Note! This is a selection of tasks from last year, the current ones will be similar but modified. Provided only as an example, for better understanding!

Preparation of the environment

1. Account on the server **xor.math.uni.lodz.pl** (assuming: "frydrych" as the account administrator), it is obligatory to check the *login* and *password*
2. Set up *passing* subdirectory in your home directory:

   ```
   $ mkdir ~/passing
   ```

3. Check if you are a member of our *CSS* team (*BSK* for Polish students) at `MS_Teams` (mandatory).

Topics

1. A non-empty password for the user *root*
2. Create group *animals*

   Hint - Handbook/Users and Basic Account Management chapter 3.3

3. Create a users *dog* and *cat* in the group *animals*, first name and surname *"Rex Barks"* for user *dog* and *"Felix Meow"* for user *cat*, home directory with rights `0700`, shell `/bin/sh`, password like login

   Hint - Handbook/Users and Basic Account Management chapter 3.3

4. Edit the main configuration file `/etc/rc.conf`:

   ```
   hostname="LOGIN_FROM_XOR"  # replace
   ifconfig_em0="DHCP"
   ifconfig_em1="inet 192.168.56.201/24"
   ifconfig_em2="inet 172.31.255.99/24"
   sshd_enable="yes"
   pf_enable="yes"
   pflog_enable="yes"
   growfs_enable="yes"
   ```

   if above there are some extra lines, then rewrite them as well and *reboot* your machine

5. Check your VM's internet connection to the world and bi-directional connection to the host
6. Configure the *SSH server* (*sshd*, file `/etc/ssh/sshd_config`) changing option for remote user *root* login, via a *private/public* key pair only:

```
PermitRootLogin   prohibit-password
```

Remember to reload the service:

```
# service sshd reload
```

Hint - Handbook/OpenSSH chapter 16.7

7. Generate *SSH* keys for all users (*root*, *dog*, *cat*) algorithms: *ecdsa* and *ed25519* (no passwords)

```
$ ssh-keygen -t ecdsa
$ ssh-keygen -t ed25519
```

Hint - Handbook/OpenSSH chapter 16.7

8. Connect via *ssh* (use *Putty* and *PuttyGen* on *Host* and/or *ssh-keygen ...* on *VM*) **Host-> Guest** as users *dog* and *cat* without passwords, using a *private/public key* pairs

9. Connect via *ssh* (use *Putty* and *PuttyGen* on *Host* and/or *ssh-keygen ...* on *VM*) **Host-> Guest** as user *root* without password (mind the `PermitRootLogin prohibit-password` option in *sshd* configuration file), using a *private/public key* pair

10. Setup firewall *PF* (*PacketFilter*), configuration file `/etc/pf.conf`, *block/pass* for some selected services like *ssh, http, https*

    Hint - review the chapter: Handbook/Firewalls/PF Packet Filter, chapter 33.3

11. Try to defend against malicious attacks (DDoS attacks) on *ssh* service, by blocking such IP addresses (apply `max-src-conn` and `max-src-conn-rate` *pf* rules options)

    Hint - review the chapter: Handbook/Firewalls/PF Packet Filter, chapter 33.3

Extra, self-studying topics - for higher grade (> 4.0)

1. Setup web server `nginx` and try to enable *PHP* processor

```
# pkg install nginx php81 php81-extensions
...
# # adjust nginx configuration file, especially php section:
# ee /usr/local/etc/nginx/nginx.conf
...
```

add two extra lines to `/etc/rc.conf` file:

```
nginx_enable="yes"
php_fpm_enable="yes"
```

and start new services:

```
# service php-fpm start
# service nginx start
```

Hint - adjust almost ready configuration sampels in `/usr/local/etc/nginx` *VM* directory and/or Nginx documentation

2. Check *WWW* connections **Host-> Guest** (by browser, url: `http://192.168.56.201` or/and `http://192.168.56.201/info.php`)

```
# cat /usr/local/www/nginx/info.php
```

```
<?php
    phpinfo();
    exit( 0 );
?>
```

3. For the web server `nginx` setup the OpenSSL self-signed certificate (translate from Polish) and check secure/encrypted *HTTPS* connections **Host-> Guest** (by browser, url: `https://192.168.56.201` or/and `https://192.168.56.201/info.php`) adding *unknown issuer* browser-exception.

   Hint - review the chapter: Handbook/OpenSSL, chapter 15.6