rules.sonarsource.com

C++ static code analysis: Account validity should be verified when authenticating users with PAM

2 minutes

Pluggable authentication module (PAM) is a mechanism used on many unix variants to provide a unified way to authenticate users, independently of the underlying authentication scheme.

When authenticating users, it is strongly recommended to check the validity of the account (not locked, not expired …), otherwise it leads to unauthorized access to resources.

Noncompliant Code Example

The account validity is not checked with `pam_acct_mgmt` when authenticating a user with `pam_authenticate`:

```
int valid(pam_handle_t *pamh) {
    if (pam_authenticate(pamh,
PAM_DISALLOW_NULL_AUTHTOK) != PAM_SUCCESS)
{ // Noncompliant - missing pam_acct_mgmt
        return -1;
    }

    return 0;
```

```
}
```

The return value of `pam_acct_mgmt` is not checked:

```
int valid(pam_handle_t *pamh) {
    if (pam_authenticate(pamh,
PAM_DISALLOW_NULL_AUTHTOK) != PAM_SUCCESS)
{
        return -1;
    }
    pam_acct_mgmt(pamh, 0); // Noncompliant
    return 0;
}
```

Compliant Solution

When authenticating a user with `pam_authenticate`, check the account validity with `pam_acct_mgmt`:

```
int valid(pam_handle_t *pamh) {
    if (pam_authenticate(pamh,
PAM_DISALLOW_NULL_AUTHTOK) != PAM_SUCCESS)
{
        return -1;
    }
    if (pam_acct_mgmt(pamh, 0) != PAM_SUCCESS) { //
Compliant
        return -1;
    }
    return 0;
}
```

See

- OWASP Top 10 2021 Category A7 - Identification and Authentication Failures
- OWASP Top 10 2017 Category A5 - Broken Access Control
- MITRE, CWE-304 - Missing Critical Step in Authentication