15. CHECK-LIST DI SICUREZZA

In questo capitolo ti verranno indicate le check-list di sicurezza consigliate per effettuare una buona configurazione di sicurezza sulla maggior parte delle macchine che verranno messe in rete e che ospiteranno web servers.

Ti consigliamo di effettuare una copia di ognuna di esse e di spuntarle quando ne avrai bisogno: per ogni voce, spunta con una X il ❒ contrassegnato per ogni riga e, se necessario, specifica eventuali note nella riga a destra.

Tieni presente che questo è solo un modello base, non un format standard e specifico: con il tempo, potresti voler creare la tua check-list in base a ciò che devi fare e a quello che devi configurare.

	Analisi	Note
	Web Server	
☐	Il Sistema Operativo è aggiornato	
☐	Il Web Server è aggiornato	
☐	Gli interpreti (PHP, Perl, Python etc...) sono aggiornati	
☐	I Framework sono aggiornati	
☐	Le estensioni sono aggiornate	
☐	Il server non consente il banner grabbing	
☐	Il server non permette il directory listing su uno o più path critici	
☐	L'infrastruttura non permette di risolvere l'IP della macchina ma solo del reverse proxy	
☐	Tutti i servizi non necessari sono disabilitati	
☐	Tutti gli utenti e i gruppi non necessari sono stati rimossi	
☐	Il server logga correttamente tutte le richieste e risolve tutti gli indirizzi IP	
☐	Le estensioni del web server non utilizzate sono state disattivate	
☐	I contenuti demo dei web server che potrebbero permettere la profilazione della versione del Web Server sono disabilitati	
☐	Le pagine di stato d'errore (404) sono personalizzate per evitare la determinazione del web server	
☐	L'utente HTTP usato dal web server ha permessi limitati al solo funzionamento della web app	
☐	I moduli di sicurezza del Web Server (ModSecurity etc...) sono stati correttamente configurati e attivi	
☐	Il Server è stato testato con un WASS e non rileva vulnerabilità visibili	
☐	Se è stato implementato un IDS, è stato verificato manualmente	

	Database
☐	Il DBMS è aggiornato
☐	Il software viene eseguito da un utente con privilegi limitati
☐	Utenti e database demo sono stati rimossi
☐	Gli account non hanno alcuna password di default

	Web App
☐	Se un CMS è stato aggiornato, così come plugin e temi
☐	Le password sono complesse e il web form limita i login errati
☐	Il codice sorgente (in HTML) non contiene informazioni che spiegano l'infrastruttura web
☐	I file di test vengono rimossi prima della produzione
☐	Il web login ha un sistema anti-bot efficace (come il CAPTCHA) e richiede la validazione email
☐	L'applicazione è stata testata con un WASS e, opzionalmente, una sessione di pentesting

16. HACKING CRIBSHEET

Carattere	Descrizione
	Programmazione
' "	Apice o doppio-apice, definiscono le stringhe in PHP, Javascript e SQL. Indicano anche i valori in HTML.
;	Separatore di comando, usato in PHP, Javascript, CSS e SQL
<	Apre un tag in HTML
>	Chiude un tag in HTML
?	Determina i valori in una query-string (metodo GET)
=	Si pone solitamente tra variabile e valore
<?php	Apre un tag PHP
?>	Chiude un tag PHP
<script>	Apre un tag clientscript (solitamente Javascript)
</script>	Chiude un tag clientscript (solitamente Javascript)
+	Separa valori in Javascript e nella query-string
.	Permette di accedere a cartelle e sottocartelle (in combinazione con /)
/	Permette di accedere a cartelle e sottocartelle (in combinazione con .)
$	Usato in PHP per indicare una variabile

Carattere	Descrizione
	SQL Injection
'	Apostrofo, usato per verificare la presenza di vulnerabilità SQLi
--	Commento su linea singola, permette di ignorare eventuali caratteri successivi in SQL
%	Wildcard, permette di verificare la presenza di caratteri multipli senza conoscerne il contenuto
OR 1=1	Crea la condizione vera sull'SQL. È la base di un attacco SQLi.
OR '1'='1	Come prima, utilizzato su query che però richiedono un valore stringa.
UNION ALL SELECT	Funzione SQL per andare a prelevare valori da altre tabelle

Porta	Servizio (comune)

21	FTP, servizio usato per il trasferimento di file
22	SSH, servizio usato per la gestione remota di un sistema
23	Telnet, come SSH ma meno sicuro
80	Porta standard della comunicazione HTTP
81	Porta alternativa alla 80
88	Porta alternativa alla 88
443	Porta standard della comunicazione HTTPS
8000	Porta alternativa alla 80, spesso usata come web cache
8001	Porta alternativa alla 80, spesso usata come web server manager
8888	Porta alternativa alla 80

17. CHEATSHEET COMANDI LINUX

Per conoscere i file e le cartelle presenti nella directory in cui ci troviamo:

```
$ ls
```

Per accedere a una cartella (dove {nomecartella} sarà il nome della cartella a cui vogliamo accedere):

```
$ cd {nomecartella}
```

Per tornare indietro di una cartella:

```
$ cd ..
```

Per copiare un file:

```
$ cp {nomefile} {nomefilenuovo}
```

Per spostare o rinominare un file:

```
$ mv {nomefile} {nomefilenuovo}
```

Per cancellare un file:

```
$ rm {nomefile}
```

Per copiare, spostare o cancellare un'intera cartella useremo il parametro -r. Nel caso della cancellazione il comando sarà:

```
$ rm -r {nomecartella}
```

Per creare una cartella:

```
$ mkdir {nomecartella}
```

Per usare un editor di testo (useremo la combinazione di CTRL+X per chiudere, tasto S per confermare e INVIO per salvare il file; qualora non esista il file, ne verrà creato uno nuovo):

```
$ nano {nomefile}
```

Questi e altri programmi sono spesso documentati. Per accedere alla documentazione di essi si potrebbe usare il parametro --help:

```
$ ls -- help
```

Se è presente un'integrazione con man possiamo testare anche il comando:

```
$ man ls
```

Nei due esempi precedenti otterremo la documentazione relativa a ls, il programma che permette di listare file, directory, permessi e così via.

Potremmo voler decidere di installare un programma all'interno della nostra Debian (o derivata); in questo caso il comando da utilizzare è:

```
$ apt install {nomeprogramma}
```

Oppure decidere di rimuoverlo:

```
$ apt remove {nomeprogramma}
```

Il comando apt è in grado anche di aggiornare i repository della nostra distribuzione:

```
$ apt update
```

E anche di aggiornare tutti i programmi:

```
$ apt upgrade
```

Aggiornare sia repository che programmi è un'operazione spesso eseguita in contemporanea, ecco perché possiamo concatenare i due programmi con l'operatore &&:

```
$ apt update && apt upgrade
```

Il comando apt tuttavia andrebbe lanciato da root; per farlo possiamo anteporre il comando sudo:

```
$ sudo apt update
```

Oppure accedere come utente root (se si conosce la password dell'utente root):

```
$ su
```

O elevare il nostro utente a sudo (se presente nella lista sudoers):

```
$ sudo -s
```

Sebbene non ufficialmente supportati da questo libro è possibile che molto di quello che è stato descritto funzioni anche per Sistemi Operativi basati su distribuzioni differenti; una delle maggiori differenze che potremmo trovare è l'installazione dei pacchetti, per il resto (come i comandi sopra descritti) non dovremmo avere problemi.

Sui Sistemi Operativi a base Red Hat (Fedora, CentOS etc...) il comando per installare un programma è:

```
$ yum install {nomeprogramma}
```

Mentre per i Sistemi Operativi a base Arch Linux il comando per installare un programma è:

```
$ pacman -S [nomeprogramma]
```

La maggior parte dei programmi non pre-installati saranno disponibili su GitHub o GitLab. Per scaricare il source il comando è:

```
$ git clone [url.git]
```