



```
adhi@parrot: ~/hello time
```

File Actions Edit View Help

```
15.9 MiB / 15.9 MiB (100 %) 4.0 MiB/s in 3s ETA: 0s
info: installing component 'rust-std'
26.8 MiB / 26.8 MiB (100 %) 15.6 MiB/s in 1s ETA: 0s
info: installing component 'rustc'
66.9 MiB / 66.9 MiB (100 %) 18.6 MiB/s in 3s ETA: 0s
info: installing component 'rustfmt'
info: default toolchain set to 'stable-x86_64-unknown-linux-gnu'
```

```
stable-x86_64-unknown-linux-gnu installed - rustc 1.81.0 (eeb90cda1 2024-09-04)
```

Rust is installed now. Great!

To get started you may need to restart your current shell. This would reload your **PATH** environment variable to include Cargo's bin directory (`$HOME/.cargo/bin`).

To configure your current shell, you need to source the corresponding **env** file under `$HOME/.cargo`.

This is usually done by running one of the following (note the leading DOT):

```
. "$HOME/.cargo/env" # For sh/bash/zsh/ash/dash/pdksh
source "$HOME/.cargo/env.fish" # For fish
```

```
(adhi@parrot)-[~]  
$ source $HOME/.cargo/env
```

rustc 1.81.0 (eeb90cda1 2024-09-04)

```
(adhi@parrot)-[~]  
$ A[[200~cargo new hello_time  
zsh: bad pattern: ^[[200~cargo
```

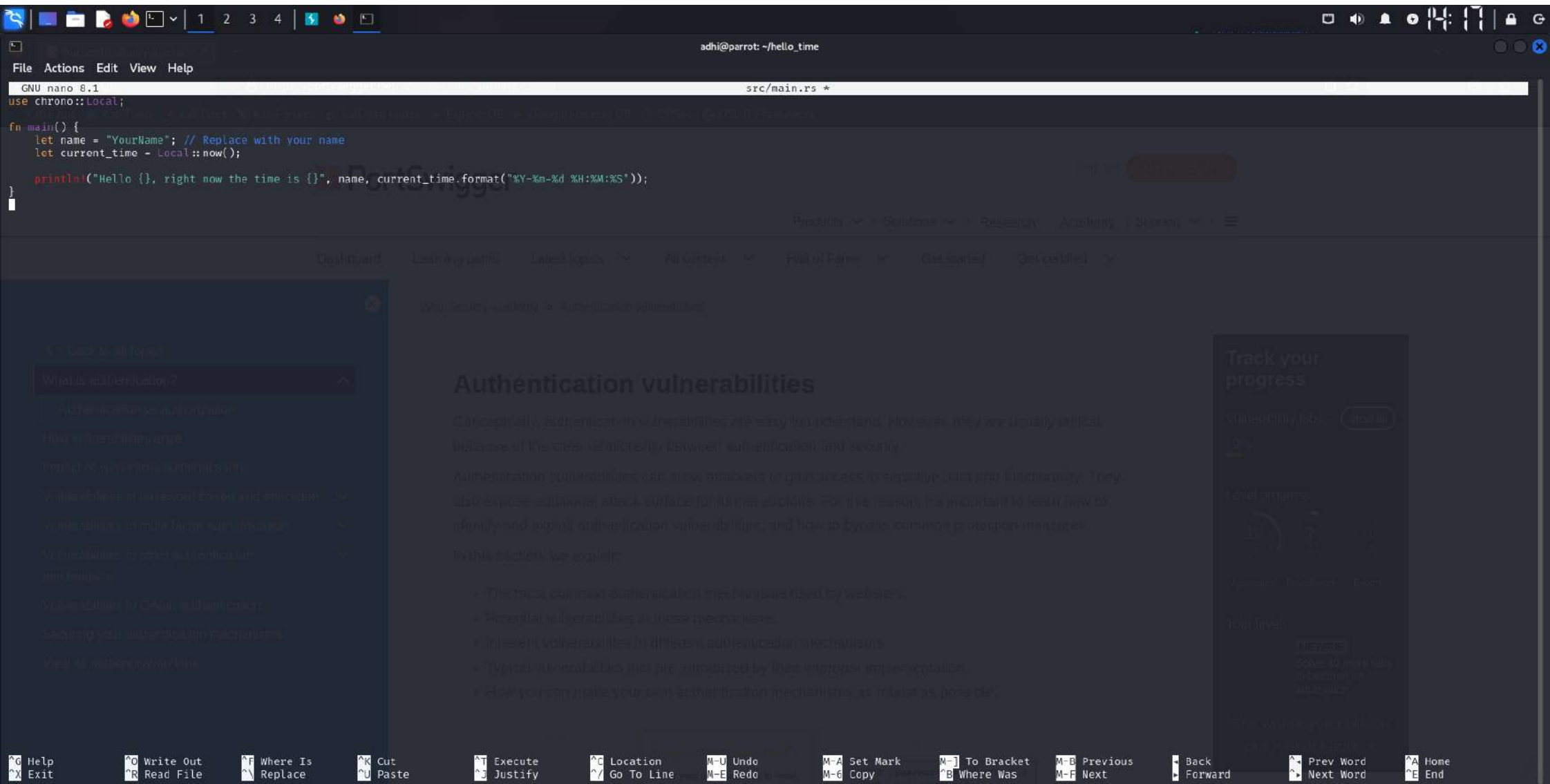
```
(adhi@parrot)-[~]  
$ cd hello_time  
cd: no such file or directory: hello_time
```

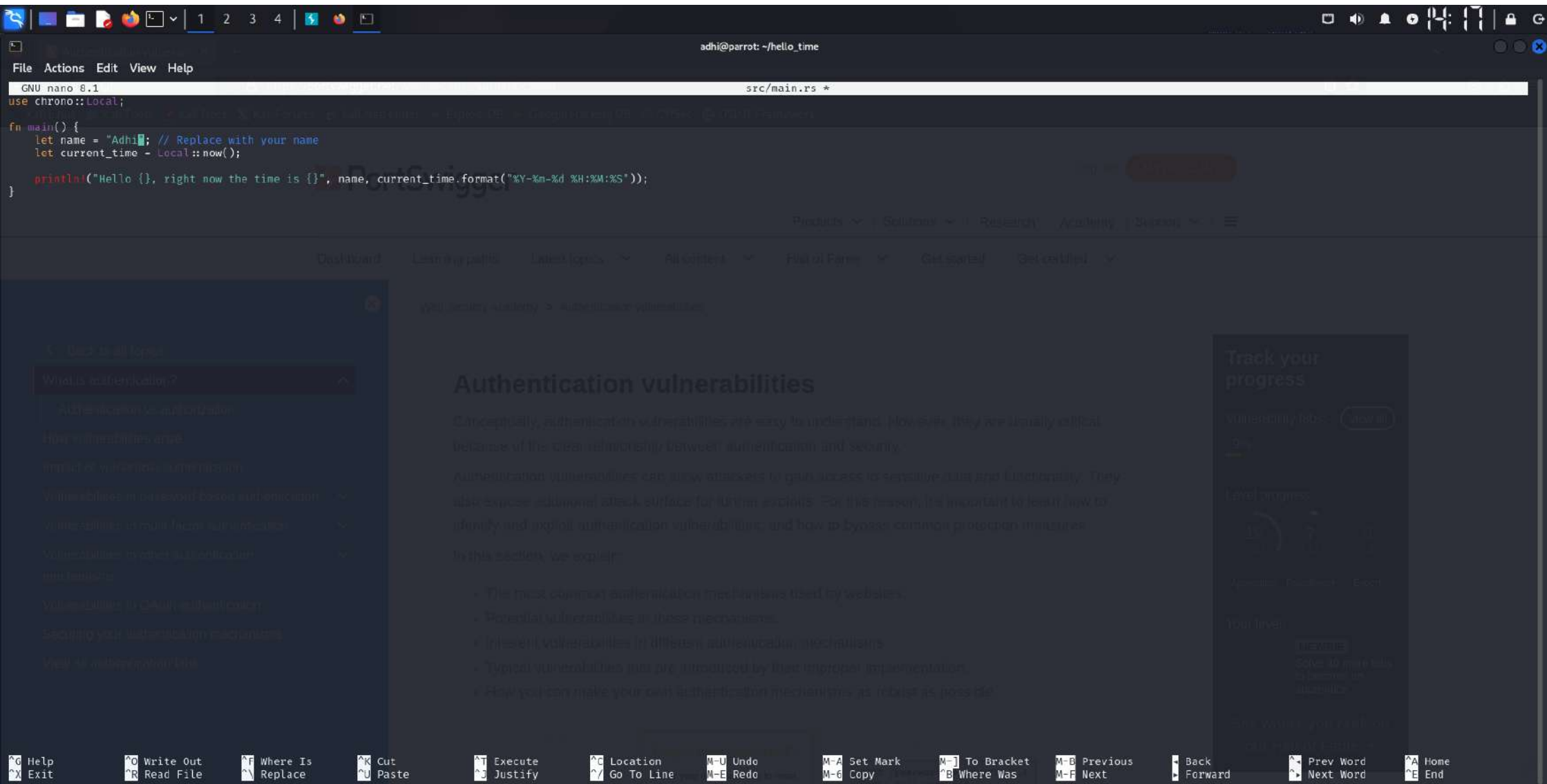
```
(adhi@parrot)-[~]  
$ cargo new hello_time  
cd hello_time
```

```
Creating binary (application) `hello_time` package
note: see more `Cargo.toml` keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html
```

```
(adhi@parrot)-[~/hello_time]
$ nano src/main.rs
```

```
(adhi@parrot)-[~/hello_time]
```





GNU nano 8.1

Cargo.toml *

[package]
name = "hello_time"
version = "0.1.0"
edition = "2021"

[dependencies]
chrono = "0.4"

adhi@parrot: ~/hello_time

File Actions Edit View Help

PortSwigger

Products Solutions Research Academy Store

Dashboard Learning paths Latest topics All content Hall of Fame Get started Get certified

Web Security Academy > Authentication vulnerabilities

Back to all topics

What is authentication?

Authentication vs authorization

How authentication works

Impact of authenticating users

Vulnerabilities in password-based authentication

Vulnerabilities in OAuth 2.0 authentication

Vulnerabilities in OpenID Connect authentication

Vulnerabilities in SAML authentication

Securing your authentication mechanisms

What is authentication flow

Authentication vulnerabilities

Conceptually, authentication vulnerabilities are easy to understand. However, they are usually critical because of the close relationship between authentication and security.

Authentication vulnerabilities can allow attackers to gain access to sensitive data and functionality. They also expose additional attack surface for future exploits. For this reason, it's important to learn how to identify and exploit authentication vulnerabilities, and how to bypass common protection measures.

In this section, we explore:

- The most common authentication mechanisms used by websites.
- Potential vulnerabilities in these mechanisms.
- Inherent vulnerabilities in different authentication mechanisms.
- Typical vulnerabilities you are introduced by their improper implementation.
- How you can make your own authentication mechanisms as robust as possible.

Track your progress

Vulnerability index

Level progress

Your level

NEW! Come to our talk at Blackhat Asia 2024

See what you can do with our new feature

Help Exit

Write Out Read File

Where Is Replace

Cut Paste

Execute Justify

Location Go To Line

Undo Redo

Set Mark Copy

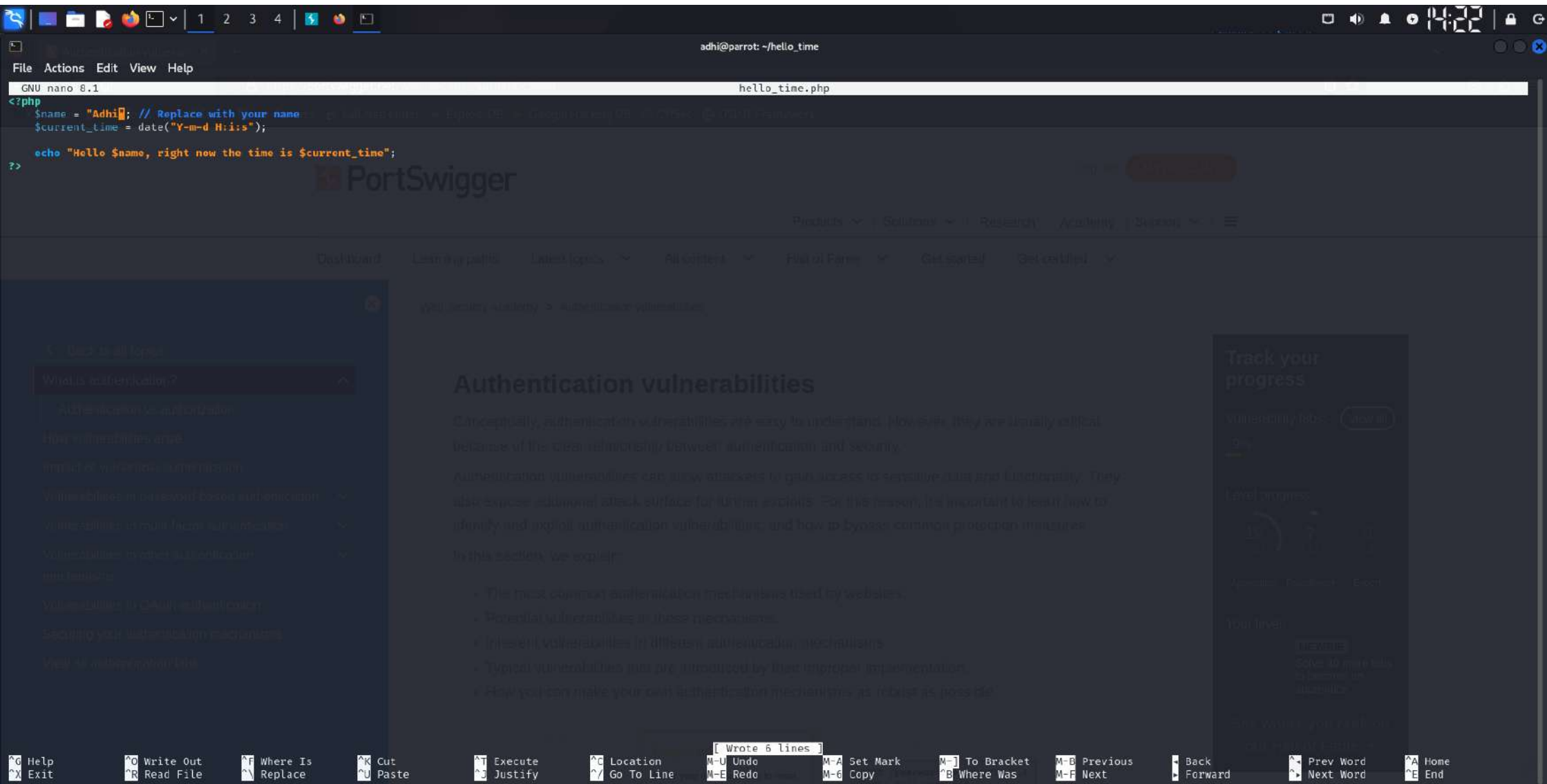
To Bracket Where Was

Previous Next

Back Forward

Prev Word Next Word

Home End





File Actions Edit View Help

(adhi@parrot)-[~/hello_time]

```
$ sudo apt update
sudo apt install php
```

[sudo] password for adhi:

```
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.1 MB]
Get:3 https://ngrok-agent.s3.amazonaws.com buster InRelease [20.3 kB]
Get:4 https://ngrok-agent.s3.amazonaws.com buster/main amd64 Packages [5,783 B]
Get:5 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.2 MB]
Get:6 http://kali.download/kali kali-rolling/contrib amd64 Packages [110 kB]
Get:7 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [269 kB]
Get:8 http://kali.download/kali kali-rolling/non-free amd64 Packages [193 kB]
Get:9 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [873 kB]
Get:10 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [33.1 kB]
Get:11 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [17.2 kB]
Fetched 70.8 MB in 21s (3,405 kB/s)
```

909 packages can be upgraded. Run 'apt list --upgradable' to see them.

php is already the newest version (2:8.2+93+nmul).

The following packages were automatically installed and are no longer required:

```
libdaxctl1 libjxl0.7 libpmem1 libre2-10 libsvtav1enc1d1 libx265-199 openjdk-21-jre-headless python3-mistune0 python3-pytzdata samba-dsdb-modules
libgeos3.12.1t64 libndctl6 libray1e0 libroc0.3 libu2f-udev openjdk-21-jre python3-diskcache python3-pendulum samba-ad-provision
```

Use 'sudo apt autoremove' to remove them.

Summary:

Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 909

(adhi@parrot)-[~/hello_time]

```
$ php --version
```

```
PHP 8.2.21 (cli) (built: Jul 25 2024 10:12:09) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.21, Copyright (c) Zend Technologies
with Zend OPcache v8.2.21, Copyright (c), by Zend Technologies
```

(adhi@parrot)-[~/hello_time]

```
$ nano hello_time.php
```

(adhi@parrot)-[~/hello_time]

```
$ php hello_time.php
```

Hello Adhi, right now the time is 2024-09-18 08:52:58.: not enough arguments

(adhi@parrot)-[~/hello_time]

```
$ php hello_time.php
```

Hello Adhi, right now the time is 2024-09-18 08:53:08

(adhi@parrot)-[~/hello_time]

```
$
```

Auth

Authentication

Authentication vulnerabilities

Conceptually, authentication vulnerabilities are easy to understand. However, they are usually difficult because of the deep relationship between authentication and security.

Authentication vulnerabilities can allow attackers to gain access to sensitive data and functionality. They also expose additional attack surface for further exploits. For this reason, it's important to learn how to identify and exploit authentication vulnerabilities, and how to bypass common protection measures.

In this section, we explore:

- The most common authentication mechanisms used by websites.
- Potential vulnerabilities in these mechanisms.
- Typical vulnerabilities and are associated by their improper implementation.
- How you can make your own authentication mechanisms as robust as possible.

Learn more about this topic

Enter your email address to read

ENTER YOUR EMAIL ADDRESS TO READ

Track your progress

Vulnerability Hub

9/9

Learn more

100% Complete

Upgrade to Premium

100% Done

NEW

Save to your list of favorites

Get more information

Get more information