>_WCNIS:~ francescocarlucci$

security by design in WordPress

>_WCNIS:~ francescocarlucci$

hi, I am francesco

full stack developer
WordPress expert at Codeable

>_WCNIS:~ francescocarlucci$

let’s start:

how to secure a WordPress site

```
>_WCNIS:~ francescocarlucci$


< just joking :P >
  -----------------
          \    ^__^
           \   (oo)\_________
               (__)\         )\/\
                   ||-----w  |
                   ||        ||
```

>_WCNIS:~ francescocarlucci$

this talk is about

securty by design

>_WCNIS:~ francescocarlucci$

is WP secure by design?

About 406,000,000 results (0.47 seconds)

>_WCNIS:~ francescocarlucci$

what is security by design?

>_WCNIS:~ francescocarlucci$

1. minimise attack surface area
2. establish secure defaults
3. the principle of Least privilege (POLP)
4. the principle of Defence in depth
5. fail securely

>_WCNIS:~ francescocarlucci$

6. don’t trust services
7. separation of duties
8. avoid security by obscurity
9. keep security simple
10. fix security issues correctly

>_WCNIS:~ francescocarlucci$

back to WordPress...

minimise attack surface area

aka **do not overdo with plugins**

```
>_WCNIS:~ francescocarlucci$
```

establish secure defaults

echo get_post_meta() <- is that secure? :/

>_WCNIS:~ francescocarlucci$

establish secure defaults

password rotation

WP does not, should I?

>_WCNIS:~ francescocarlucci$

the principle of least privilege (POLP)

be careful with roles

>_WCNIS:~ francescocarlucci$

the principle of defence in depth

is not only about WordPress

>_WCNIS:~ francescocarlucci$

separation of duties

starting from remote backups :)

>_WCNIS:~ francescocarlucci$

so, the right question would be:

can *your* WP env be secure by design?

>_WCNIS:~ francescocarlucci$

but wait! can multiple entities be

responsible for security of one object?

>_WCNIS:~ francescocarlucci$

the **shared** responsability model

>_WCNIS:~ francescocarlucci$

thank you

and keep up the secure stuff :)