Software Security 04

pwntools

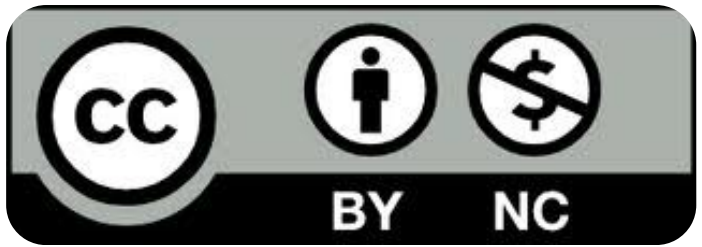
Gianluca Amato
Università di Chieti-Pescara

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the Creative Commons BY-NC License



To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

© CINI – 2021, Gianluca Amato – 2025

Pwntools

- Pwntools è una libreria per lo sviluppo di exploit
  - scritta in Python;
  - ha lo scopo di rendere la scrittura di exploit la più semplice possibile.
- Un exploit è una procedura per sfruttare una vulnerabilità allo scopo di eseguire operazione non autorizzate.
  - Vedi slide ufficiali della cyberchallenge per la lezione SS_1.1.
- La libreria è molto sofisticata:
  - noi ne vedremo solo una piccola parte;
  - sono disponibili dei tutorial online molto più completi.

© CINI – 2021, Gianluca Amato – 2025

Cosa vuol dire pwn ?

- pwn è un termine che viene dalla comunità gaming e hacker
  - Secondo Urban Dictionary è nato come errore tipografico di un designer di mappe per *Warcraft*, che ha scritto “have been pwned” invece di “have been owned” (per dire che è stato sconfitto, dominato)
  - Il termine è stato ripreso dalla comunità hacker per indicare che il proprio sistema è ora sotto il controllo di un’altra persona.

© CINI – 2021, Gianluca Amato – 2025

Tubi (1)

- Per sviluppare un exploit per un programma, bisogna interagire con esso.
- Pwntools, mette a disposizione vari meccanismi per connettersi ad un programma
  - tutti astratti nel concetto di **tubo**;
  - implementati nel modulo `pwnlib.tubes`
- Usando i tubi è possibile comunicare con
  - Processi locali
  - Sessioni TCP o UDP remote
  - Processi in esecuzione su server remoti tramite SSH
  - Porte seriali

© CINI – 2021, Gianluca Amato – 2025

Tubi (2)

- Tramite i tubi è possibile
  - Inviare dati
    - `send(data)`: inviare dati
    - `sendline(line)`: inviare dati seguiti da un carattere di andata a capo (newline)
  - Ricevere dati
    - `recv(n)`: ricevere un dato numero di byte
    - `recvline()`: ricevere dati fino a trovare un carattere newline
    - `recvuntil(delim)`: ricevere dati fino a trovare il delimitatore
    - `recvregex(pattern)`: ricevere dati fino a che non viene soddisfatta una espressione regolare
    - `recvrepeat(timeout)`: ricevere dati ripetutamente finché non si verifica un timeout
    - `clean()`: elimina tutti i dati nel buffer di lettura
  - Trasformare interi in sequenze di byte e viceversa
    - `pack(int)`:
    - `unpack()`:

© CINI – 2021, Gianluca Amato – 2025

Processi (1)

- Un tubo può essere usato per interagire con un processo passando:
  - il nome del processo;

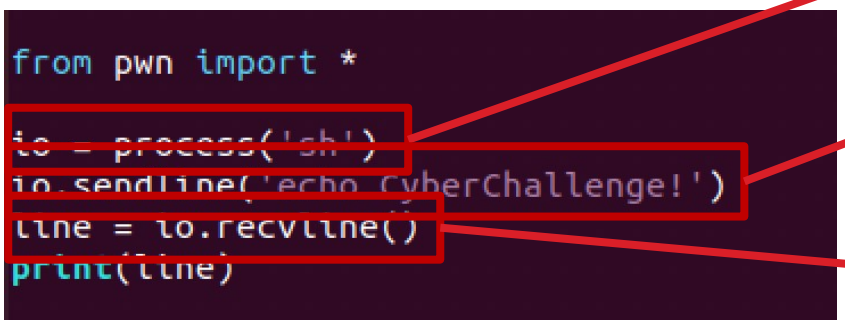
```
io = process('sh')
```

  - oppure, la lista di argomenti su riga di comando e l'ambiente

```
io = process(['sh', '-c', 'echo $MYENV'], env={'MYENV': 'MYVAL'})
```


© CINI – 2021, Gianluca Amato – 2025

Processi (2)



Esegue una *shell*

Invia un comando

Riceve il risultato

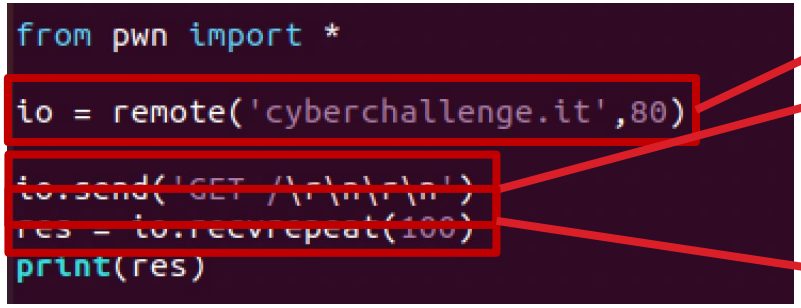
© CINI – 2021, Gianluca Amato – 2025

Networking (1)

- Con *pwntools* è possibile creare connessioni di rete o aspettare l'arrivo di connessioni in ingresso.
  - `remote` si può usare per aprire una connessione a un sever remote;
  - `listen` è usato per aspettare un connessione in ingresso.
- Sono supportate sia connessioni TCP che UDP.

© CINI – 2021, Gianluca Amato – 2025

Networking (2)



Apre una connessione

Invia una richiesta

Riceve la risposta

© CINI – 2021, Gianluca Amato – 2025

SSH

- Pwntools supporta l'interazione con processi remoti tramite una sessione SSH.

```
session = ssh('username', 'hostname', password='password')
```

- Una volta creata una sessione, si può eseguire un processo come su un sistema locale:

```
io = session.process('sh')
```


© CINI – 2021, Gianluca Amato – 2025

Altre caratteristiche

- Un modulo di supporto con funzioni per
  - lettura e scrittura di file;
  - funzioni di hashing (md5, sha1, …) e codifica (base64, url encoding, …);
  - generazione di pattern.
- Manipolazione di file ELF.
- Assemblaggio e disassemblaggio.

© CINI – 2021, Gianluca Amato – 2025

Log delle comunicazioni

- Quando il vostro script non funziona, o quando è in fase di sviluppo, può essere conveniente mostrare lo scambio dei messaggi nel tubo.
- Per mostrare tutti i messaggi scambiati:
  - `context.log_level="debug"`

© CINI – 2021, Gianluca Amato – 2025

Demo

Svolgere la challenge

SS_0.05 - piecewise

© CINI – 2021, Gianluca Amato – 2025

Installare pwntools

- Tramite i pacchetti della vostra distribuzione Linux
  - `apt install python3-pwntools`
- Tramite il comando pip:
  - `pip install pwntools`
  - A seconda del sistema operativo che avete installato, potreste dover prima creare un ambiente virtuale Python.

© CINI – 2021, Gianluca Amato – 2025

Esercizi

Automatizzare la challenge (facile)

SS_2.01 Digital billboard

Già risolta, adesso si tratta di automatizzare l'exploit con pwntools.

© CINI – 2021, Gianluca Amato – 2025

Esercizi

Svolgere le challenge (medie)

Software 17 – Pwntools 1

Software 18 – Pwntools 2

© CINI – 2021, Gianluca Amato – 2025

Software Security 04

pwntools

FINE

Gianluca Amato
Università di Chieti-Pescara

https://cybersecnatlab.it