

```
public static boolean reachable(String name, int port) {
    try(Socket ignored = new Socket(InetAddress.getByName(name), port)) {
        return true;
    } catch (Exception e) {
        return false;
    }

    public static String currentWorkingDirectory() {
        String location;
        File file = new File(".");
        try {
            location = file.getCanonicalPath();
        } catch (IOException e) {
            location = file.getAbsolutePath();
        }
        return location;
    }

    public static int getIntProperty(String propertyName, int defaultValue) {
        try {
            return Integer.parseInt(System.getProperty(propertyName));
        } catch (NumberFormatException e) {
            return defaultValue;
        }
    }
}
```

"base/src/main/java/com/thoughtworks/go/util/SystemUtil.java" 177L, 6350C 136, 77

Unencrypted socket to com.thoughtworks.go.util.SystemUtil (instead of SSLSocket)

The communication channel used is not encrypted. The traffic could be read by an attacker intercepting the network traffic.

Vulnerable Code:

Plain socket (Cleartext communication):

```
Socket soc = new Socket("www.google.com",80);
```

Solution:

SSL Socket (Secure communication):

```
Socket soc = SSLSocketFactory.getDefault().createSocket("www.google.com", 443);
```

Beyond using an SSL socket, you need to make sure your use of SSLSocketFactory does all the appropriate certificate validation checks to make sure you are not subject to man-in-the-middle attacks. Please read the OWASP Transport Layer Protection Cheat Sheet for details on how to do this correctly.

References

OWASP: Top 10 2010-A9-Insufficient Transport Layer Protection

OWASP: Top 10 2013-A6-Sensitive Data Exposure

OWASP: Transport Layer Protection Cheat Sheet

WASC-04: Insufficient Transport Layer Protection

CWE-319: Cleartext Transmission of Sensitive Information