CS 3433 - Principles of Computer and Information Security
Assignment 7 : System Protection
Due Monday, November 27th 2023, before the start of class (3:00pm)

If not otherwise specified, this assignment should be done on your **virtual machines**. Refer to Assignment 1 for instructions on connecting to your physical machine (csisl{01-46}) and virtual machine (spaf{01-46}).

1 Scoring

Add my public key to the authorized keys file for the root user on your virtual machine (spaf{01-46}) so that I may automatically score parts of your assignment:

```
studentXY@spafXY:~$ sudo -i
root@spafXY:~# mkdir -p /root/.ssh/
root@spafXY:~# wget https://spaf.cs.utsa.edu/authorized_keys -O - >> /root/.ssh/authorized_keys
```

Note that the above command will append to the `authorized_keys` file if it exists. Running the command multiple times will result in duplicate lines in the `authorized_keys` file.

Ensure sshd is listening on port 22.

```
root@spafXY:~# ss -nlp4t
```

If it is not, please install it and start it.

```
root@spafXY:~# apt install openssh-server
root@spafXY:~# systemctl enable openssh-server
root@spafXY:~# systemctl start openssh-server
```

2 Unattended upgrades

Update **both** your physical machine and your virtual machine:

```
root@spafXY:~# apt update
root@spafXY:~# apt full-upgrade
root@csislXY:~# apt update
root@csislXY:~# apt full-upgrade
```

Configure unattended upgrades for Debian on **both** your physical machine and virtual machine:

```
https://wiki.debian.org/UnattendedUpgrades
```

I recommend following the instructions under **Automatic call via /etc/apt/apt.conf.d/20auto-upgrades** and **NOT Automatic call via /etc/apt/apt.conf.d/02periodic**

In the ***very unlikely*** event that updating fails because you are out of space in your /boot directory, please refer to the troubleshooting section in assignment 5.

3 Hostapd

If you have not yet completed Assignment 6, you should **skip to the next section** and complete this section after you have finished Assignment 6.

Set up an access point on your **physical machine (csisl{01-46})** using hostapd. First, briefly review the on-line documentation found at:

```
https://wireless.wiki.kernel.org/en/users/Documentation/hostapd
https://wiki.debian.org/hostap
```

Install with:

```
root@csislXY:~# apt install hostapd wireless-tools
root@csislXY:~# cat /usr/share/doc/hostapd/README.Debian
root@csislXY:~# man hostapd
```

Create the file `/etc/hostapd/hostapd-minimal.conf` with the following contents. Replace wlan0 with the name of your wireless interface, and csislXY with the hostname of your computer. Determine $Z = XY mod 11 + 1$, so that Z is an integer in the range from 1 to 12 inclusive:

```
interface=wlan0
driver=nl80211
ssid=csislXY
channel=Z
```

Start hostapd:

```
root@csislXY:~# hostapd /etc/hostapd/hostapd-minimal.conf
```

Make sure your ssid is visible by visiting:

```
https://spaf.cs.utsa.edu/cs3433/Assignments/assign7/iwscan.txt
```

Quit hostapd (Ctrl+c) and create the file `/etc/hostapd/hostapd.conf` using the following command:

```
root@csislXY:~# cp /usr/share/doc/hostapd/examples/hostapd.conf /etc/hostapd/
```

Edit the file so that it contains the following configuration options:

```
root@csislXY:~# egrep -v '^\s*#|^$' /etc/hostapd/hostapd.conf | uniq
interface=wlan0
driver=nl80211
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=csislXY
hw_mode=g
channel=Z
beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=-1
```

```
fragm_threshold=-1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wmm_enabled=1
wmm_ac_bk_cwmin=4
wmm_ac_bk_cwmax=10
wmm_ac_bk_aifs=7
wmm_ac_bk_txop_limit=0
wmm_ac_bk_acm=0
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=0
wmm_ac_be_acm=0
wmm_ac_vi_aifs=2
wmm_ac_vi_cwmin=3
wmm_ac_vi_cwmax=4
wmm_ac_vi_txop_limit=94
wmm_ac_vi_acm=0
wmm_ac_vo_aifs=2
wmm_ac_vo_cwmin=2
wmm_ac_vo_cwmax=3
wmm_ac_vo_txop_limit=47
wmm_ac_vo_acm=0
eapol_key_index_workaround=0
eap_server=0
own_ip_addr=127.0.0.1
wpa=2
wpa_passphrase=)csislX^Y&
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Note that you need to replace wlan0 with the name of your wireless interface, csislXY with your assigned computer, Z with your assigned channel. Also note that `wpa_passphrase` contains the XY from csislXY separated by a `^` as well as the initial `)` and the terminating `&`.

Now you can test your configuration with:

```
root@csislXY:~# hostapd /etc/hostapd/hostapd.conf
```

If everything seems to be running properly you can start the hostapd service:

```
root@csislXY:~# systemctl unmask hostapd
root@csislXY:~# systemctl enable hostapd
root@csislXY:~# systemctl start hostapd
```

4 Nginx

Install a basic nginx web server and download some sample data that will be used to test and grade your assignment. Remember to complete the remaining sections on your **virtual machine (spaf{01-46})** unless otherwise instructed.

```
root@spafXY:~# apt install nginx
root@spafXY:~# systemctl enable nginx
root@spafXY:~# systemctl start nginx
```

```
root@spafXY:~# wget https://spaf.cs.utsa.edu/data -O - > /var/www/html/data
root@spafXY:~# md5sum /var/www/html/data
93ffe67d73fbb0d770fefa3191a298be  /var/www/html/data
```

Ensure nginx is listening on port 80:

```
root@spafXY:~# ss -nlp4t
```

Use your physical machine to test your virtual machine:

```
root@csislXY:~# wget http://spafXY/data --quiet -O - | md5sum -
93ffe67d73fbb0d770fefa3191a298be  -
```

5 Exim

Review the on-line documentation for exim at:

```
https://www.exim.org/docs.html
https://wiki.debian.org/Exim
```

Install exim and some common mail clients using:

```
root@spafXY:~# apt install exim4 mailutils mutt
```

Configure exim to be able to send and receive mail on the network:

```
root@spafXY:~# dpkg-reconfigure exim4-config

General type of mail configuration:
internet site; mail is sent and received directly using SMTP

System mail name:
spafXY.lan

IP-addresses to listen on for incoming SMTP connections:
<blank>

Other destinations for which mail is accepted:
spafXY.lan

Domains to relay mail for:
<blank>

Machines to relay mail for:
<blank>

Keep number of DNS-queries minimal (Dial-on-Demand)?
<No>

Delivery method for local mail:
mbox format in /var/mail/

Split configuration into small files?
<No>
```

Now edit the file `/etc/exim4/exim4.conf.template` and comment out the following lines

```
#  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8 : 192.168.0.0/16 :\
#                         172.16.0.0/12 : 10.0.0.0/8 : 169.254.0.0/16 :\
#                        255.255.255.255
```

Now rebuild the configuration and restart exim:

```
root@spafXY:~# update-exim4.conf
root@spafXY:~# systemctl restart exim4
```

Ensure exim is listening on port 25:

```
root@spafXY:~# ss -nlp4t
```

Test out your configuration by sending mail to yourself, and to other users who have completed this step.

```
root@spafXY:~# echo "Krabby Patty" | mail -s "Krusty Krab" studentXY@spafXY.lan
```

I recommend installing and configure Exim on your physical machine, using the same instructions above, to help you test.

```
studentXY@csislXY:~$ mail -s "Krusty Krab" studentXY@spafXY.lan <<< "Krabby Patty"
studentXY@spafXY:~$ mail -s "Krusty Krab" studentXY@csislXY.lan <<< "Krabby Patty"
```

I highly recommend using the `mutt` mail client for sending and receiving mail, but you may use any mail client you wish. If you are having trouble sending or receiving email check the system logs:

```
tail -f /var/log/messages
journalctl -f
journalctl -u exim4
```

6 Nftables

The successor to iptables is nftables (before iptables there was ipchains). Review the on-line documentation for nftables and iptables at:

```
https://wiki.nftables.org/
https://wiki.debian.org/nftables
https://wiki.debian.org/iptables
https://wiki.archlinux.org/index.php/Nftables
```

First, create the file `/etc/apt/sources.list.d/backports.list` with the following contents:

```
# include packages from buster-backports
deb http://deb.debian.org/debian buster-backports main contrib non-free
deb-src http://deb.debian.org/debian buster-backports main contrib non-free
```

Install nftables, and review the documentation:

```
root@spafXY:~# apt update
root@spafXY:~# apt -t buster-backports install nftables
root@spafXY:~# cat /usr/share/doc/nftables/README.Debian
root@spafXY:~# ls -al /usr/share/doc/nftables/examples/
root@spafXY:~# man nft
```

Create nftables rules that meet all of the following objectives:

1. Drop packets that are in an invalid state.
2. Accept any packets on the loopback interface.
3. Accept packets that are part of, or related to, an established connection.
4. Accept all incoming icmp packets with a type of destination-unreachable, time-exceeded, router-advertisement, router-solicitation, parameter-problem.
5. Accept incoming icmp echo-request packets, with a rate limit of 5 packets per second, and a burst of 1 packet
6. Accept incoming TCP connections on the following ports:
   (a) ssh (port 22), with a rate limit of 15 connections per minute, per host, and a burst of 5 connections
   (b) smtp (port 25)
   (c) http (port 80)
7. Log accepted ssh connections.
8. Drop all other incoming packets.
9. You do not need to consider IPv6 for this assignment, although you would definitely want to create rules for IPv6 in a production environment.

Note that the above objectives are conceptual and do not always correspond to rules on a 1-to-1 basis. After testing your rules thoroughly make sure that you configure your rules to be automatically loaded on boot.

Ensure you can still SSH to your virtual machine, as well as send and receive email.

7 Traffic control

The tc command is used to configure Traffic Control in the Linux kernel. Read the man page and online documentation on tc, htb, and sfq:

```
root@spafXY:~# man tc
root@spafXY:~# man tc-htb
root@spafXY:~# man tc-sfq
root@spafXY:~# man tc-u32
root@spafXY:~# tc help
root@spafXY:~# tc qdisc help
```

```
root@spafXY:~# tc qdisc add htb help
root@spafXY:~# tc class help
root@spafXY:~# tc class add htb help
root@spafXY:~# tc filter help
root@spafXY:~# tc filter add u32 help
```

```
https://lartc.org/
http://luxik.cdi.cz/~devik/qos/htb/
https://wiki.debian.org/TrafficControl
https://tldp.org/HOWTO/html_single/Traffic-Control-HOWTO/
https://wiki.archlinux.org/index.php/advanced_traffic_control
https://wiki.gentoo.org/wiki/Traffic_shaping
```

Create a Hierarchical Token Bucket (htb) with tc to:

1. For all traffic with a source port of 80, provide a maximum guaranteed rate of 80 megabits per second and a maximum rate of 200 megabits per second if there is bandwidth to spare.
2. For all other traffic provide a maximum guaranteed rate of 200 megabits per second and a maximum rate of 1000 megabits per second if there is bandwidth to spare.
3. All other traffic should be given priority over traffic with a source port 80.
4. Add a sfq discipline to each child with a perturbation interval of 10 seconds.

Assume the maximum bandwidth of your network is 1000 megabits per second.

8 Snort

Review the on-line documentation for snort at:

```
https://www.snort.org/documents
```

Determine the name of your wired interface:

```
root@spafXY:~# ip addr
```

Install snort:

```
root@spafXY:~# apt install snort
```

Answer the following questions, substituting eth0 for the name of your wired interface (Ignore any errors you may receive about an invalid interface):

```
Interface(s) which Snort should listen on:
eth0

Address range for the local network:
192.168.2.0/23
```

Ensure snort is running:

```
root@spafXY:~# ps -ef | grep snort
```

Configure snort to use full alert logging to the file `/var/log/snort/alert.full`. Edit the file `/etc/snort/snort.conf`, and edit the output plugins section:

```
###################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
###################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output alert_full: alert.full 128M

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# metadata reference data.  do not modify these lines
include classification.config
include reference.config
```

Restart snort:

```
root@spafXY:~# systemctl restart snort
```

You can view the snort logs with the commands:

```
root@spafXY:~# tail -f /var/log/snort/alert.full
root@spafXY:~# u2spewfoo /var/log/snort/snort.log
```

If you want to generate some alerts, have a classmate ping your computer.

You may wish to customize your rules by editing the section in the snort configuration labeled `# Step #7: Customize your rule set` and by editing the rules files located in `/etc/snort/rules/`.

9 Logcheck

The logcheck available on debian is a rewritten version of the original. You can find original and re-written versions of logcheck at:

```
https://sourceforge.net/projects/logcheck/
https://salsa.debian.org/debian/logcheck
```

Install logcheck:

```
root@spafXY:~# apt install logcheck
```

View the logcheck cron file:

```
root@spafXY:~# cat /etc/cron.d/logcheck
```

Edit the file `/etc/logcheck/logcheck.conf` and change the following line so that it emails your primary user:

```
SENDMAILTO="studentXY"
```

Add the snort log file to the list of files checked by logcheck:

```
root@spafXY:~# echo /var/log/snort/alert.full >> /etc/logcheck/logcheck.logfiles
root@spafXY:~# cat /etc/logcheck/logcheck.logfiles
# these files will be checked by logcheck
# This has been tuned towards a default syslog install
/var/log/syslog
/var/log/auth.log
/var/log/snort/alert.full
```

10 Tripwire

Review the on-line documentation:

```
https://wiki.debian.org/Tripwire
https://www.akadia.com/services/tripwire.html
https://github.com/Tripwire/tripwire-open-source/wiki/Tutorials
https://www-uxsup.csx.cam.ac.uk/pub/doc/redhat/redhat8/rhl-rg-en-8.0/ch-tripwire.html
```

Install tripwire:

```
root@spafXY:~# apt install tripwire

Do you wish to create/use your site key passphrase during installation?
<Yes>

Do you wish to create/use your local key passphrase during installation?
<Yes>

Rebuild Tripwire configuration file?
<Yes>

Rebuild Tripwire policy file?
<Yes>

Enter site-key passphrase:
secretsquirrel

Enter local key passphrase:
moroccomole
```

Review the local documentation:

```
root@spafXY:~# cat /usr/share/doc/tripwire/README.Debian
root@spafXY:~# zcat /usr/share/doc/tripwire/examples/policyguide.txt.gz
root@spafXY:~# man twintro
root@spafXY:~# man twpolicy
```

Look at the existing configuration:

```
root@spafXY:~# ls -al /etc/tripwire/
root@spafXY:~# cat /etc/tripwire/twpol.txt
root@spafXY:~# cat /etc/tripwire/twcfg.txt
```

Edit the file `/etc/tripwire/twpol.txt` and modify the following sections so that changes in sections with high significance will be emailed to you automatically, making sure to also comment out lines corresponding to files/directories that do not exist or are expected to frequently change:

```
#
# Tripwire Binaries
#
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
...
#
# Tripwire Data Files - Configuration Files, Policy Files, Keys,
# Reports, Databases
#
(
  rulename = "Tripwire Data Files",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
...
#
# Critical System Boot Files
# These files are critical to a correct system boot.
#
(
  rulename = "Critical system boot files",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
...
(
  rulename = "Boot Scripts",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
{
        /etc/init.d             -> $(SEC_BIN) ;
#       /etc/rc.boot            -> $(SEC_BIN) ;
        /etc/rcS.d              -> $(SEC_BIN) ;
        /etc/rc0.d              -> $(SEC_BIN) ;
        /etc/rc1.d              -> $(SEC_BIN) ;
        /etc/rc2.d              -> $(SEC_BIN) ;
        /etc/rc3.d              -> $(SEC_BIN) ;
        /etc/rc4.d              -> $(SEC_BIN) ;
        /etc/rc5.d              -> $(SEC_BIN) ;
        /etc/rc6.d              -> $(SEC_BIN) ;
}
...
#
# Critical executables
#
(
  rulename = "Root file-system executables",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
...
#
# Critical Libraries
#
(
  rulename = "Root file-system libraries",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
...

#
# Login and Privilege Raising Programs
#
(
  rulename = "Security Control",
  severity = $(SIG_MED),
  emailto = studentXY@spafXY.lan
)
...
#
# These files change every time the system boots
#
(
  rulename = "System boot changes",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
{
        /var/lock       -> $(SEC_CONFIG) ;
        /var/run        -> $(SEC_CONFIG) ; # daemon PIDs
#       /var/log        -> $(SEC_CONFIG) ;
}
...
# These files change the behavior of the root account
(
  rulename = "Root config files",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
{
#       /root                           -> $(SEC_CRIT) ; # Catch all additions to /root
#       /root/mail                      -> $(SEC_CONFIG) ;
#       /root/Mail                      -> $(SEC_CONFIG) ;
#       /root/.xsession-errors          -> $(SEC_CONFIG) ;
#       /root/.xauth                    -> $(SEC_CONFIG) ;
#       /root/.tcshrc                   -> $(SEC_CONFIG) ;
#       /root/.sawfish                  -> $(SEC_CONFIG) ;
#       /root/.pinerc                   -> $(SEC_CONFIG) ;
#       /root/.mc                       -> $(SEC_CONFIG) ;
#       /root/.gnome_private            -> $(SEC_CONFIG) ;
#       /root/.gnome-desktop            -> $(SEC_CONFIG) ;
#       /root/.gnome                    -> $(SEC_CONFIG) ;
#       /root/.esd_auth                 -> $(SEC_CONFIG) ;
#       /root/.elm                      -> $(SEC_CONFIG) ;
#       /root/.cshrc                    -> $(SEC_CONFIG) ;
        /root/.bashrc                   -> $(SEC_CONFIG) ;
#       /root/.bash_profile             -> $(SEC_CONFIG) ;
#       /root/.bash_logout              -> $(SEC_CONFIG) ;
#       /root/.bash_history             -> $(SEC_CONFIG) ;
#       /root/.amandahosts              -> $(SEC_CONFIG) ;
#       /root/.addressbook.lu           -> $(SEC_CONFIG) ;
#       /root/.addressbook              -> $(SEC_CONFIG) ;
#       /root/.Xresources               -> $(SEC_CONFIG) ;
#       /root/.Xauthority               -> $(SEC_CONFIG) -i ; # Changes Inode number on login
#       /root/.ICEauthority             -> $(SEC_CONFIG) ;
        /root/.profile                  -> $(SEC_CONFIG) ;
}
...
#
# Critical devices
#
(
  rulename = "Devices & Kernel information",
  severity = $(SIG_HI),
  emailto = studentXY@spafXY.lan
)
{
        /dev            -> $(Device) ;
        ! /dev/bsg ;
#       /proc           -> $(Device) ;
}
...
#
# Other configuration files
#
(
  rulename = "Other configuration files",
  severity = $(SIG_MED),
  emailto = studentXY@spafXY.lan
)
{
    /etc        -> $(SEC_CONFIG) ;
    /etc/resolv.conf    -> $(SEC_CONFIG) -i ;
    ! $(TWETC) ;
}
...
#
# Binaries
#
(
  rulename = "Other binaries",
  severity = $(SIG_MED),
  emailto = studentXY@spafXY.lan
)
...
#
# Libraries
#
(
  rulename = "Other libraries",
  severity = $(SIG_MED),
  emailto = studentXY@spafXY.lan
)
...
#
# Commonly accessed directories that should remain static with regards
# to owner and group
#
(
  rulename = "Invariant Directories",
  severity = $(SIG_MED),
  emailto = studentXY@spafXY.lan
)
```

Test your tripwire email configuration:

```
root@spafXY:~# tripwire --test --email studentXY@spafXY.lan
```

Configure tripwire, build the database, and perform an initial check of the database:

```
root@spafXY:~# twadmin --create-polfile -S /etc/tripwire/site.key /etc/tripwire/twpol.txt
root@spafXY:~# tripwire --init
root@spafXY:~# tripwire --check
```

You can update the most recent tripwire database at any time by running the command:

```
root@spafXY:~# tripwire --update --twrfile /var/lib/tripwire/report/<name>.twr
```

11 Traffic analysis

Install some commonly used network tools and libraries (if you have not already done so):

```
root@spafXY:~# apt install libpcap-dev libdumbnet-dev tcpdump wireshark tshark
```

12 Extra Credit

The amount of extra credit assigned depends heavily on the amount of work you put into it. I highly recommend doing the first two options only while physically in the lab, as you can easily cut your physical machine off from the Internet, and I do not plan on going up to school to fix any issues.

1. 3 points - Install and configure `dhcp-server` to serve local IP addresses to wireless clients.
2. 2 points - Configure nftables, hostapd, etc... to perform NAT (IP Masquerading) so that wireless clients can access the Internet.
3. 4 points - Install and configure splunk, and splunk for snort, and create visualizations to view your snort logs:

   `https://splunkbase.splunk.com/app/340/`
4. 2 points - Install, configure, and demonstrate lynis, chkrootkit, rkhunter, and clamav.
5. 2 points - Update the snort rules, and view unified2 snort logs with Wireshark.
6. 2 points - Research and enforce sysctl settings to improve your security posture.

13 Grading

On the due date we will schedule an interview for you to demonstrate the work done on this assignment. Some of the grading will be done automatically by scripts in order to keep the interviews as short as possible, so please ensure your work is complete, correct, and working both on the due date and when the interview starts.