POS02-C. Follow the principle of least privilege

The principle of least privilege states that every program and every user of the system should operate using the least set of privileges necessary to complete the job [Saltzer 1974, Saltzer 1975]. The Build Security In website [DHS 2006] provides additional definitions of this principle. Executing with minimal privileges mitigates against exploitation in case a vulnerability is discovered in the code.

Noncompliant Code Example

Privileged operations are often required in a program, though the program might not need to retain the special privileges. For instance, a network program may require superuser privileges to capture raw network packets but may not require the same set of privileges for carrying out other tasks such as packet analysis. Dropping or elevating privileges alternately according to program requirements is a good design strategy. Moreover, assigning only the required privileges limits the window of exposure for any privilege escalation exploit to succeed.

Consider a custom service that must bind to a well-known port (below 1024). To prevent malicious entities from hijacking client connections, the kernel imposes a condition so that only the superuser can use the bind() system call to bind to these ports.

This noncompliant code example is configured as setuid-superuser. It calls bind() and later forks out a child to perform the bookkeeping tasks. The program continues to run with superuser privileges even after the bind() operation is completed.

```c
int establish(void) {
    struct sockaddr_in sa; /* listening socket’s address */
    int s; /* listening socket */

    /* Fill up the structure with address and port number */
    sa.sin_port = htons(portnum);
    /* Other system calls like socket() */
    if (bind(s, (struct sockaddr *)&sa, sizeof(struct sockaddr_in)) < 0) {
        /* Perform cleanup */
    }

    /* Return */
}

int main(void) {
    int s = establish();

    /* Block with accept() until a client connects */
    switch (fork()) {
    case -1 : /* Error, clean up and quit */
    case 0 : /* This is the child, handle the client */
    default : /* This is the parent, continue blocking */
    }
    return 0;
}
```

If a vulnerability is exploited in the main body of the program that allows an attacker to execute arbitrary code, this malicious code will run with elevated privileges.

Compliant Solution

The program must follow the principle of least privilege while carefully separating the binding and bookkeeping tasks. To minimize the chance of a flaw in the program from compromising the superuser-level account, it should drop superuser privileges as soon as the privileged operations are completed. In the following code, privileges are permanently dropped as soon as the bind() operation is carried out. The code also ensures privileges may not be regained after being permanently dropped, as in POS03-C. Ensure that privilege relinquishment is successful.
Risk Assessment

Failure to follow the principle of least privilege may allow exploits to execute with elevated privileges.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS02-C</td>
<td>High</td>
<td>Likely</td>
<td>High</td>
<td>P9</td>
<td>L2</td>
</tr>
</tbody>
</table>

Automated Detection

<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Checker</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klocwork</td>
<td>2018</td>
<td>SV.BRM.HKEY_LOCAL_MACHINE</td>
<td>^</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SV.USAGERULES.PERMISSIONS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related Vulnerabilities

CVE-2009-2031 results from a violation of this recommendation. OpenSolaris, in smbsfs snv_84 through snv_110, sets permissions based on mount-point options and not actual user information (obtained from the getuid() and getgid() functions). An attacker can exploit this to achieve higher permissions. Also, in a certain initialization mode, the code grants read, write, and execute permissions to users other than the owner, which can be exploited to make files world readable [xorl 2009].

Search for vulnerabilities resulting from the violation of this rule on the CERT website.
Related Guidelines

<table>
<thead>
<tr>
<th>ISO/IEC TR 24772</th>
<th>Adherence to Least Privilege [XYN]</th>
</tr>
</thead>
</table>
| MITRE CWE        | CWE-250, Execution with unnecessary privileges  
CWE-272, Least privilege violation |

Bibliography

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>[DHS 2006]</td>
<td>Least Privilege</td>
</tr>
<tr>
<td>[Saltzer 1974]</td>
<td></td>
</tr>
<tr>
<td>[Saltzer 1975]</td>
<td></td>
</tr>
<tr>
<td>[Wheeler 2003]</td>
<td>Section 7.4, &quot;Minimize Privileges&quot;</td>
</tr>
<tr>
<td>[xorl 2009]</td>
<td>&quot;OpenSolaris CIFS/SMB Invalid File Flags&quot;</td>
</tr>
</tbody>
</table>