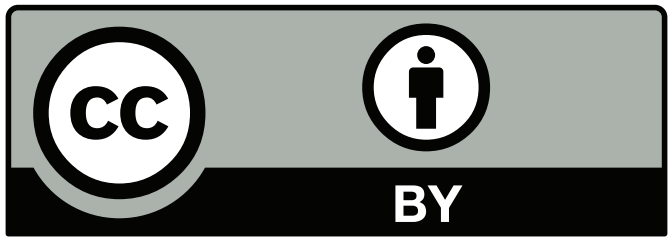
Introduzione alle CTF

Corso in Ingegneria del Software

Alessandro Righi Cristiano Di Bari

Università degli Studi di Verona

9 Novembre 2023



Chi siamo?

- laurea magistrale @ UniVR
- istruttore @ CyberChallenge
- *System Developer* @ IOTINGA

Cristiano Di Bari

Alessandro Righi

- laurea magistrale @ UniVR
- istruttore @ CyberChallenge
- *Applied Research Director* @ IOTINGA

Cosa sono le CTF?

Le *Capture The Flag* (CTF) sono delle sfide in cui i partecipanti devono trovare delle *flag*, ossia delle stringhe di testo, all'interno di sistemi informatici contenenti delle vulnerabilità di sicurezza.

Esempio di flag

`CCIT{Th1s-1sY0uR-F1rst-Fl4ag}`

Una volta trovate le flag vanno, solitamente, inviate ad una piattaforma di gara, che si occupa di validarle, assegnando il punteggio della challenge nel caso siano corrette.

Perché fare CTF?

Avete mai fatto CTF?

Se no... ecco alcune ragioni per iniziare:

- per divertirsi!
- per imparare cose nuove (tante)
- per scrivere software (più) sicuro
- per conoscere nuova gente, creare networking

Tipologie di CTF

Esistono due tipologie di CTF:

- *Jeopardy*: il partecipante attacca una serie di servizi malevoli e sottomette le flag ad un sistema di verifica
- *Attack-Defence (AD)*: competizione a squadre dove ogni team deve attaccare i sistemi dell'avversario, e difendere i propri

Per queste lezioni di concentreremo sul primo tipo (*Jeopardy*), di gran lunga le più diffuse, che è anche quella organizzata da *Würth Phoenix*.

Tipologie di challenge

Tipicamente le challenge che si affrontano possono essere di 4 macro categorie:

- *Binary*: è necessario ricercare vulnerabilità in un eseguibile, quali ad es. buffer overflow
- *Web*: si tratta di trovare vulnerabilità in una web app, web API, o comunque applicativo esposto in rete
- *Crypto*: è necessario decodificare un testo cifrato con un algoritmo (ovviamente vulnerabile)
- *Misc*: sono challenge che non rientrano in nessuno dei tipi precedenti, e richiedono spesso creatività per essere affrontate

Per queste lezioni ci concentreremo sulla categoria *Web*.

Iniziamo!

Path traversal

Immaginiamo di avere una pagina web che per caricare l'immagine del profilo di un utente effettua una richiesta a:

Richiesta

```
https://mysecureapp.com/assets?name=image.jpeg
```

Cosa succede se modifico la richiesta in questo modo?

Richiesta alterata

```
https://mysecureapp.com/assets?name=../index.php
```

Se il server non effettua adeguati controlli, è possibile leggere file fuori dalla *root* directory del web server!

Path traversal

Cosa consente di fare questa vulnerabilità?

- leggere *segreti* altrimenti non accessibili, ad es. file di configurazione quali `/etc/passwd`
- ottenere il *codice sorgente* dell'applicazione web
- accedere ai dati di altri utenti, bypassando restrizioni imposte dall'applicazione web

Suggerimento

È possibile aggiungere tanti `../` fino a raggiungere la directory *root*, ad esempio `../../../../../etc/passwd`

Challenges

Vediamo la prima challenge. Per queste lezioni utilizzeremo delle challenge prese dalla piattaforma di allenamento delle Olimpiadi di Cybersecurity (`https://olicyber.it`), a cui vi invitiamo ad iscrivevi.

Traverse me

`https://training.olicyber.it/challenges#challenge-504`

Ho trovato questa galleria di quadri, chissà se ce ne sono di nascosti.

- In questa applicazione vengono caricati dei file salvati sul server, sai individuare dove?
- Forse è possibile modificare il nome del file per visualizzare altri elementi presenti sul filesystem.

Domanda

É possibile stampare altri file?

Traverse me more

`https://training.olicyber.it/challenges#challenge-507`

Questa volta ho fixato il problema della galleria d'arte, non riuscirai mai a carpire il mio segreto.

- Provando ad applicare la soluzione per la challenge precedente ci accorgiamo che non funziona!
- Guardando il codice sorgente, notiamo che viene fatto un *sanitize* del nome del file che rimuove alcuni caratteri.
- Possiamo bypassare questo controllo?

Light or dark

`https://training.olicyber.it/challenges#challenge-49`

- La challenge mostra un sito web in cui è possibile selezionare il tema, come viene applicato lo stile alla pagina html?
- Scaricando il file sorgente php notiamo che lo sviluppatore ha applicato dei filtri per evitare che un utente malintenzionato possa caricare un file diverso dai fogli css.
- Il secondo controllo verifica che l'estensione del file da caricare sia `.css`, in caso contrario aggiunge l'estensione richiesta tramite una concatenazione di stringhe. Questo controllo sembra difficile da aggirare...o forse no.

Suggerimento

Avete mai sentito parlare di "Null Byte Injection"?

Burp Suite

Introduciamo un tool che ci sarà utile per le prossime challenge.

- Il tool *Burp Suite* (scaricabile in versione community gratuitamente) agisce come proxy HTTP
- Agendo da proxy blocca le richieste HTTP fra il browser ed il server, e consente di ispezionarle, eventualmetne modificarle e quindi farne il forward al server.
- Include anche una serie di utilità, ad esempio la possibilità di decodificare valori in vari formati, ad esempio base64.

Vediamone una demo!

Flags shop

`https://training.olicyber.it/challenges#challenge-43`

- La challenge mostra un sito web dal quale è possibile acquistare dei prodotti
- Forse guardare le richieste che effettua l'applicazione web aiuta?

Domanda

Quale tool visto prima potrebbe agevolarci nello svolgimento di questa challenge?

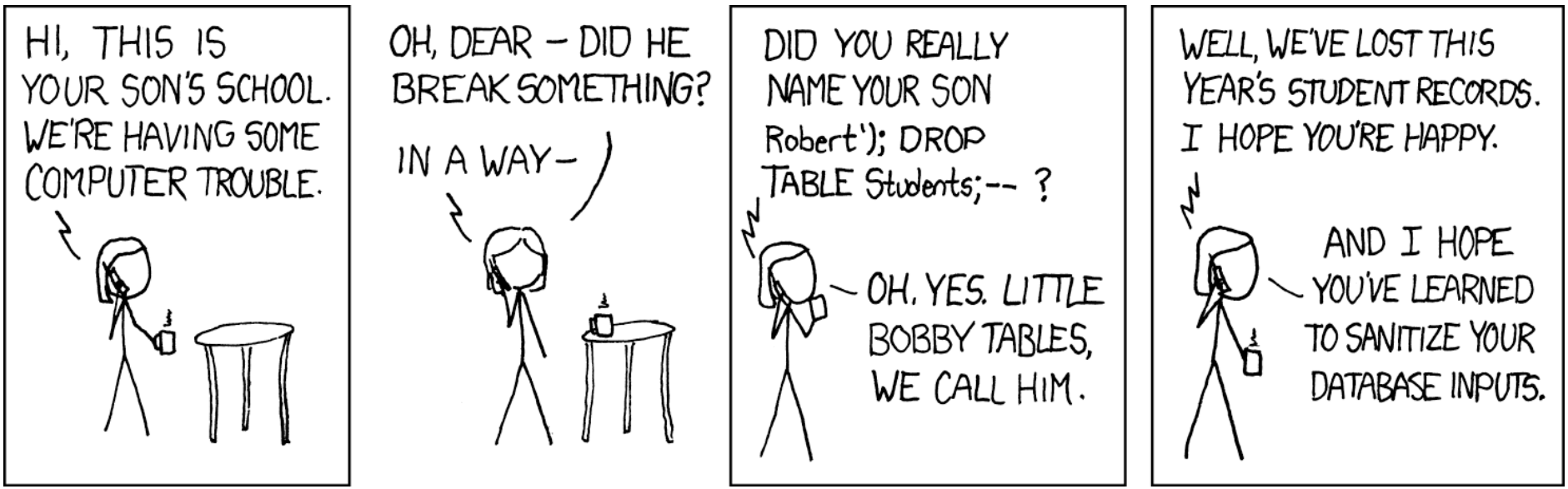
SQL Injection

```php
<?php
$user = mysql_query("SELECT * FROM users WHERE
    username = '$username' AND password = '$password');
if (!$user) {
    echo "utente non trovato!";
} else {
    echo "Accesso eseguito!";
}
```

Domanda

Il codice in questa slide è sicuro?

SQL Injection



https://xkcd.com/327/

SQL Injection

Vediamo cosa accade andando a settare la variabile username al valore `"admin'--"`:

```
SELECT * FROM users WHERE
username = 'admin'--' AND password = 'boh'
```

Il controllo della password è completamente stato eluso!

Prevenzione

Mai compore query SQL interpolando stringhe!

SQL Injection

- le SQL injection rientrano nella più ampia categoria delle *code injection*
- le SQL injetion sono *purtroppo* ancora una delle vulnerabilità più diffuse
- in questo caso, si trattava di una *blind* SQL injection, in quanto il risultato non è direttamente riflesso all'utente
- usa sempre i *prepared statement* messi a disposizione della libreria che usi per interrogare il database per effettuare query!

Basic SQLi

`https://training.olicyber.it/challenges#challenge-48`

- il titolo dice tutto!
- ricorda la slide precedente...
- SQL supporta altri caratteri per aggiungere commenti

Admin's secret

`https://training.olicyber.it/challenges#challenge-44`

- analizza attentamente il codice sorgente
- nel login vengono usati i prepared statement, forse il problema è nella registrazione
- riesci a creare un utente di tipo admin?

Password changer 3000

`https://training.olicyber.it/challenges#challenge-59`

Riesci a cambiare la password dell'utente "admin"?

- Un sito per cambiare le password degli utenti...non sembra molto utile.
- Provando a cambiare la password dell'utente "pippo" otteniamo in risposta una nuova password. Proviamo ad analizzare la richiesta HTTP che abbiamo appena effettuato.
- Dovremmo riuscire ad impersonare l'utente "admin", forse la codifica `Base64` può esserci d'aiuto.

Curl

- per le prossime challenge, sarà necessario effettuare richieste HTTP a servizi REST.
- esistono svariati tool per farlo, quali *Insomnia*, *Postman*, etc.
- vediamo brevemente il tool da riga di comando *curl*
- molto probabilmente lo hai già installato sul tuo sistema

Per gli utenti Windows

Utilizza il comando `curl.exe` in quanto `curl` usa opzioni diverse.

Curl

- GET: `curl -v http://example.com`
- POST: `curl -v -X POST http://example.com -d "body"`
- set di un header: `curl -v -H 'chiave:  valore' http://example.com`

Suggerimento

Dagli strumenti di sviluppo di Firefox e Chrome nella pagina *Rete* è possibile copire il comando curl di una richiesta nel menù che appare cliccando con il tasto destro così da poterlo incollare nel terminale

A TOO small reminder...

`https://training.olicyber.it/challenges#challenge-36`

- assicurati di specificare il *Content-Type* corretto!
- non sembrano esserci vulnerabilità nel login
- guarda attentamente il cookie di sessione che viene assegnato, noti nulla?
- possiamo presumere che l'admin abbia una sessione aperta

Prevenzione

I cookie di sessione dovrebbero essere impossibili da indovinare

ZioFrank

`https://training.olicyber.it/challenges#challenge-53`

- la vulnerabilità non è un SQL injection
- controlla *attentamente* lo schema dei dati
- sembra che venga effettuata una query separata per capire se l'utente è admin
- qualcosa previene il creare un nuovo utente con lo stesso username di uno esistente?

Suggerimento

È sempre buona norma definire le colonne univoche con `UNIQUE`

NGROK

- *NGROK* è uno strumento che permette di esporre una applicazione in ascolto su `localhost` ad internet
- viene generalmente usato per il testing di applicazioni web o API
- per prima cosa tiriamo su un server http: `php -S 127.0.0.1:5000`
- attiviamo il tunnel con ngrok: `ngrok http 5000`

Il tool è gratuitamente scaricabile da `https://ngrok.com/download`.

Cross-Site Scripting

```
document.innerHTML = "Benvenuto, " + username
```

Domanda

Cosa c'è di sbagliato in questo codice apparentemente innocuo?

Cross-Site Scripting

- cosa accade se username contiene degli elementi HTML?
- in particolare, se contiene dei tag `<script>`?
- è possibile far eseguire del codice JavaScript a chiunque apra quella pagina!
- ad esempio, possiamo leggere il cookie di sessione e farlo inviare ad un nostro server
- oppure fare richieste alla pagina stessa con i cookie dell'utente

Prevenzione

Usa sempre un framework o una libreria che si occupa di sanificare il testo che inserisci nelle pagine HTML dinamiche!

Curious George

`https://training.olicyber.it/challenges#challenge-40`

- Dopo aver fatto il login accediamo alla pagina riservata del nostro utente. Se ricarichiamo la pagina come fa il sito a capire la nostra identità?
- Il sito sembra navigabile specificando nel URL il parametro GET id, purtroppo questa funzione é riservata all'amministratore.
- La funzionalità principale della piattaforma consiste nel modificare la descrizione del proprio utente. Chissá chi la leggerà mai...
- Esiste anche una pagina per segnalare bug all'admin della piattaforma, possiamo sfruttarla a nostro vantaggio?

Grazie per l'attenzione

Ci auguriamo che queste due lezioni ti abbiano lasciato più che altro curiosità di approfondire gli aspetti di sicurezza!

Se sì... hai mai sentito parlare della *CyberChallenge*?

Ringraziamenti

CyberChallenge.IT

IOTINGA SRL