Password Management: Empty Password Mitigation SOP

# Development Mitigation SOP

Empty passwords may compromise system security in a way that cannot be easily remedied. It is never a good idea to have an empty password. It also makes fixing the problem extremely difficult once the code is in production. The password cannot be changed without patching the software. If the account protected by the empty password is compromised, the owners of the system will be forced to choose between security and availability.

# Defense Against [DEFECT]

Avoid empty passwords in source code and avoid using default passwords. If an empty password is the default, require that it be changed and remove it from the source code. When identifying null, empty, or hardcoded passwords, default rules only consider fields and variables that contain the word password.

# Examples

…

String storedPassword = “”;

String temp;

if ((temp = readPassword()) != null) {

storedPassword = temp;

}

If(storedPassword.equals(userPassword))

//Access protected resources

…

}

…

## Explanation

The code above initializes a password variable to an empty string, attempts to read a stored value for the password, and compares it against a user-supplied value. If readPassword() fails to retrieve the stored password due to a database error or another problem, then an attacker could trivially bypass the password check by providing an empty string for userPassword.

**Resources**

1. [Fortify Taxonomy: Software Security Errors](https://vulncat.fortify.com/en/detail?id=desc.semantic.java.password_management_empty_password#Java%2FJSP)