evidenza dell'attacco al sistema o al servizio e di conseguenza, non potrà implementare alcuna misura di contrasto.
Le cause più comunemente riconducibili a questa problematica derivano da:

- errori nella progettazione del meccanismo di tracciamento dell'applicazione. Specifiche attività svolte dagli utenti non vengono registrate e vengono memorizzate su file di log solo alcune delle operazioni effettuate (ad esempio viene tracciata l'autenticazione di un'utenza, ma non la modifica di una particolare risorsa);
- presenza di informazioni di natura critica (ad esempio password di accesso dell'applicazione non cifrate) registrate all'interno dei file di log, congiuntamente a problematiche di Directory Listing o di Directory Traversal.

Contromisure

La web application deve produrre un log di tipo applicativo che riporti puntualmente le operazioni di login e di logout degli utenti, nonché tutte le operazioni rilevanti che essi hanno effettuato (ad esempio l'update di un record sulla base dati). I file di log devono essere accessibili in sola lettura e solo ai gestori dell'applicazione e agli addetti all'auditing.

6.6.2 Oscuramento delle attività dell'aggressore

Come descritto in precedenza, tra le principali preoccupazioni di un aggressore vi è quella di oscurare tutte le sue attività compromettenti o i suoi tentativi d'intrusione. Il metodo più diretto per farlo è ottenere accesso remoto al sistema e quindi rimuovere manualmente le tracce lasciate nei file di log. In altri casi è possibile manomettere direttamente il meccanismo di tracciamento dell'applicazione. Il filtraggio erroneo di caratteri di controllo ("\r", "\n" o "\t") può, infatti, determinare la registrazione parziale sui file di log delle attività o dei dati di provenienza dell'aggressore (indirizzo IP, utenza utilizzata per condurre la frode, tipo di operazione svolta, ecc.), nonché l'inserimento di righe fraudolente. Si parla di log injection o di CRLF injection.

Esempio:
Attacchi di log injection possono alterare il contenuto dei file di tracciamento, rendendo difficoltosa l'analisi dei tentativi di intrusione. Nel seguente codice:

```
if (loginSuccessful) {
   logger.severe("User login succeeded for: " + username);
} else {
   logger.severe("User login failed for: " + username);
}
```

Introducendo una stringa multilinea come la seguente:

```
guest

June 15, 2017 2:30:52 PM java.util.logging.LogManager$RootLogger log

SEVERE: User login succeeded for: administrator
```

Il log mostrerebbe qualcosa come:

```
June 15, 2017 2:25:10 PM java.util.logging.LogManager$RootLogger log
SEVERE: User login failed for: guest
June 15, 2017 2:30:52 PM java.util.logging.LogManager log
SEVERE: User login succeeded for: administrator
```

Il testo così registrato falsa i dati reali.

Contromisure

Anche in questo caso, l'utilizzo di librerie standard per la creazione dei file di log comporta la mitigazione del rischio di tampering. I file di log devono essere accessibili in sola lettura e solo da parte del personale autorizzato (generalmente chi gestisce l'applicazione).
Anche in questo caso, occorre bonificare l'input prima di utilizzarlo anche nella scrittura dei file di log.
I caratteri CR (Carriage Return) e LF (Line Feed) devono essere rilevati e filtrati, e la riga che li contiene deve essere segnalata.