Attraverso lo sfruttamento di altre vulnerabilità, quali la SQL injection, il buffer overflow, il directory listing e altre, un aggressore può introdursi nel sistema e carpire queste informazioni.
Non direttamente correlabile con problematiche crittografiche in senso stretto, la tecnica di File system Polling viene spesso utilizzata da un aggressore con accesso locale ad un sistema per appropriarsi dei dati fintanto che essi permangono memorizzati su disco in forma non cifrata. Questa condizione si verifica quando tali dati vengono temporaneamente salvati per lunghi periodi in tabelle di staging o in punti ben precisi del filesystem, prima di essere definitivamente cifrati. L'aggressore, utilizzando script automatici, può copiare ciclicamente il contenuto di queste tabelle e directory in locazioni del disco differenti e mantenere i relativi dati in forma intelligibile per i suoi scopi.
Esempio:
È banale accedere a un file non cifrato, contenente dati elaborati, collocato in una directory raggiungibile del file system.
L'esecuzione del comando `more /usr/app/data/accounts.txt` rivela i dettagli degli account che non dovrebbero essere divulgati.

Contromisure

Occorre applicare le misure di sicurezza citate in precedenza per impedire le problematiche che permettono agli attaccanti di raggiungere il file system. I file e i dati sensibili o cruciali devono essere salvati nel filesystem in collocazioni dotate permessi restrittivi, solo dopo averli correttamente criptati con un algoritmo di crittografia "forte".

6.4 Gestione degli errori, delle eccezioni

La gestione degli errori, delle eccezioni o delle circostanze fuori dalla norma sono tutti quanti aspetti frequentemente trascurati dagli sviluppatori di software. La non corretta implementazione delle eccezioni può indurre l'applicazione a:

- bloccarsi o sospendersi;
- rilasciare informazioni utili all'aggressore per avanzare con successo nella sua azione intrusiva nel sistema;
- permettere all'aggressore di acquisire il controllo diretto del sistema o dell'applicazione.

Esempio:
Se l'applicazione non gestisce bene l'errore, le indicazioni che possono essere mostrate possono fornire molte informazioni all'attaccante, sia sull'applicazione, sia sull'ambiente nel quale gira. Ad esempio si guardi il seguente `stack overflow` mostrato in chiaro sulla pagina web, in seguito a un errore dell'applicazione:

```
Exception sending context initialized event to listener instance of class
com.selexes.gcm.server.MyServletContextListener java.lang.ArithmeticException: /
by zero at
com.selexes.gcm.server.MyAppServerBase.<init>(MyAppServerBase.java:46)at
com.insecurefirm.MyApp.server.MyAppServerXmpp.<init>(MyAppServerXmpp.java:33) at
com.insecurefirm.MyApp.server.MyAppServerXmpp.getInstance(MyAppServerXmpp.java:77)
at
com.insecurefirm.MyApp.server.MyAppServerFactory.<init>(MyAppServerFactory.java:76
)at
com.insecurefirm.MyApp.server.MyAppServerFactory.getInstance(MyAppServerFactory.ja
va:27)at
com.insecurefirm.MyApp.server.MyServletContextListener.contextInitialized(MyServle
tContextListener.java:34)at
org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4812)a
t
org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5255)a
t org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:147)at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1408)at
org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1398)at
java.util.concurrent.FutureTask.run(Unknown Source)at
```