

# Low level attacks

## Shellcode (part 2)

Mario Alviano

University of Calabria, Italy

A.Y. 2019/2020

# Exploiting is a very slow process!

- We have to encode addresses in little-endian
- Those addresses changes very easily
- Plus, we have to compute a few offsets

Can we improve a bit the process?

- So far, we disabled ASLR
- Still, the addresses retrieved via `gdb` do not match

## Why?

- The stack also contains environment variables
- Check it on `addresses.c`
- Run with and without `gdb`
- The environment is different!

- If we are starting the process, we can also choose the environment
- Let's simplify the environment!

## Ignore the environment

- Use `env -i` to start the process (use absolute paths)
- Try again `addresses.c`:  

```
$ env -i /tmp/a.out  
$ gdb a.out  
(gdb) set exec-wrapper env -i  
(gdb) r
```
- If you need some environment variable, add it inline  
eg. `$ env -i SHELL="/bin/sh" /tmp/a.out`

# Pack memory with Python

- Most of the machines we use are little-endian
- Humans are error-prone on repetitive tasks
- Better to automate the conversion!

## Pythonic solution

```
import struct
struct.pack("<I", address)
```

```
malvi@pandora:~$ python -c 'import struct; print(struct.pack("<I", 0x41424344))'
DCBA
```

- There are a few enhancement for `gdb`
- They add pretty printing functionalities
- One of them is `peda`
- Execute `source <path to peda.py>` in `gdb`
- Try on `victim.c`

Download `peda` from github

<https://github.com/longld/peda>

You may want to add the following lines to `.gdb_init`

- **Disable `less` for long output**  
`set pagination off`
- **Keep a history of all the commands typed (search with `ctrl-r`)**  
`set history save on`  
`set history filename ~/.gdb_history`  
`set history size 32768`  
`set history expansion on`





- Use `pattern search` to find the address of the pattern

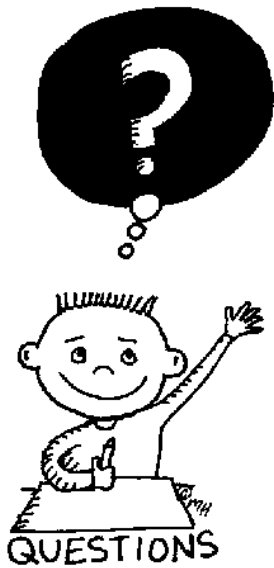
```
gdb-peda$ pattern search
Registers contain pattern buffer:
EIP+0 found at offset: 524
EBP+0 found at offset: 520
Registers point to pattern buffer:
[EDX] --> offset 1018 - size ~6
[ESP] --> offset 528 - size ~203
[EAX] --> offset 1018 - size ~6
Pattern buffer found at:
0xffffcc10 : offset    0 - size 1024 ($sp + -0x210 [-132 dwords])
0xffffd0b6 : offset    0 - size 1024 ($sp + 0x296 [165 dwords])
References to pattern buffer found at:
0xffffcbf0 : 0xffffcc10 ($sp + -0x230 [-140 dwords])
0xffffcc00 : 0xffffcc10 ($sp + -0x220 [-136 dwords])
0xffffcc04 : 0xffffd0b6 ($sp + -0x21c [-135 dwords])
gdb-peda$
```

# Build skeletons for your exploits

- Use strings of the same length
- Script your exploit as much as possible
- Try `skeleton.shellcode.py` on `victim.c`

## When you cannot start the process

- You have to try several addresses
- Again, a script may help you!



END OF THE  
LECTURE