Linux for New Hackers – Pt. II

Sheffield Ethical Student Hackers

SHEFFIELD Ethical Student Hackers

Breaking into security.

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.
- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.
- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

SHEFFIELD Ethical Student Hackers
Breaking into security.

Today's session – get stuck in

- This session we're going to achieve something useful with the basic Linux skills we picked up last week.
- I've created an AWS EC2 (virtual machine) instance for everybody, you'll be interacting with this machine's terminal via SSH.
- Each instance is running Ubuntu 18.04 Server Edition and features a whopping:
  - 512MB RAM
  - 1 shared CPU core
  - 8GB disk space
- You can do a surprising amount with such constrained resources!

SHEFFIELD Ethical Student Hackers
Breaking into security.

Code of Conduct

- Before proceeding past this point you must read and agree our Code of Conduct, this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.

- Code of Conduct can be found at https://wiki.shefesh.com/doku.php?id=code_conduct

SHEFFIELD Ethical Student Hackers
Breaking into security.

In addition to the Code of Conduct

- You have sudo rights on your instance, please remember with great power comes great responsibility.
- Your instance will be deleted after this session, if you'd like extended access, please ask me at the end – our society funds can cover learning resources.
- I have enabled network-level firewall rules that will limit inbound networking to source addresses originating from UoS – so please disable any VPNs you may be using (or inform me of the IP so I can whitelist it).
- Don't be afraid of breaking something unintentionally! It'll take less than 10 seconds to restore your instance back to a clean state.

SHEFFIELD Ethical Student Hackers
Breaking into security.

Getting connected

- If you're using Windows, PuTTY is a good SSH client for connecting to remote servers.
  - Use our download mirror or if you don't trust us, make sure you get it from the developer's site!
  - chiark.greenend.org.uk – putty.org is owned by a competitor, don't encourage domain squatting
- Linux/Mac, use the built in ssh client in the terminal
  - ssh shefeshuser@<IPADDRESS>

Today we'll be setting up a WordPress blog

- WordPress is a favourite for hacking into – we'll try and set up a *reasonably* secure WordPress installation
- WordPress is a CMS (content management system), it allows users to quickly and easily manage websites/blogs. It is written in PHP and stores data in a SQL database.
- To make WordPress function we will need the following:
- L*inux*
- A*pache*
- M*ySQL*
- P*HP*

SHEFFIELD Ethical Student Hackers
Breaking into security.

Just as a side-note

- I will be using *apt* as opposed to *apt-get* throughout this session. You will frequently come across *apt-get* used in tutorials. In most cases, *apt* is a drop-in replacement but is more beginner friendly, offering more human-friendly output
- Managing packages requires root permissions, so we need to prepend our commands with *sudo* when logged-in as the shefeshuser

Once we're logged in

- You might want to change your password! Recall from last week, this is the ***passwd*** command
- We should ensure the system is fully up-to-date.
- This is an Ubuntu (Debian-based) system, so repository and package management is done with *APT (advanced package tool)*
- First, refresh the package lists on our system, so APT knows if any packages need updating.
- *sudo apt update*
- If there any package updates available, APT will inform us at the end of the output
- *sudo apt upgrade*
- This will upgrade all packages on the system

Installing a web server

- We'll need an application to communicate over HTTP(S) and serve our files/web pages
- Lots of potential options, but apache2 is the most common
- Alternatives include nginx, lighttpd and LiteSpeed
- *sudo apt install apache2*
- You should now try to navigate to your instance in your web browser, you should see a test page demonstrating that apache2 was installed successfully!
- We'll also need to install PHP (and extensions) to allow WordPress to operate properly
- *sudo apt install php libapache2-mod-php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip*

A note on PHP extensions

- It's best practice to not install PHP extensions that aren't used, there isn't a definitive list for WordPress, but the ones listed in the previous slide are almost-always used
- We'll need to restart the apache2 service to load enable php to work. We'll use *systemctl* (interface for the systemd process manager)
- *sudo systemctl restart apache2*

Installing MySQL

- WordPress stores data in a relational database
- We'll be using MySQL for this, you could also use MariaDB
- As before *sudo apt install mysql-server*
- The default MySQL installation shouldn't be used as-is, instead, let's run:
- *sudo mysql_secure_installation*
- Go through this script and lock-down the install

Very basic SQL

- We're not going to go in depth with SQL, it isn't too difficult to pick up
- We need to create a database for WordPress to store its data in
- First, we should create a user for WordPress to interact with the database with
- *sudo mysql*
- This will open up a MySQL prompt, from here we can manipulate users, databases and tables.
- It's important to note that the SQL users are **unrelated** to the Linux users on your system
- But as with Linux, you shouldn't be using your root MySQL user for your WordPress site!

Initialising MySQL for WordPress

- *GRANT ALL ON wordpress.* TO ‘wpuser'@'localhost' IDENTIFIED BY 'password’;*
- Notice SQL statements must always be terminated with a semi-colon
- Technically, statements aren’t case-sensitive, but it’s good practice to capitalise commands
- Breakdown of the statement:
  - This statement will create a database called wordpress.
  - A user called wpuser will be created, and we will grant ALL permissions to manipulate this database (and any tables) to the wpuser
  - This wpuser has a password defined after IDENTIFIED BY – don’t use ‘password’!

Flushing privileges and exiting

- That's all we need to do regarding the database, WordPress will create and populate the tables
- We need to flush the privileges to apply our changes
- *FLUSH PRIVILEGES;*
- And
- *EXIT;*

Modifying apache2 settings

- WordPress and some plug-ins use *.htaccess* files to modify apache2 configuration
- By default, *.htaccess* files aren't processed by apache2, so we need to change some configurations
- We'll also want to enable Virtual Hosts, it allows apache2 to host multiple sites from one server (though we won't be doing that today)
- Let's create a directory for our WordPress site
- *sudo mkdir /var/www/wordpress*

Modifying apache2 settings

sudo nano /etc/apache2/sites-available/wordpress.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName xxx.sesh.systems
    DocumentRoot /var/www/wordpress
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

SHEFFIELD Ethical Student Hackers

Breaking into security.

Modifying apache2 settings

<Directory /var/www/wordpress/>

AllowOverride All

</Directory>

- Ctrl-X and save the file
- *sudo a2enmod rewrite*
- *sudo apache2ctl configtest*
- *sudo systemctl restart apache2*

Getting WordPress

- *cd /tmp*
- *wget https://wordpress.org/latest.tar.gz*
- *tar xzvf latest.tar.gz*
- *touch /tmp/wordpress/.htaccess*
- *cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php*
- *mkdir /tmp/wordpress/wp-content/upgrade*
- *sudo cp -a /tmp/wordpress/. /var/www/wordpress*

Fixing Permissions

- *sudo chown -R www-data:www-data /var/www/wordpress*
- *sudo find /var/www/wordpress/ -type d -exec chmod 750 {} \;*
- *sudo find /var/www/wordpress/ -type f -exec chmod 640 {} \;*
- *curl -s https://api.wordpress.org/secret-key/1.1/salt/*
- *sudo nano /var/www/wordpress/wp-config.php*
- *Make sure to add:*

  define('FS_METHOD', 'direct');