Attack Surfaces Model

Version 2: Edits highlighted

Natasha Kubiak

ECE 531, SUMMER 2022

DAEMON - SAMPLED.C

Run time:

Runs as Root

Problems: Program runs as the Root User. Root has complete permissions to read/write/modify/delete files.Attackers can exploit the elevated privileges of the root user like modifying sensitive files, and creating files in your system without you knowing that could be a backdoor into your system

Solution: You should give the least amount of privilege to an application, and it should only have enough privileges so that it can run and complete the tasks that it needs to. Avoid having read/write permissions.

Variables:

#define:

DAEMON_NAME
OK
ERR_SETSID
ERROR
ERR_FORK
ERR_CHDIR
True
SIGTERM
SIGHUP

char

*ERROR_FORMAT

Functions

void _do_work(void)

void _signal_handler(const int signal)

- Int max size : 2147483647

Library Functions

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <unistd.h>
#include <syslog.h>
#include <string.h>
#include <time.h>
```

Problems: Program loads and executes functions from an external library. Could have vulnerabilities you are unaware of that attackers are looking to exploit, or you may use a function incorrectly, leaving you open to attack or even facilitating an attack to happen.

Solution: Make sure you are using library functions correctly and how they were intended and documented to be used. Look into the attack risks of the library's and 3rd party dependencies you are using. Also check on when they were last modified/updated. When using a function be aware of permissions it is granting or exploiting.

Inputs

Signal:

SIGTERM:Terminated. A gentle kill that gives processes a chance to clean up.

SIGHUP:Hangup. Usually means that the controlling terminal has been disconnected.

Problems: Any unhandled signals simply returns to the log “unhandled signal”

Solution: Handle other signals, by closing log and exiting.

Standard input output are closed, no console.

```
close(STDIN_FILENO)
close(STDOUT_FILENO)
close(STDERR_FILENO)
```

Outputs

syslog

outputs to: var/log/messages , var/log/syslog

Problems:

- Large amounts of data to syslog can fill disk space.
- Logs are no longer saved when space is full, attacks would not leave a trail.

Solution

- Only listen on localhost to mitigate attack
- _do_work does not run indefinitely, will run for 100 seconds.