L'importanza della sicurezza informatica

*Appunti di **Giuseppe Pitruzzella** - Corso di Internet Security @ DMI, UniCt*

La sicurezza informatica è l'informatica. Non esiste un campo dell'informatica dove non è presente la sicurezza.
Per questo motivo, creare un servizio implica garantire la sicurezza per lo stesso. Un esempio è possibile a partire da tutti gli applicativi che installiamo giornalmente, sui cui poniamo notevole fiducia, una fiducia necessaria a credere che nessuno di essi sia un applicativo malevolo, come per esempio, un trojan.

Trojan

Un esempio di trojan (in particolare una certa famiglia di Trojan) è descritto dal codice seguente e nasce a partire dalla ricezione di un file malevolo (*matrice socio-tecnologica*) che ipotizziamo sia uno script chiamato `ls.sh` (che eseguirà le istruzioni al dì sotto e viene eseguito dalla vittima stessa).

Sappiamo che ogni file ha un proprietario ed un gruppo e, normalmente, ogni programma viene eseguito secondo i permessi che possiede l'utente che lo lancia (il comando `whoami` ritorna, come intuibile, chi siamo).

Attivare il permesso `setuid` (il bit `s`) indica al SO di *eseguire il programma con i permessi del proprietario*, in aggiunta ai permessi dell'utente che lancia quest'ultimo. Intuiamo, quindi, che assegnare il bit `s` ad un file che è posseduto da `root` implica il programma sarà aperto con i permessi da `root`.

```
cp /bin/sh /tmp/.xxsh
chmod u+s,o+x /tmp/.xxsh
rm ./ls.sh
ls
```

Del codice soprastante è necessario notare le seguenti nozioni:

- Il primo comando copia la shell `/bin/sh` in una cartella temporanea, ossia `tmp`, notoriamente utilizzata dal sistema per funzioni analoghe (per es. buffer); Il motivo per cui quest'ultima viene copiata all'interno di questa cartella è un euristica *socio-tecnologica*. Inoltre il file copiato è preceduto da un punto, il che indica che sarà visibile solo se il comando ls è accompagnato dal flag `-a` (questa è un euristica socio-tecnologica). Si noti, inoltre, che i permessi del file copiato dipendono dalla distro utilizzata.
- Il secondo comando cambia i permessi in modo conveniente. Infatti, il file, poiché copiato dall'utente, tendenzialmente avrà i suoi stessi permessi. Potrebbe anche accadere che il file copiato abbia gli stessi permessi del proprietario della cartella ricevente, quindi tmp, ossia root. Il tutto dipende dalla policy per il comando cp. Quindi, a partire da una policy, o meglio una scelta progettuale, permetto o meno una famiglia di Trojan.
- Il terzo comando elimina il file ls (eseguibile) all'interno della cartella corrente. Si presuppone, infatti, che vi siano due ls, il vero ls di sistema ed un falso ls, ovvero `ls.sh`.
- Il quarto comando richiama il vero `ls` di sistema, il che potrebbe far intendere alla vittima che ciò che ha eseguito è stato un semplice `ls`.

[*NB*] Ogni comando è eseguito dall'utente vittima, che esegue a sua volta il file `ls.sh` pensando di eseguire il comanda di sistema `ls`.

Il risultato è una shell nel sistema con i permessi di root. In altre parole, l'attaccante è riuscito ad essere root nel sistema della vittima bypassando l'autenticazione.
Si noti esista la possibilità di creare delle varianti a partire da questo attacco (Trojan). Una di queste

potrebbe essere il semplice bypass dell'autenticazione, attraverso cui si potrebbe creare un nuovo utente (l'attaccante) root, oppure ancora inviare i dati della vittima ad un suo server a partire da un client ftp.
Il Trojan adesso è concettualmente obsoleto poichè (i) il trojan scaricato da parte della vittima non sarebbe di per sè eseguibile e (ii) nessuno adesso eseguirebbe un file con "`./nome`", ciò nonostante alcuni SO hanno fixato il problema solo pochi anni fa.
Il motivo per cui un tempo il Trojan aveva senso è fortemente legato alla cultura del tempo.

> [$Approfondimento$] La **Open Source INTelligence**, acronimo **OSINT** (in italiano: "Intelligence su fonti aperte"), è quella disciplina dell'intelligence che si occupa della ricerca, raccolta ed analisi di dati e di notizie d'interesse pubblico tratte da fonti aperte. Può essere intesa come la prima fase di un attacco.

> [$Esame$] Cos'è un trojan? Un esempio di trojan?

> [$Esame$] Spiegare l'attacco all'interno di questo primo trojan.

Password

Il concetto di **password** è anch'esso estremamente importante all'interno della sicurezza informatica e con esso anche il concetto di "*scelta della password*", il quale deve essere difficile da indovinare ed allo stesso tempo facile da ricordare.
Sappiamo bene, infatti, che scegliere una parola come "ciao" non è desiderabile per la sicurezza del nostro account poichè potrebbe essere facilmente indovinata da un **attacco dizionario**, un attacco che prevede di provare tutte le password possibili a partire da un certo dizionario di parole.
Oggigiorno sappiamo che ciò non accade spesso, ed il motivo riguarda le regole imposte da un sito web rispetto la scelta della password, la quale *deve* rispettare certi **criteri**. Purtroppo queste regole non esistono da sempre, infatti sono state introdotte dal **NIST** solo nel 2004, anno in cui vengono pubblicate le regole per la robostezza di una password.

> [$Approfondimento$] Il **NIST**, a partire dal 2004, afferma che una password è robusta non è stata già diffuse a seguito di altri attacchi, non sia una parola del dizionario, non vi siano dei caratteri sequenziali o ripetuti (come "1234" e "aaaa"), non siano delle parole derivate dal nome del servizio (per esempio, Facebook dovrebbe impedire agli utenti di usare password come "Facebook" o "Facebook01" o "Facebook-Giuseppe") ed, infine, che non siano parole che contengono il nome, il cognome o l'user-id dell'utente.

Le password però possono essere facilmente rotte se il **tempo** lo permette, motivo per cui il *mist* (l'ente che ha disposto i criteri rispetto una password) afferma che ogni password deve essere cambiata ogni 6 mesi. A questo punto l'unico problema potrebbe essere relativo ad cambiamento futile rispetto la nuova password per l'utente.
Un altro problema rispetto le password è il **riutilizzo** di quest'ultima.

> [$Approfondimento$] **Heartbleed** è un bug di sicurezza nella libreria di OpenSSL, un'implementazione open-source ampiamente usata del protocollo TLS. Heartbleed potrebbe essere sfruttato indipendentemente dal fatto che l'istanza OpenSSL stia girando come server o client TLS. È il risultato di una validazione input impropria (data dalla mancanza di controllo dei limiti nell'implementazione dell'estensione *heartbeat* del protocollo TLS.

Cosa si intende con "*usabilità*"? Il professore spiega questo secondo una vecchia piattaforma che permetteva di invare un messaggio ad un numero telefonico senza alcuna autenticazione.