Assignment 2: Due 11 April 2023
Application security

Security

Motivation

In this assignment project you will apply the knowledge about vulnerabilities from the lectures to exploit a test system.

Connection instructions

You can work alone or in pairs. The system to be exploited is an Ubuntu virtual machine at address `appsec2023.liacs.nl`. You can access it via `ssh`, first jumping to the LIACS `ssh` gateway:

1. Enrol into a group on Brigthspace (**AppSec groups**) with your partner or alone.
2. Send an email to `security2023@liacs.leidenuniv.nl` with your group number. You will receive a password for your group.
3. Use the terminal. First you `ssh sXXXXXXXX@ssh.liacs.nl`(Replace the X's with your student ID). Use the password for your student account.
4. Once connected, run the following command `ssh GroupY@appsec2023.liacs.nl` (Where Y is your group). Enter your group password. You are now connected to the vulnerable VM.
5. **Connection issues.** Sometimes your IP address can become blocked by the LIACS server. You will be getting a time out message if this is the case. Usually your IP is unblocked automatically in 30 minutes, but the automatic ban can be extended longer if you keep trying to connect. If after an hour you still are not able to connect, send the lecturer `o.gadyatskaya@liacs.leidenuniv.nl` your public IP address and the student account number to check with the system administrator. In case your student account is non-standard (you are an exchange student, etc.) and you cannot connect to the LIACS `ssh` gateway with it, please also contact the lecturer, mentioning the student ID, your public IP address, and the error message that you get.

System description

- The VM contains 10 levels, with an increasing difficulty. Each level is represented by some challenge binaries and other auxiliary files. You task is to exploit the binaries to obtain elevated shell privileges and move to the next level.
- The `/levels/level n` directories contain the challenges, with:
  - a `setgid` binary running as group `lev[n]`, where `[n]` stands for the level. I.e., this binary, if you can get it to do what you want, is your pass for obtaining the privileges for the next level.
  - the source code of this binary
- Challenge `n+1` will be accessible only if you are a member of the group `lev[n]`.
- You need to come up with a way to exploit the binaries to move to the next level (by becoming a member of the group for the next level).
- Useful binaries:

  - `escalate` permanently adds the current group ID to your user; call this binary from the obtained privileged shell. You may need to reconnect for `escalate` to have effect.

Tasks

When starting with the assignment, execute `escalate` once to get to the starting level `level0`.

For each level that you have found an exploit for:

- Your exploit executes `escalate` to advance your group in the privileges.
- Create a shell script `exploit.sh` that fully performs this exploit.
  - The script should include any dependencies/compilations that are required.
  - It should result into a privileged shell or a call to `escalate`.
  - For submission, it should be placed into a directory `level[n]` where `[n]` stands for the number of the level.
  - All source files `exploit.sh` might need should be positioned in the same directory.
  - No other files should be there (no target binaries).

Short PDF report

- Write a short PDF report summarizing the vulnerabilities you have found and successfully exploited, per each successfully attacked level.
- You need to include a short description of the vulnerability(s) and a short description of your exploit strategy.

Hints

- You need to know the Unix system access control principles (see, e.g., Sec. 5.3-5.5 in van Oorschot).
- You need to know the principles of shell scripting and command execution in shells.
- You need to consider the various vulnerability types that we studied (and study more about vulnerabilities yourself).
- Debugger is your friend; we suggest using `gdb`.
- You can check the current level using the `id` command, that will list the groups your user is a member of. You start working up from `level0` (execute `escalate` once to get into this group).

POLICY

- You will be using a shared machine hosted on the LIACS infrastructure.
- **Do not** attack anything but the target binaries.
- **Do not** overload the machine.
- **Do not** share your solutions with others in any way.
- We may kill your processes and delete your files if needed for other students' access.
- We may permanently disable your account and change your final grade (up to issuing to grade for this assignment) in case of deliberate offences.
- **Deliberate attacks to anything but the target binaries on this VM will be reported to the Examination Board and may result into punishments much more serious than a missing course grade.**

Submission

Your submission must be *one* `GroupY.zip` or `GroupY.tar.gz` archive file, where Y stands for your group number. It must contain a folder `Exploits` with exploit scripts, and a short pdf report.

Checklist to make sure that your report will meet our standards:

- ☐ Report is in pdf.
- ☐ Report is in English.
- ☐ Names and student numbers of all group members are indicated in the report.
- ☐ Exploit script folders are correctly structured (it is called `Exploits` and it contains subfolders called `level[n]` for all levels you have compromised; e.g. subfolders `level1, level2, level3`).

Evaluation criteria

This project will be evaluated as follows:

- Base grade = number of levels beaten fairly.
- Bonus grade for the first 5 groups who achieved the highest levels among all: bonus grade = 1 - $n/5$, where n is the number of groups who have more levels or who have beaten the levels earlier (by the time of submission on Brightspace).
- You are eligible for a bonus grade only if this is the first time you **and** your group peer are taking this course, and you have submitted before the deadline.
- Assignment grade = base grade + bonus grade.
- The grading system can be revised, based on the average performance of the class.