CIS 418/518 – Secure Software Engineering

Environment Variables and Attacks (Chapter 3)

1. What is the difference between environment variables and shell variables?

2. In Bash, if we run "`export foo=bar`", does it change the environment variable of the current process? How would you verify this?

3. The following are two different ways to print out the environment variables. Describe their differences.

```
$ /usr/bin/env
$ /usr/bin/strings /proc/$$/environ
```

4. In our code, when we use `execve()` to execute an external program `xyz`, we pass NULL for the third argument. How many environment variables will the process running `xyz` has?

5. Bob says that he never uses any environment variables in his code, so he does not need to worry about any security problem caused by environment variables. Is he correct?

6. A program `abc` invokes an external program `xyz` using `system()` function, which is affected by the PATH environment variable. When we invoke `abc` from a shell prompt, how does the shell variable PATH in the current shell end up affecting the behavior of the `system()` function?

7. There are two typical approaches for letting normal users do privileged tasks. One approach is to write a root-owned Set-UID program and let the user run that program. Another approach is to use a dedicated root daemon to do those privileged tasks for the users. Compare the attack surface of these two approaches and describe which one is more secure.