### 7.9.4 File Disclosure

**Come riconoscerla**

Si ha quando l'applicazione si affida all'input dell'utente per decidere a quali file o directory accedere. Se l'input non viene verificato né bonificato, un malintezionato può scegliere di leggere file arbitrari oltre quelli previsti, divulgando il contenuto di questi file.
Questa vulnerabilità è sovrapponibile al path traversal, cui ci si riferisce di solito per indicare l'abusto di percorsi in input.

**Come difendersi**

- Occorre prendere in considerazione l'utilizzo di una soluzione statica per la lettura di file, ad esempio un elenco di file consentiti da cui poter scegliere. Oppure si potrebbe utilizzare un database, al posto di file e directory.
- Se la lettura di file locali dal disco è assolutamente necessaria, assicurarsi che i file vengano letti da una cartella specifica e limitare l'accesso al codice solo a questa cartella; inoltre, quando un utente fornisce un percorso, è necessario ripulire la stringa dagli eventuali metacaratteri tipici del file system, come le barre e i punti, per impedire ogni tentativo di manipolazione del percorso per accedere a una directory riservata.

Esempio:
Codice vulnerabile:

```
if (isset($_GET['imagenamè])) {
          $filename = $_GET['imagenamè]; // qui un attaccante può fornire un
      percorso
                                          // assoluto come "/etc/passwd"
  readfile($filename);
}
```

La versione sicura toglie di mezzo il path, rendendo impossibile l'abuso:

```
if (isset($_GET['imagenamè])) {
  $filename = getcwd()."/images/".basename($_GET['imagenamè]);
  readfile($filename);
}
```

Per ulteriori informazioni si veda: https://cwe.mitre.org/data/definitions/538.html
CWE-538: File and Directory Information Exposure

### 7.9.5 Remote File Inclusion

**Come riconoscerla**

Un malintenzionato potrebbe tramite questa vulnerabilità avere accesso alle librerie di sistema presenti sul server. Se non adeguatamente protette, potrebbero essere attaccate librerie di sistema installate sul server (ad esempio in caso di attacco nella fase di caricamento delle stesse librerie) rendendo il sistema completamente sotto controllo dell'attaccante.
Ciò può accadere perché l'applicazione utilizza i dati non attendibili ricevuti tramite l'input dell'utente per caricare dinamicamente la libreria, senza una corretta sanitizzazione. Il framework malevolo caricherà qualsiasi codice arbitrario specificato dall'applicazione, e potrebbe anche scaricare file di codice remoto ospitati su un server esterno, se specificato. Il codice caricato verrà quindi eseguito come se fosse un software assolutamente affidabile rendendo il sistema estremamente vulnerabile.

**Come difendersi**

- Non caricare in modo dinamico le librerie relative a codice software, in particolare basate sull'input non controllato dell'utente.