Editing Files

There are several ways to edit a file. One of the most common text editors for this is Vi and Vim. More rarely, there is the Nano editor. We will first deal with the Nano editor here, as it is a bit easier to understand. We can create a new file directly with the Nano editor by specifying the file's name directly as the first parameter. In this case, we create a new file named notes.txt.

```
amit8986@htb[/htb]$ nano notes.txt
```

Now we should see a so-called "pager" open, and we can freely enter or insert any text. Our shell should then look something like this.

Nano Editor

Below we see two lines with short descriptions. The caret (^) stands for our "[CTRL]" key. For example, if we press [CTRL + W], a "Search:" line appears at the bottom of the editor, where we can enter the word or words we are looking for. If we now search for the word "we" and press [ENTER], the cursor will move to the first word that matches.

```
Nano Editor

GNU nano 2.9.3 notes.txt

Here ∰we can type everything we want and make our notes.

Search: notes

^G Get Help M-C Case Sens M-B Backwards M-J FullJstify ^W Beg of Par ^Y First Line ^P PrevHstory

^C Cancel M-R Regexp ^R Replace ^T Go To Line ^O End of Par ^V Last Line ^N NextHstory
```

To jump to the next match with the cursor, we press [CTRL + W] again and confirm with [ENTER] without any additional information.

```
Nano Editor

GNU nano 2.9.3 notes.txt

Here we can type everything ∰we want and make our notes.

Search [we]:

^G Get Help M-C Case Sens M-B Backwards M-J FullJstify ^W Beg of Par ^Y First Line ^P PrevHstory

^C Cancel M-R Regexp ^R Replace ^T Go To Line ^O End of Par ^V Last Line ^N NextHstory
```

Now we can save the file by pressing [CTRL + 0] and confirm the file name with [ENTER].

```
GNU nano 2.9.3 notes.txt

Here we can type everything we want and make our notes.

File Name to Write: notes.txt

^G Get Help M-C Case Sens M-B Backwards M-J FullJstify ^W Beg of Par ^Y First Line ^P PrevHstory

^C Cancel M-R Regexp ^R Replace ^T Go To Line ^O End of Par ^V Last Line ^N NextHstory
```

After we have saved the file, we can leave the editor with [CTRL + X].

Back on the Shell

To view the contents of the file, we can use the command cat.

```
Back on the Shell

amit8986@htb[/htb]$ cat notes.txt

Here we can type everything we want and make our notes.
```

There are many files on Linux systems that can play an essential role for us as penetration testers whose rights have not been correctly set by the administrators. Such files may include the file "/etc/passwd".

VIM

Vim is an open-source editor for all kinds of ASCII text, just like Nano. It is an improved clone of the previous Vi. It is an extremely powerful editor that focuses on the essentials, namely editing text. For tasks that go beyond that, Vim provides an interface to external programs, such as grep, awk, sed, etc., which can handle their specific tasks much better than a corresponding function directly implemented in an editor usually can. This makes the editor small and compact, fast, powerful, flexible, and less error-prone.

Vim follows the Unix principle here: many small specialized programs that are well tested and proven, when combined and communicating with each other, resulting in a flexible and powerful system.

Vim

```
Vim
amit8986@htb[/htb]$ vim
```

```
Vim
1 $
                            VIM - Vi IMproved
                             version 8.0.1453
                         by Bram Moolenaar et al.
         Modified by pkg-vim-maintainers@lists.alioth.debian.org
               Vim is open source and freely distributable
                         Sponsor Vim development!
              type :help sponsor<Enter>
                                            for information
              type :q<Enter>
                                            to exit
              type :help<Enter> or <F1> for on-line help
              type :help version8<Enter>
                                           for version info
                                                                 0,0-1
                                                                               A11
```

In contrast to Nano, Vim is a modal editor that can distinguish between text and command input. Vim offers a total of six fundamental modes that make our work easier and make this editor so powerful:

Mode	Description
Normal	In normal mode, all inputs are considered as editor commands. So there is no insertion of the entered characters into the editor buffer, as is the case with most other editors. After starting the editor, we are usually in the normal mode.
Insert	With a few exceptions, all entered characters are inserted into the buffer.
Visual	The visual mode is used to mark a contiguous part of the text, which will be visually highlighted. By positioning the cursor, we change the selected area. The highlighted area can then be edited in various ways, such as deleting, copying, or replacing it.
Command	It allows us to enter single-line commands at the bottom of the editor. This can be used for sorting, replacing text sections, or deleting them, for example.
Replace	In replace mode, the newly entered text will overwrite existing text characters unless there are no more old characters at the current cursor position. Then the newly entered text will be added.

When we have the Vim editor open, we can go into command mode by typing ":" and then typing "q" to close Vim.

```
Vim

1 $

VIM - Vi IMproved

version 8.0.1453

by Bram Moolenaar et al.

Modified by pkg-vim-maintainers@lists.alioth.debian.org

Vim is open source and freely distributable

Sponsor Vim development!

type :help sponsor<Enter> for information

type :q<Enter> to exit

type :help<Enter> or <F1> for on-line help

type :help version8<Enter> for version info
```

Vim offers an excellent opportunity called vimtutor to practice and get familiar with the editor. It may seem very difficult and complicated at first, but it will only feel that way for a short time. The efficiency we gain from Vim once we get used to it is enormous.

VimTutor

```
VimTutor
amit8986@htb[/htb]$ vimtutor
```

VimTutor

______ Welcome to the VIM Tutor - Version 1.7 ______ Vim is a very powerful editor that has many commands, too many to explain in a tutor such as this. This tutor is designed to describe enough of the commands that you will be able to easily use Vim as an all-purpose editor. The approximate time required to complete the tutor is 25-30 minutes, depending upon how much time is spent with experimentation. ATTENTION: The commands in the lessons will modify the text. Make a copy of this file to practice on (if you started "vimtutor" this is already a copy). It is important to remember that this tutor is set up to teach by use. That means that you need to execute the commands to learn them properly. If you only read the text, you will forget the commands! Now, make sure that your Caps-Lock key is NOT depressed and press j key enough times to move the cursor so that lesson 1.1 completely fills the screen.

Optional Exercise:

Play with the vimtutor. Get familiar with the editor and experiment with their features.

Start Instance
0 / 1 spawns left

Waiting to start...

♣ Previous

Next →

V IVIa

✓ Mark Complete & Next

Cheat Sheet