



×



CYBERCHALLENGE.IT WORKSHOP 2025

Hosting scalabile di una CTF Jeopardy

Augusto Zanellato

 pwnlentoni.team

\$ whoami

- LT Informatica @ UniPD
- LM Cybersecurity @ UniPD (wip)
- Finalista CyberChallenge @ UniPD 2022
- Tutor CyberChallenge @ UniPD 2023-2025

CTF Player @ pwnlentoni && about:blankets :mr96arrabbiato:

- web + misc
- A/D tooling & infra
- sysadmin

Background


Il 21/06/2025 abbiamo hostato la nostra prima CTF:



cornCTF 2025



Alcune stats:

- ~400 utenti registrati
- ~270 utenti hanno risolto ≥ 1 challenge
- ~2300 flags processate
- ~500 flags valide
- ~1800 istanze di challenge avviate
- ~200 connessioni contemporanee alla piattaforma per tutto l'evento
- **0 downtime!** 

Come?

- Nessuna esperienza pregressa nell'organizzazione di CTF
 - ▶ Nessuna previsione di traffico
- Necessità di infrastruttura scalabile!

Come?

- Nessuna esperienza pregressa nell'organizzazione di CTF
 - ▶ Nessuna previsione di traffico
- Necessità di infrastruttura scalabile!
 - ▶ Kubernetes!

Kubernetes?

Concettualmente simile a Docker: permette il deployment di containers e molto altro

Kubernetes?

Concettualmente simile a Docker: permette il deployment di containers e molto altro

Pros:

- **Cluster:** permette di distribuire il lavoro su più server;
- **API potenti:** molto estensibili *by design*, permettono di gestire risorse custom;
 - ▶ **RBAC** (Role Based Access Control): Controllo granulare dei permessi;
- **Controllo granulare del traffico:** si possono definire *policy* per controllare la comunicazione tra container;

Kubernetes?

Concettualmente simile a Docker: permette il deployment di containers e molto altro

Pros:

- **Cluster:** permette di distribuire il lavoro su più server;
- **API potenti:** molto estensibili *by design*, permettono di gestire risorse custom;
 - ▶ **RBAC** (Role Based Access Control): Controllo granulare dei permessi;
- **Controllo granulare del traffico:** si possono definire *policy* per controllare la comunicazione tra container;

Cons:

- **Complessità:** maggiore rispetto alla gestione di un host che esegue Docker;
- **Uso di risorse:** maggiore dato che K8s ha più “componenti” rispetto a Docker;

Estensibilità di Kubernetes

Permette di definire **oggetti custom** (CRD: Custom Resource Definition):

- Le istanze dei CRD vengono gestite dagli **Operator**:
 - ▶ solitamente gli Operator mappano un'istanza di CRD ad altri oggetti;
- La creazione ed ogni modifica di istanze di CRD inizia un *reconcile*:
 - ▶ *Generazione* di altre risorse a partire dal CRD;
- Gli Operator possono inoltre validare i dati dei CRD al loro inserimento:
 - ▶ Possono restituire warning o annullare l'inserimento;

Questa possibilità rende Kubernetes adattabile a moltissimi tipi di *workload* diversi.

Requisiti funzionali

- Challenge istanziate, possibilmente instancer integrato nella piattaforma CTF;
- Tutte le challenge servite over TLS, routando il traffico in base all'hostname:
 - ▶ Richiede di esporre una singola porta ai giocatori;
 - ▶ Evita eventuali *brutini* sulle porte per accedere a istanze di altri team, bisognerebbe “indovinare” l'hostname: difficile;
- Le challenge non devono poter accedere a cose “sensibili”:
 - ▶ altre challenge;
 - ▶ API Kubernetes;
 - ▶ Metadata endpoint;
- Non deve esplodere durante i primi 5 minuti! e possibilmente per il resto della CTF

Piattaforma CTF

Dopo molte ricerche abbiamo scelto CTFd

Piattaforma CTF

Dopo molte ricerche abbiamo scelto CTFd

Cons:

- Performance non eccellenti;
- Codebase non particolarmente leggibile o documentata;

Piattaforma CTF

Dopo molte ricerche abbiamo scelto CTFd

Cons:

- Performance non eccellenti;
- Codebase non particolarmente leggibile o documentata;

Pro:

- Supporto a **plugin** per aggiungere funzionalità!

Piattaforma CTF

Dopo molte ricerche abbiamo scelto CTFd

Cons:

- Performance non eccellenti;
- Codebase non particolarmente leggibile o documentata;

Pro:

- Supporto a **plugin** per aggiungere funzionalità!


Quello è davvero l'unico motivo per cui l'abbiamo scelto

CTFd Tuning Guide

- Aumentare `SQLALCHEMY_MAX_OVERFLOW` per gestire più connessioni;
- Usare cache **Redis**;
- **Storage S3-like** per file: AWS S3 / Google GCS / Cloudflare R2;
- Di norma per gestire più richieste basta allocargli **più CPU**, non ha particolari richieste di RAM;
 - ▶ `WORKERS = n_{core} * 2 + 1` thanks @simonedimaria ❤️
- Abbiamo allocato a CTFd due macchine 8 core, 32GB ram:
 - ▶ Ogni macchina gestisce 500 req/s con latenza < 100ms;
 - ▶ In realtà una sola macchina era più che sufficiente;

Database per CTFd

MariaDB Operator per gestire il database *the Kubernetes way*:

- Permette di creare e configurare database usando **CRD**;
- Permette di configurare facilmente un cluster Galera:
 - ▶ Non l'abbiamo usato in quanto dai test bastava un singolo nodo "grosso";
- Facile configurazione di backup periodici;
- Nodo dedicato su cui gira solo il database:
 - ▶ I/O performance  ;
- Aumentare `max_connections`:
 - ▶ Almeno $\text{CTFd}_{\text{\$SQLALCHEMY_MAX_OVERFLOW}} * \text{CTFd}_{\text{\$WORKERS}} * \text{CTFd}_{\text{replicas}}$

Traffico in ingresso

Per tutto il traffico in ingresso abbiamo usato **Traefik**:

- Reverse proxy;
- Supporta sia **HTTP** che **TCP**, anche over-TLS;
- **Supporto Kubernetes** nativo;
- Ottime performance;
- Uso di risorse contenuto;
- Monitorabile facilmente;
- **ATTENZIONE:** *path normalization* non disattivabile! Avrei voluto saperlo prima...

Traffico in ingresso

- Un'istanza di Traefik per ogni nodo:
 - ▶ DaemonSet in Kubernetes;
- È stato configurato un load balancer davanti a tutte le istanze Traefik:
 - ▶ Fornito dal cloud provider, ma in alternativa MetalLB o KlipperLB;
- Uso di **Gateway API** per *service discovery*;
- Uso di **cert-manager** per ottenere i certificati TLS:
 - ▶ `play.cornc.tf` per CTFd;
 - ▶ `*.challs.cornc.tf` per le challenge;

Ok, ma le challenge?

Abbiamo analizzato alcune delle soluzioni preesistenti:

- **TJCSec/Klodd:**
 - ▶ Supporta solo rCTF con delle patch applicate;
 - ▶ Network policy non abbastanza restrittive (consente accesso a metadati);
 - ▶ Bisogna esplicitamente disattivare i *service account* per ogni challenge
- **DUCTF/kube-ctf:**
 - ▶ Espone API, quindi integrabile in qualsiasi piattaforma;
 - ▶ I template delle challenge sono assai verbosi da scrivere: bisogna scrivere esplicitamente Deployment, Service, IngressRule per ogni container;
 - ▶ Deploya tutte le istanze nello stesso namespace 🤔 ;

Ok, ma le challenge? - 2

- CTFer.io/**chall-manager**:
 - ▶ Espone API;
 - ▶ I template delle challenge sono scritti in Go usando Pulumi:
 - Troppo complesso, anche se estremamente flessibile;
- google/**kCTF**:
 - ▶ Operator Kubernetes;
 - ▶ Estremamente legato a Google Cloud Platform:
 - GCP sponsorizza l'infrastruttura a molte CTF, ma vogliamo evitare *vendor lock-in*;

Quindi?

Abbiamo sviluppato il nostro sistema di deploy per le challenge!

Quindi?

Abbiamo sviluppato il nostro sistema di deploy per le challenge!

pwnlenti/prism-ctf! *insert kube->prism joke here*

- Operator per Kubernetes;
- Supporta challenge condivise tra tutti i giocatori ma anche challenge istanziate!
- Indipendente dalla piattaforma per CTF:
 - ▶ Abbiamo sviluppato un plugin per CTFd; esperienza diciamo “peculiare”
- I template delle challenge non sono eccessivamente verbosi;
- Ogni istanza di challenge è in un namespace isolato;
- Network policy restrittive;
- *Service account* disattivati di default;

Alcuni esempi di template

Template per una challenge condivisa tra tutti i team:

```
apiVersion: prism-ctf.pwnlenti.team/v1
kind: SharedChallenge
metadata:
  name: aeronaut
spec:
  flag: "corn{1_d0n7_g4mb13_i_a1w4y5_w1n}"
  containers:
    - ports: # porte esposte servizio
      - port: 5000
        proto: TCP
  replicas: 1 # optional
  egress: false # no traffico in uscita
  spec: # core/v1 Container
    name: backend
    image: registry.lan/aeronaut:latest
    resources:
      limits:
```

```
    cpu: 4000m
    memory: 8000Mi
  requests:
    cpu: 2000m
    memory: 4000Mi
  env:
    - name: FLAG
      valueFrom:
        configMapKeyRef:
          key: flag
          name: flag
  exposes: # porte esposte ai player
    - container: backend
      port: 5000
      protocol: HTTP
```

Alcuni esempi di template

Template per una challenge istanziata:

```
apiVersion: prism-ctf.pwnlentoni.team/v1
kind: IsolatedChallenge
metadata:
  name: simple-chat
spec:
  # template per flag univoche
  flag_template: "corn{flag_{{ random 8 }}}}"
  lifetime: 10m
  containers:
    - ports: # porte esposte servizio
      - port: 3000
        proto: TCP
    replicas: 1 # opzionale, default 1
    egress: false # no traffico in uscita
    spec: # core/v1 Container
      name: backend
      image: registry.lan/simple-chat
```

```
resources:
  limits:
    cpu: 100m
    memory: 128Mi
  requests:
    cpu: 50m
    memory: 64Mi
env:
- name: __CHAL_HOST
  # simple-chat-$ID.challs.corn.c.tf
  valueFrom:
    configMapKeyRef:
      name: services
      key: chall
- name: CHALLENGE_HOST
  value: "https://${__CHAL_HOST}"
```



```
- name: FLAG
  valueFrom:
    configMapKeyRef:
      key: flag
      name: flag
- name: HEADLESS_HOST
  value: http://manager.headless
- ports: # porte esposte servizio
  - port: 5432
    proto: TCP
    egress: false
  spec:
    name: db
    image: postgres:17.4
    # resources: ...
    # env: ...
exposes: # porte esposte ai player
- container: backend
  port: 3000
  protocol: HTTP
```

Gestione delle istanze - 1

Le piattaforma CTF per creare un'istanza deve creare un oggetto Kubernetes del tipo:

```
apiVersion: prism-ctf.pwnlentoni.team/v1
```

```
kind: ChallengeInstance
```

```
metadata:
```

```
  name: simple-chat-1 # challName-teamId, ma può avere qualsiasi formato
```

```
spec:
```

```
  challenge: simple-chat # metadata.name della challenge
```

```
  team: "1" # id/nome del team, usato per limitare massimo istanze
```

Gestione delle istanze - 2

L'Operator riceve l'oggetto, lo valida e procede ad effettuare il deploy dell'istanza per poi mettere nello status dell'oggetto l'URL a cui è disponibile la challenge.

```
apiVersion: prism-ctf.pwnlentoni.team/v1
kind: ChallengeInstance
metadata:
  name: simple-chat-1
spec:
  # ...
  expiration: "2025-07-05T11:21:23Z"
  flag: corn{flag_ec198a7c276245e1}
status:
  exposedUrls:
  - hostname: simple-chat-e8185d21.challs.cornc.tf
    protocol: HTTP
```

Integrare piattaforme CTF

Tutto l'*heavy lifting* è fatto dall'Operator, quindi l'integrazione di altre piattaforme per CTF è *facile*:

- Recupero degli `exposedUrLs` per le challenge (sia istanziate che condivise);
- Gestione degli oggetti `challengeInstance`:
 - ▶ Watch delle modifiche;
 - ▶ Extend;
 - ▶ Recupero della flag generata per l'istanza;
 - ▶ Eventuale cancellazione quando la challenge viene risolta;
- Opzionalmente gestire l'upload di `sharedChallenge` e `IsolatedChallenge` su Kubernetes;

Integrare piattaforme CTF

Tutto l'*heavy lifting* è fatto dall'Operator, quindi l'integrazione di altre piattaforme per CTF è *facile*:

- Recupero degli `exposedUrLs` per le challenge (sia istanziate che condivise);
- Gestione degli oggetti `challengeInstance`:
 - ▶ Watch delle modifiche;
 - ▶ Extend;
 - ▶ Recupero della flag generata per l'istanza;
 - ▶ Eventuale cancellazione quando la challenge viene risolta;
- Opzionalmente gestire l'upload di `sharedChallenge` e `IsolatedChallenge` su Kubernetes;

`pwnlentoni/prism-ctf-ctfd` supporta tutte queste funzionalità!

Consigli generali

- Automatizzare il setup dell'infrastruttura il più possibile, anche usando strumenti di *Infrastructure as Code* (IaC):
 - ▶ Fare le cose manualmente è molto più error-prone;
 - ▶ Terraform, Pulumi;
- Automatizzare anche il setup di Kubernetes:
 - ▶ k3s, Talos;
- Usare sistemi di *Continuous Delivery* (CD) e *GitOps*:
 - ▶ FluxCD, ArgoCD;
- **Monitorare** il cluster:
 - ▶ kube-prometheus-stack è un buon punto di partenza;

Cose da NON Fare

- **Montare service account nelle challenge:** non dovrebbe essere un problema se RBAC è configurato correttamente, ma meglio evitare a prescindere;
- **Lasciare esposti metadati alle challenge:** a seconda della configurazione del cluster possono esserci diverse conseguenze:
 - ▶ **Leak di (parti di) configurazione:** credenziali per registry Docker, altro?
 - ▶ **Impersonare nodi del cluster:** attacchi tipo **Kubeletmein:**
 - <https://www.4armed.com/blog/kubeletmein-kubelet-hacking-tool/>
 -

Cose da NON Fare

- **Montare service account nelle challenge:** non dovrebbe essere un problema se RBAC è configurato correttamente, ma meglio evitare a prescindere;
- **Lasciare esposti metadati alle challenge:** a seconda della configurazione del cluster possono esserci diverse conseguenze:
 - ▶ **Leak di (parti di) configurazione:** credenziali per registry Docker, altro?
 - ▶ **Impersonare nodi del cluster:** attacchi tipo **Kubeletmein:**
 - <https://www.4armed.com/blog/kubeletmein-kubelet-hacking-tool/>
 - The DefCamp 2024 Quals incident

Ops 🙌 🙌

```
$ kubectl --client-certificate node2.crt --client-key k8shack.key --certificate-authority apiserver.crt --server https://${KUBERNETES_PORT_443_TCP_ADDR} get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE          NOMINATED NODE   READINESS GATES
c-d144-c20242u-l2081-you-can-trust-me-8595b67954-jnz4r  2/2   Running   0           7d3h   10.56.1.3    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d146-c7483u-l2016-rtfm-8475758577-k69gl             2/2   Running   0           7d3h   10.56.1.5    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d155-c6711t-l2111-admin-start-b4c97f965-kjqbg       2/2   Running   0           7d3h   10.56.1.4    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d166-c18879u-l2197-spanzuratoarea-57f8676c8d-dqb6c  2/2   Running   0           7d3h   10.56.1.6    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d182-c7125t-l2463-ftp-console-7999fb8f8c-w9jd4     2/2   Running   0           61m    10.56.176.38 gke-challs-cpu10-20-19f5d7f4-7qm8  <none>           <none>
c-d182-c7126t-l2468-sherlockholmes-76bd588794-gf2qm  2/2   Running   0           56m    10.56.194.28 gke-challs-cpu10-20-19f5d7f4-dh1b  <none>           <none>
c-d182-c7130t-l2434-notes-6b98965677-c2wns          2/2   Running   4 (21m ago) 24m    10.56.202.17 gke-challs-cpu10-20-19f5d7f4-hx6w  <none>           <none>
c-d182-c7130t-l2466-pyterm-db598d5bb-mkxt8          2/2   Running   0           29m    10.56.1.37    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d182-c7133t-l2460-ctr-864989c46d-snq6s           2/2   Running   0           5m38s   10.56.195.36 gke-challs-cpu10-20-19f5d7f4-xlr6  <none>           <none>
c-d182-c7134t-l2434-notes-856b996685-2985t         2/2   Running   0           61m    10.56.194.26 gke-challs-cpu10-20-19f5d7f4-dh1b  <none>           <none>
c-d182-c7134t-l2456-reelfreaks-54bd7c4959-v25q2     2/2   Running   0           49m    10.56.212.9   gke-challs-cpu10-20-19f5d7f4-r4r6  <none>           <none>
c-d182-c7134t-l2463-ftp-console-57446b58f8-pbdgp     2/2   Running   0           100m   10.56.178.58 gke-challs-cpu10-20-19f5d7f4-tln5  <none>           <none>
c-d182-c7137t-l2432-mobisec-7674f795b6-gkw5f       2/2   Running   0           8m30s   10.56.211.18 gke-challs-cpu10-20-19f5d7f4-6cgp  <none>           <none>
c-d182-c7137t-l2457-noogle-58cdd484c-n2kw8         2/2   Running   0           32m    10.56.214.6   gke-challs-cpu10-20-19f5d7f4-pqms  <none>           <none>
c-d182-c7137t-l2459-production-bay-6c5bd7b6f7-q24pf 3/3   Running   0           11m    10.56.210.24 gke-challs-cpu10-20-19f5d7f4-bk2v  <none>           <none>
c-d182-c7137t-l2461-aptssh-69fc9dd696-lp2vj        2/2   Running   0           41m    10.56.1.31    gke-challs-cpu4-ram8-352df4d3-ysi3  <none>           <none>
c-d182-c7140t-l2432-mobisec-7df4645678-5945q       2/2   Running   0           32m    10.56.214.7   gke-challs-cpu10-20-19f5d7f4-pqms  <none>           <none>
c-d182-c7140t-l2456-reelfreaks-8c4b9f5ff-z27tt     2/2   Running   0           47m    10.56.179.50 gke-challs-cpu10-20-19f5d7f4-688h  <none>           <none>
c-d182-c7140t-l2457-noogle-69bbfcdcd5-k9j68        2/2   Running   0           33m    10.56.214.4   gke-challs-cpu10-20-19f5d7f4-pqms  <none>           <none>
c-d182-c7140t-l2459-production-bay-568768bdbc-bqt7c 3/3   Running   0           30m    10.56.178.76 gke-challs-cpu10-20-19f5d7f4-tln5  <none>           <none>
c-d182-c7140t-l2460-ctr-6c4468487b-98qpx           2/2   Running   0           57m    10.56.204.14 gke-challs-cpu10-20-19f5d7f4-4rvq  <none>           <none>
c-d182-c7146t-l2435-login-6dfd586d6-x9t4b          2/2   Running   0           2m24s   10.56.178.85 gke-challs-cpu10-20-19f5d7f4-tln5  <none>           <none>
c-d182-c7146t-l2461-aptssh-77fd765668-swbzz         2/2   Running   0           61m    10.56.179.47 gke-challs-cpu10-20-19f5d7f4-688h  <none>           <none>
c-d182-c7148t-l2432-mobisec-6875c58fcd-8crzd       2/2   Running   0           79m    10.56.210.8   gke-challs-cpu10-20-19f5d7f4-bk2v  <none>           <none>
c-d182-c7155t-l2460-ctr-65ccbfcdd8-vgvjg           2/2   Running   0           2m18s   10.56.176.46 gke-challs-cpu10-20-19f5d7f4-7qm8  <none>           <none>
c-d182-c7157t-l2457-noogle-5f49b8c679-kp96w        2/2   Running   0           47m    10.56.176.45 gke-challs-cpu10-20-19f5d7f4-7qm8  <none>           <none>
c-d182-c7157t-l2460-ctr-7448c84b44-g7qs5          2/2   Running   0           28m    10.56.188.46 gke-challs-cpu10-20-19f5d7f4-gsxj  <none>           <none>
c-d182-c7157t-l2468-sherlockholmes-dbd7c8c78-xghh5  2/2   Running   0           11m    10.56.196.33 gke-challs-cpu10-20-19f5d7f4-zjnc  <none>           <none>
c-d182-c7162t-l2435-login-6cb4db8b94-4wxvh         2/2   Running   0           2m17s   10.56.203.16 gke-challs-cpu10-20-19f5d7f4-vnjb  <none>           <none>
c-d182-c7170t-l2457-noogle-78b7c69d79-l2kgn        2/2   Running   0           18m    10.56.202.22 gke-challs-cpu10-20-19f5d7f4-hx6w  <none>           <none>
c-d182-c7170t-l2468-sherlockholmes-74fc94f858-msjtx 2/2   Running   0           9m24s   10.56.202.24 gke-challs-cpu10-20-19f5d7f4-hx6w  <none>           <none>
c-d182-c7178t-l2458-onacle-srL-54c899d946-jwxn9     3/3   Running   0           2m38s   10.56.189.57 gke-challs-cpu10-20-19f5d7f4-2j7b  <none>           <none>
c-d182-c7179t-l2435-login-78685b4f4-vczd9         2/2   Running   0           14m    10.56.179.58 gke-challs-cpu10-20-19f5d7f4-688h  <none>           <none>
c-d182-c7179t-l2456-reelfreaks-6cb97bc47-xl9x5     2/2   Running   1 (26m ago) 62m    10.56.203.13 gke-challs-cpu10-20-19f5d7f4-vnjb  <none>           <none>
c-d182-c7187t-l2454-siem-logs-565d754756-62bf7     3/3   Running   0           40m    10.56.213.9   gke-challs-cpu10-20-19f5d7f4-s64b  <none>           <none>
c-d182-c7187t-l2460-ctr-774586b646-rmf18          2/2   Running   0           2m18s   10.56.188.48 gke-challs-cpu10-20-19f5d7f4-gsxj  <none>           <none>
c-d182-c7187t-l2466-pyterm-6ddb679f78-jz2pf        2/2   Running   0           16m    10.56.213.12 gke-challs-cpu10-20-19f5d7f4-s64b  <none>           <none>
c-d182-c7189t-l2456-reelfreaks-5696c575f9-fqzpj    2/2   Running   0           5m45s   10.56.198.35 gke-challs-cpu10-20-19f5d7f4-q9sv  <none>           <none>
```

Idee per il futuro

- **Centralizzare log** delle challenge in modo da poterli consultare agevolmente da Grafana. Setup basato su Grafana Loki e Alloy;
- **Catturare traffico verso le challenge** per consultarlo in caso di dubbi su soluzioni o investigare DoS. Usare PCAP-over-IP e Arkime?
- **Node Autoscaler** per aumentare e diminuire automaticamente il numero di server in base alle richieste;
- **Storage distribuito** con Ceph Rook per le challenge che lo richiedono
nessuna?

Idee per il futuro - 2

- **Automatizzare setup CTFd**: caricare tutte le challenge correttamente è abbastanza *error-prone*:
 - ▶ Abbiamo fatto un typo nella flag di una challenge 😅
- Plugin che converte **flag template socaz** → **regex**
-

Idee per il futuro - 2

- **Automatizzare setup CTFd**: caricare tutte le challenge correttamente è abbastanza *error-prone*:
 - ▶ Abbiamo fatto un typo nella flag di una challenge 😅
- Plugin che converte **flag template socaz** → **regex**
- Piattaforma CTF custom? 🙄

Ringraziamenti

Cybersecurity National Lab per gli ottimi tool resi disponibili:

- socaz
- challenge-jail
- challenge-headless

Ringraziamenti

Cybersecurity National Lab per gli ottimi tool resi disponibili:

- socaz
- challenge-jail
- challenge-headless

Avete mai pensato di renderli open source? 🙄

Ringraziamenti

Il **Comitato organizzatore** del Workshop per avermi dato la possibilità di essere qui oggi

Ringraziamenti

Tutti gli amici **pwnlenti** per la scrittura delle challenge ❤️

Ringraziamenti

Grazie anche al pubblico per l'attenzione!

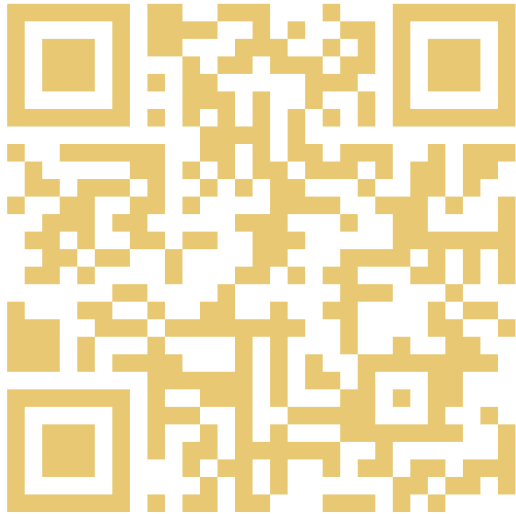
Click to exit presentation...

Click to exit presentation...

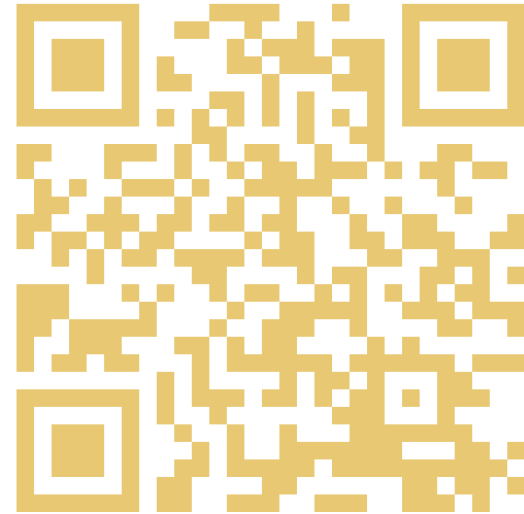
Or maybe not 🤔

Open source!

`prism-ctf` e `prism-ctf-ctfd` sono da ora pubblicamente disponibili su GitHub!



↑ `prism-ctf`



↑ `prism-ctf-ctfd`