Computer Systems Security: Winter 2018 Experiences

Contents

Thinking

Common Tools

List computer security tools/mechanisms that you use on a regular basis. For each item, indicate with a * if you believe you have a poor understanding of how it works or what its true purpose is. (If you think you understand it reasonably well, just list it without a *.)

Doing

Replacement passwd program

Try to write a replacement for `passwd` that is standard on most UNIX-like systems (such as most Linux distributions). Your program can be written in any language; if you write it in a scripting language, however, you may have to use a C wrapper in order to make it work as setuid root.

Creating a properly hashed password may be tricky; as a stepping stone towards `passwd`, try writing `chsh` (change shell) or `chfn` (change finger information).

You only need to provide basic functionality. The only command line argument your program must take is an optional username. Note that your `passwd` program should only allow root to change an arbitrary user's password; otherwise, it should only allow changing of the password for the current user.

Optionally, try to drop all unnecessary privileges. You may need to install libcap (libcap-dev on Debian/Ubuntu systems) so you can use `cap_get_proc()` and `cap_set_proc()`.

Setuid root binaries and capabilities

Choose at least four setuid root binaries, each of which requires a distinct set of Linux capabilities. Below is a sample list:


```
passwd
mount
ping
sudo
```


You can look at alternatives, e.g., `su` instead of `sudo`, but don't look at both as they require similar capabilities.

For each binary, determine what capabilities it requires to function properly. How can you figure this out without reading the source?

Try removing the setuid bit from the file and replacing it with file-based capabilities using `setcap`. Can you get it to work as before?

Can you create a program that, given one capability, is able to get other capabilities that it wasn't explicitly given? Which capabilities are useful for getting other capabilities?

Restricting network access

It is possible to restrict network access using application-specific configurations, generalized userspace restrictions (e.g., TCP wrappers), host (kernel) level firewalls, and network firewalls.

For a specific application or protocol, implement rules to restrict access at the application, host, and network levels. Verify that your rules are working properly, i.e., that the protocol is being blocked where you think it is being blocked.

Once you've implemented a complete block, implement a partial block (e.g., allow some hosts and restrict other hosts).

Note that while you can do this exercise in a virtual environment, it is probably easier to do it on a regular network you control (e.g., a home network).

Reading

Foundational Security Papers 1