



SECURE CODE WARRIOR

DEBUG FEATURES ENABLED

We'll go through

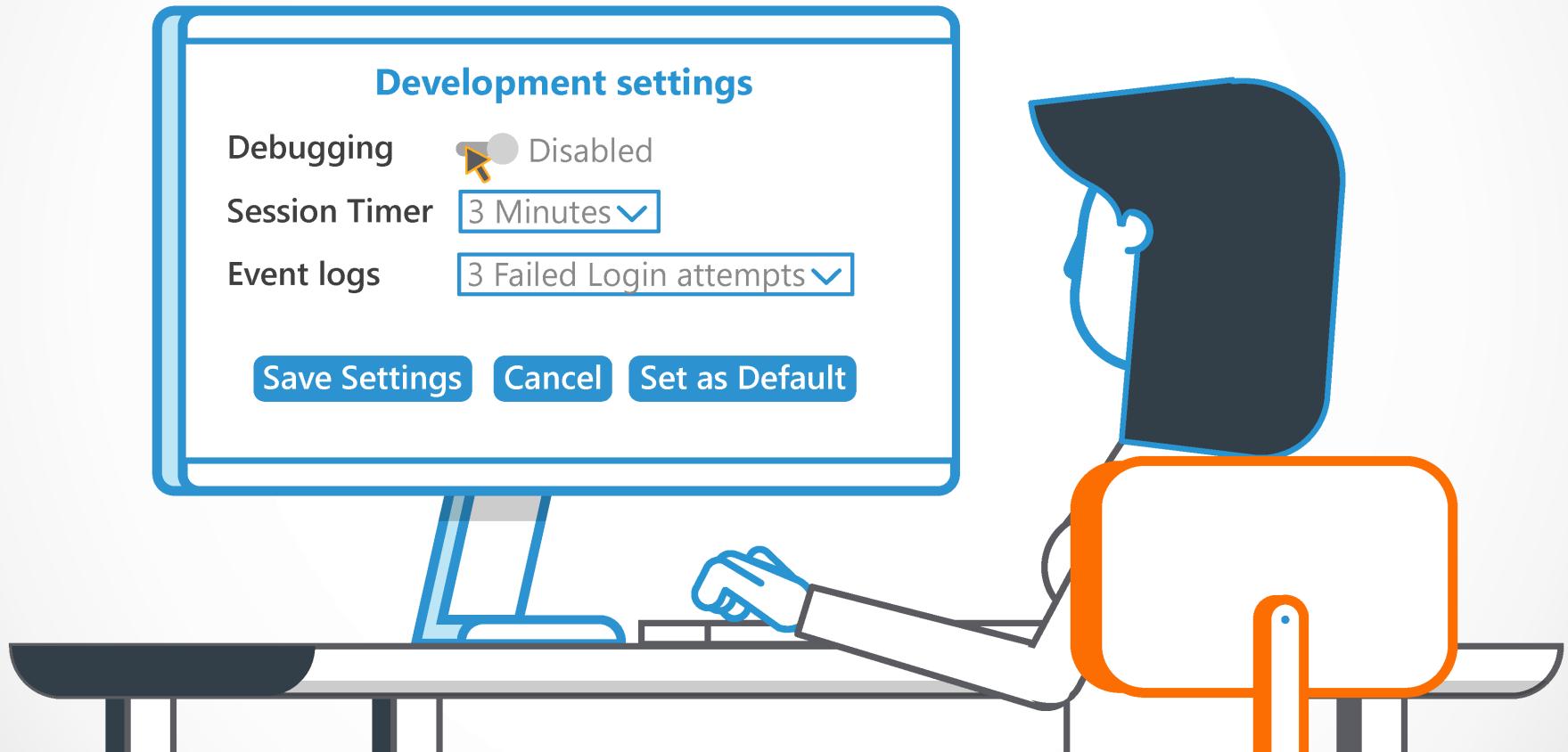
some causes and preventions of
vulnerabilities in this category

**WHAT DO WE MEAN BY
DEBUG FEATURES ENABLED**

Debug Features are features that help developers know the internal state of an application while investigating the cause of errors.



Generally, these should not be enabled in a Production environment.



LET'S LOOK AT AN EXAMPLE

A web application is correctly configured to catch any errors and forward them to a generic error page.



Application Server Logs

```
Main [Java Application]
Main at localhost:59588
Thread [main] [Suspended]
ClassNotFoundException(Throwable).<init>(String, Throwable) line: 286
ClassNotFoundException(Exception).<init>(String, Throwable) line: not available
ClassNotFoundException(ReflectiveOperationException).<init>(String, Throwable) line: not available
ClassNotFoundException.<init>(String) line: not available
URLClassLoader$1.run() line not available
URLClassLoader$1.run() line not available
AccessController.doPrivileged(PrivilegedExceptionAction<T>, AccessControlContext) line: not available
Launcher$ExtClassLoader(URLClassLoader).findClass(String) line: not available
Launcher$ExtClassLoader(ClassLoader).loadClass(String, boolean) line: not available
Launcher$AppClassLoader(ClassLoader).loadClass(String, boolean) line: not available
Launcher$AppClassLoader.load.Class(String, boolean) line: not available
Launcher$AppClassLoader(ClassLoader).loadClass(String) line: not available
Main.main(String[])
Main [Java Application]
C:\Program Files\Java\jre7\bin\javaw.exe (May 8, 2012 9:12:37 PM)
```

However, in the latest update, a flag for development mode has been included in the production environment.



As a result, when errors come up in the application, the full stack trace is being returned.

Application Server Logs

```
Main [Java Application]
Main at localhost:59588
Thread [main] [Suspended]
ClassNotFoundException(Throwable).<init>(String, Throwable) line: 286
ClassNotFoundException(Exception).<init>(String, Throwable) line: not available
ClassNotFoundException(ReflectiveOperationException).<init>(String, Throwable) line: not available
ClassNotFoundException.<init>(String) line: not available
URLClassLoader$1.run() line not available
URLClassLoader$1.run() line not available
AccessController.doPrivileged(PrivilegedExceptionAction<T>, AccessControlContext) line: not available
Launcher$ExtClassLoader(URLClassLoader).findClass(String) line: not available
Launcher$ExtClassLoader(ClassLoader).loadClass(String, boolean) line: not available
Launcher$AppClassLoader(ClassLoader).loadClass(String, boolean) line: not available
Launcher$AppClassLoader.loadClass(String, boolean) line: not available
Launcher$AppClassLoader(ClassLoader).loadClass(String) line: not available
Main.main(String[])
Main [Java Application]
C:\Program Files\Java\jre7\bin\javaw.exe (May 8, 2012 9:12:37 PM)
```



This leaks information about the application and it's runtime environment.

Application Server Logs

Main [Java Application]

Main at localhost:59588

Thread [main] [Suspended]

ClassNotFoundException(Throwable).<init>(String, Throwable) line: 286

ClassNotFoundException(Exception).<init>(String, Throwable) line: not available

ClassNotFoundException(ReflectiveOperationException).<init>(String, Throwable) line: not available

ClassNotFoundException.<init>(String) line: not available

URLClassLoader\$1.run() line not available

URLClassLoader\$1.run() line not available

AccessController.doPrivileged(PrivilegedExceptionAction<T>, AccessControlContext) line: not available

Launcher\$ExtClassLoader(URLClassLoader).findClass(String) line: not available

Launcher\$ExtClassLoader(ClassLoader).loadClass(String, boolean) line: not available

Launcher\$AppClassLoader(ClassLoader).loadClass(String, boolean) line: not available

Launcher\$AppClassLoader.load.Class(String, boolean) line: not available

Launcher\$AppClassLoader(ClassLoader).loadClass(String) line: not available

Main.main(String[]) line: 7

Main [Java Application]

C:\Program Files\Java\jre7\bin\javaw.exe (May 8, 2012 9:12:37 PM)



A hacker can now use the captured data to look for weak points in the host platform, the application or it's dependencies and better plan their attack.



**To avoid attacks relating to Debug
Enabled Features, developers should**

- ④ ensure any information related to debugging is logged securely to the server, and not leaked to the user.
- ④ Additionally, for any compiled or obfuscated code, ensure that the obfuscation map and the debug symbols are inaccessible to end users.

Congratulations, you have now completed this module!



SECURE CODE WARRIOR

www.securecodewarrior.com