# CTF2


**Autore:** *Daniele Pellegrini*

**Matricola**: *162779*


Nel seguente report si illustrano le vulnerabilità che hanno permesso di raggiungere le flag di utente e di amministratore di sistema nell'esercitazione CTF2.

- FLAG UTENTE: `3666011249f276aee20047d2082c758c`
- FLAG ROOT: `9a31aae079ea56c53d872aeeb5db9dec`

Di seguito vengono illustrati i passaggi svolti per sfruttare le vulnerabilità di sistema (*red team*) e delle possibili soluzioni per la difesa (*blue team*).

## Scansione della Rete

RED TEAM: Con il comando `nmap`, si ricerca un dispositivo *nella nostra stessa rete* che abbia una porta aperta e che magari ospiti un server web. Una prima scansione non rileva nulla, si procede pertanto a a effettuare uno *stealth scan* specificando al tool di provare a guardare su tutte le porte mediante il comando `-p-`. Ne trova uno sulla porta *50987*, e inoltre trova la porta aperta *22* in ascolto per una possibile connessione in SSH.

BLUE TEAM: Una prima soluzione per la difesa sarebbe quella di rimuovere il server web da una rete sulla quale non si ha sempre piena conoscenza degli host che la popolano. La stessa cosa vale per la connessione SSH: mantenerla solo se si è connessi a *reti affidabili*.

## Server Web e Virtual Host

RED TEAM: Il server web sulla porta 50987 è una pagina *in costruzione* riguardante il personaggio di Super Mario. Poichè il sito web dopo una prima analisi non presenta vulnerabilità si fa attenzione sul solo indizio che si ha: il testo. Nello slider viene scritto chiaramente che il sito è in costruzione e che al momento gli sviluppatori stanno lavorando sulla nuova *wiki*. Si procede pertanto a creare una *wordlist* con le parole *mario* e *wiki* e dopo qualche tentativo con l'ausilio di `Burp` si arriva al dominio `wiki.mario.it`, hostato sulla stessa portate mediante il *virtual hosting*. La pagina, anch'essa in costruzione, presenta un form di ricerca con metodo POST.

BLUE TEAM: per evitare situazioni di questo tipo sarebbe opportuno non disseminare informazioni tanto rilevanti sulla pagina principale del sito, in modo tale da non permettere a chiunque di poterci facilmente arrivare deducendo il nome dell'host da inserire semplicemente modificando una richiesta.

## SQL Injection e Shell MySQL

RED TEAM: Essendo la form trovata adatta ad una SQL Injection, è possibile sfruttare il tool `sqlMap` per carpire quante più informazioni utili. Utilizziamo il tool con l'opzione `--file-read` per poter leggere i documenti presenti sul sistema. Già da questo momento sarebbe possibile leggere la *flag utente*, facendo un tentativo nel path: */home/mario/user.txt*. Con l'opzione specificata è possibile anche andare a leggere la configurazione del database MySQL, situata di default in: */etc/mysql/my.cnf*. In questo file, oltre alle normali informazioni sul database, è presente un commento dove vengono specificate delle credenziali di accesso in SSH come utente MySQL: `mysql:Ah,Princess-a,whatabeautifulevening`. Accedendo in SSH con le credenziali ottenute è possibile navigare nella directory principale dell'utente Mario, nella quale è presente una chiave privata per l'accesso in SSH senza richiesta di password.

BLUE TEAM: Il controllo sull'input della form è il primo requisito per evitare problemi dovuti a SQL Injection. Questo potrebbe essere fatto in diversi modi, come ad esempio assicurandosi che l'input ottenuto sia corrispondente a una determinata parola, oppure effettuando l'escape di determinati caratteri per evitare all'attaccante di effettuare dei comandi per carpire informazioni sul database e sui file presenti nel sistema. Limitare i permessi di lettura all'utente MySQL (utilizzato quando si fanno le query) è un altro requisito per rendere il sistema più sicuro, oltre ad evitare di lasciare pubbliche delle credenziali per l'accesso in SSH all'interno di file di configurazione del database situati in percorsi dove possono essere facilmente raggiungibili da chiunque.

## Accesso SSH - User Mario

RED TEAM: Copiando la chiave privata dell'utente Mario è possibile effettuare l'accesso in SSH sulla macchina vittima senza bisogno della password utente.

BLUE TEAM: Se si vuole tenere un file contenente delle credenziali o una chiave privata a portata di mano, si potrebbe quanto meno proteggere il file con una password complessa, o modificarne i permessi di lettura di modo che sia necessario essere *amministratore di sistema*. In alternativa, si potrebbe salvare il file su un *dispositivo secondario* diverso da quello che altrimenti si rende vulnerabile.

## Privilege Escalation

RED TEAM: Dopo una prima ricerca delle vulnerabilità del sistema, è possibile trovare un file sospetto nel percorso: */opt/find_princess.sh*. Provando a lanciare questo file, ci si rende conto che l'output corrisponde a un elenco dei file presenti nella directory in cui viene lanciato. Lanciando il file con l'opzione *--help* ci si rende conto che ha le stesse funzionalità del comando `find` presente sui sistemi Unix, e che pertanto presenta le stesse opzioni. Con il comando `ls -la` si possono inoltre vedere i permessi associati al file, dal quale si nota che il file appartiene all'utente root è che presenta un tipo permesso speciale: `suid`. Questo permesso ci consente di eseguire lo script come se ad eseguirlo fosse l'utente a cui appartiene il file, ovvero *root*. Pertanto è possibile lanciare il comando specificando come path di ricerca `/` corrispondente alla radice del sistema. Analizzando l'output è possibile notare che è presente un file in */root/root.txt* che fa a caso nostro: si tratta della flag di root. Eseguendo lo script in vesti di amministratore di sistema, è possibile leggere il file stampandone il contenuto a schermo utilizzando il comando `exec` come segue:

```
./opt/find_princess.sh /root -type f -iname "root.txt" -exec cat {} \;
```

BLUE TEAM: Settare il SUID su un certo tipo di file, come su quello corrispondente al comando *find* appena visto, talvolta può risultare dannoso, specie quando si è root, in quanto si da agli altri utente i *permessi di amministratore di sistema*. Una buona soluzione a questo problema potrebbe essere la semplice rimozione del SUID dal file.