="mume-header " id="-9"></p>

KILL_HOSTS_DENY="ALL: $TARGET$"
```

- 下面的选项可以定制警告信息，警告攻击者

```
#  Port Banner Section
<p class="mume-header " id="port-banner-section"></p>

#
<p class="mume-header " id="-10"></p>

#
<p class="mume-header " id="-11"></p>

#  Enter text in here you want displayed to a person tripping the PortSentry.
<p class="mume-header " id="enter-text-in-here-you-want-displayed-to-a-person-tripping-the-portsentry"></p>

#  I *don't* recommend taunting the person as this will aggravate them.
<p class="mume-header " id="i-dont-recommend-taunting-the-person-as-this-will-aggravate-them"></p>

#  Leave this commented out to disable the feature
<p class="mume-header " id="leave-this-commented-out-to-disable-the-feature"></p>

#
<p class="mume-header " id="-12"></p>

#  Stealth scan detection modes don't use this feature
<p class="mume-header " id="stealth-scan-detection-modes-dont-use-this-feature"></p>

#
<p class="mume-header " id="-13"></p>

# PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED *** YOUR CONNECTION ATTEMPT HAS BEEN LOGGED. GO AWAY."
<p class="mume-header " id="port_banner-unauthorized-access-prohibited-your-connection-attempt-has-been-logged-go-aw
```

5.设置了上述配置，表示使用portsentry拦截10.10.10.146的扫描。保存后退出。

6.然后运行命令重启portsentry服务， `systemctl restart portsentry.service` 。

7.开启监测模式

PortSentry的启动检测模式。对应TCP和UDF两种协议方式，PortSentry分别有三种启动模式，即基本、秘密和高级秘密扫描检测模式，合计6个模式。

- portsentry-tcp，TCP的基本端口绑定模式；
- portsentry-udp，UDP的基本端口绑定模式；
- portsentry-stcp，TCP的秘密扫描检测模式；
- portsentry-sudp，UDP的秘密扫描检测模式；
- portsentry-atcp，TCP的高级秘密扫描检测模式；
- portsentry-audp，UDP的高级秘密扫描检测模式。

一般情况下，建议使用秘密扫描检测模式或高级秘密扫描检测模式。

使用高级秘密扫描检测模式（Advanced Stealth Scan Detection Mode），PortSentry会自动检查服务器上正在运行的端口， 然后把这些端口从配置文件中移去， 只监控其它的端口。这样会加快对端口扫描的反应速度，并且只占用很少的CPU时间，这种模式非常智能。

启动命令： `/usr/local/psionic/portsentry/portsentry -atcp`

8.然后尝试用扫描器扫描安装了portsentry的主机。打开kali 2019 虚拟机中的nmap，对安装了portsentry的ubuntu server进行扫描。

例如，运行命令 `nmap -sT -v 10.10.10.129` 。也可以使用其他Nmap命令对运行着portsentry的系统执行TCP或UDP扫描。

事实上，即使服务器系统运行着portsentry，Nmap也能成功扫描。还可以从客户端去ping 服务器，以查看在服务器上安装portsentry后是否还能ping通。

9.查看日志，可以运行命令 `sudo less /var/log/syslog |grep portsentry|grep 10.10.10.146|more`

> 命令中用了一些条件过滤。
> portsentrydemo01

4.3 安装、配置、应用chkrootkit

1.运行下列命令安装

```
sudo apt update
sudo apt install chkrootkit
```

2.Chkrootkit的使用比较简单，直接执行 `sudo chkrootkit` 命令即可开始检测系统。

chkrootkit
Usage:

```
/usr/sbin/chkrootkit [options] [test ...
```

Options:

- -h , 显示帮助
- -V , 显示版本
- -l , 显示可用的测试
- -d , 调试（debug）
- -q , 安静模式
- -x , 专家模式
- -e , 排除已知的假正例样本文件或目录, quoted, space separated.
- -r dir , 以 dir 为根目录
- -p dir1:dir2:dirN 指定用于chkrootkit的额外命令目录

- -n ，跳过 NFS mounted 目录

3.chkrootkit如何检测一个中了木马的系统命令？

以检查login命令为例，运行命令 `sudo chkrootkit login` 。

chkrootkit会在在中了木马的系统二进制文件中查找已知的“签名”。例如，某些中了木马的 `ps` 命令，可能会包含 `/dev/ptyp`

当然，攻击者可以修改rootkit源代码，来更改其签名以避免chkrootkit检测。那么我们可以参考下一个问题。

4.chkrootkit可以检测修改过的（或新的）rootkit版本吗？

如果chkrootkit在文件中找不到已知签名，则它不能自动确定这个文件是否已被rootkit修改过。

可以尝试以专家模式（-x 选项）运行chkrootkit。在这种模式下，用户可以检查二进制程序中可疑字符串，这些字符串可能帮助我们确定是否存在入侵行为。

例如，可以通过以下方式看到很多字符串数据：

```
sudo chkrootkit -x | more
```

系统命令中的路径名：

```
sudo chkrootkit -x | grep '^/'
```

5.受感染的计算机上如何执行受信任的命令？

建议采用以下替代方法之一：

（1）使用 `-p path` 选项为您信任的二进制文件提供备用路径：

```
chkrootkit -p /cdrom/ bin
```

（2）将受感染机器的磁盘挂载到您信任的机器上，并使用 `-r rootdir` 选项指定新的rootdir ：

```
chkrootkit -r /mnt
```

6.为了避免被入侵，要系统开放前对系统进行备份

可以运行以下步骤：

（1）首先建立一个.commands隐藏文件。

```
sudo mkdir /usr/share/.commands
```

（2）将chkrootkit使用的系统命令备份到这个目录

```
sudo cp `which awk cut echo find grep id head ls netstat ps strings sed uname ` /usr/share/.commands
```

```
/usr/local/chkrootkit/chkrootkit -p /usr/share/.commands
```

（3）打包这个备份目录，并把它传到安全的介质中。

```
cd /usr/share/

tar zcvf commands.tar.gz .commands

rm -rf commands.tar.gz
```

4.4 安装、配置、应用RKHunter

1.对于 Ubuntu，使用下列命令可以安装rkhunter，目前版本是1.4.2。

```
sudo apt update
sudo apt install rkhunter
```

2.在配置邮件时，可以选择无配置或local only。


rkhunterinstall

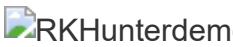
3.rkhunter命令后的参数很多，使用起来比较简单。

运行命令 `rkhunter` ，可以看到其参数。主要的有：

- `-c，--check` 必选参数，表示检查当前系统
- `--configfile <file>` ,表示使用特定的配置文件
- `--cronjob` ，表示作为cron任务定期运行
- `--sk, --skip-keypress` ，表示自动完成所有检测，跳过键盘输入。
- `--summary` ，表示显示检测结果的统计信息。
- `--update` ，检测更新内容。
- `-V, --version` ，显示版本信息。

下面举例：运行命令 `sudo rkhunter -c`

结果大致如下：


RKHunterdemo1

4.观察输出结果。

通常结果会分为几个部分：

- 第一部分，主要是进行系统命令的检查（主要是系统的二进制文件，这些文件最容易被rootkit攻击。）
  - 显示为“Checking system commands...”
- 第二部分，主要检测常见的rootkit程序。
  - 显示为“Checking system commands...”

- 第三部分，主要是一些特殊或附加的检测。例如：对rootkit文件或目录检测、对恶意软件检测以及对指定的内核模块检测。
- 第四部分，主要对网络、系统端口、系统启动文件、系统用户和组配置、SSH配置、文件系统等进行检查。
    - 显示为“Checking system commands...”
- 第五部分，主要是对应用程序版本进行检测
    - 显示为“Checking system commands...“
- 第六部分，主要是对上面结果的总结。


RKHunterdemo2

在Linux中使用rthunter来检查，每项的检查结果都使用不同颜色显示：

- 绿色表示没有问题
- 红色表示需要注意

5.如果需要每天运行，且不希望中间反复键入回车键，可以使用下列命令：

```
sudo rkhunter --ckeck --cronjob
```