

CyberSecurity: Principle and Practice

*BSc Degree in Computer Science
2024-2025*

Lesson 9: Language Vulnerabilities

Prof. Mauro Conti

Department of Mathematics

University of Padua

conti@math.unipd.it

<http://www.math.unipd.it/~conti/>

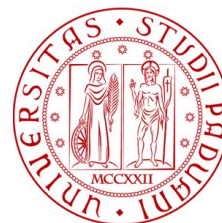
Teaching Assistants

Tommaso Bianchi

tommaso.bianchi@phd.unipd.it

Riccardo Preatoni

riccardo.preatoni@phd.unipd.it



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP



DIPARTIMENTO¹
MATEMATICA

- How many of you think about security during a system deployment?

- How many of you think about security during a system deployment?
- Hope some of you ...
- But what about the security derived from the program language that you are using?

- Program Languages are well known for several security threats that they provide
- Some functions might expose your application to threats
- It is a good practice to be aware of these risks
 - to prevent attacks

- C is kind of the father of every program language
- it is considered High Level
- several stuffs are left to the programmer
 - e.g., memory management (variable allocations / deallocations)
- several threats can be found
 - for a better description see [link](#)
- We go in details during the pwn topic at the end of the course

“gets()” function

- reads infinite characters given a stream and stores them in the string str
- in the example, what happens if we insert more than 15 characters?
 - ***memory corruption***

```
char buff[15];  
  
int pass = 0;  
  
printf("\n Enter the password : \n");  
gets(buff);
```

“strcpy()” function

- copy the characters contained in src to trg
- in the example, what happen if we copy more than 10 characters?
 - **memory corruption**

```
1 char str1[10];  
2 char str2[]="WeWantToOverwriteMemory";  
3 strcpy(str1, str2);
```

- PhP, as JS and Python, is a dynamically typed programming language
- the variables types are checked at runtime
- this sometimes can be a problem ...
- ("7 puppies" == 7) -> True
- see more at [link1](#) and [link2](#)

```
$example_int = 7  
  
$example_str = "7"  
  
if ($example_int == $example_str) {  
  
    echo("PHP can compare ints and strings.")  
  
}
```


1. Identify the programming language used in the application
2. Identify the version
3. Identify possible libraries used
4. Check on Google for possible vulnerabilities

1. You are given the code of a webapp. What is it hiding?
2. Because creating real pwn challs was to mainstream, we decided to focus on the development of our equation solver using OCR
3. The authors tried to protect their JS code ... is that enough to scare an attacker?

Questions? Feedback? Suggestions?



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

