Web Security Project Part 2

CS155: Computer Security
Spring 2013
Due May 20

Defenses

Now that you've figured out how to hack the site, it's time to don your white hat and fix the vulnerabilities.

Setup

Download the VM image from the website. You can run this VM using the exact same process that you used for Project 1. There is one user login on this VM: "webserver" with password "webserver". If you need root access to do something you can just use the "sudo" command. The VM is running a webserver that is hosting a local copy of the zoobar web application. If for some reason you need to restart the web server just run "sudo service apache2 restart". You can access the site from outside of the VM by figuring out the IP address of your VM (run ifconfig and look for the field inet addr: under eth0) and then you can visit the site by just typing this IP address into your browser's URL bar. The code for the zoobar application is located in the /srv/http/ folder.

Goals

- Prevent attacks A, B, C, D and E (preventing attack E is required, even though the attack was extra credit). There might be more than one way of implementing these attacks, so don't stop once you can defend against your particular implementation. Think about other possible attack methods.

- Do not change the site's appearance or behavior on normal inputs.
- Do not add new files. Do not edit the files in the /includes/ directory. Do not add new database tables or columns.
- There are no specific requirements for error messages on bad input. You can sanitize the input or simply die(), as long as you note your decision in the README file. Sanitizing is probably the more user-friendly option.

Deliverables

Your solution will be a tarball of the /srv/http/ directory with your changes, including a README file that contains a list of all your changes: a brief description of the change (and where in the code you made it), what vulnerability(ies) it fixes, and any changes in functionality that might be observed. Your code will be tested by being uncompressed into a clean version of the VM and run, so please do not depend on anything not already present in the VM.

Late Policy

Every student in the class is given a total of 72 late hours that can be applied to the projects and homeworks. These late hours must be taken in chunks of 24 hours (essentially 3 late days) for example, submitting a homework 3 hours later than its due counts as 24 late hours used. After all your late hours are used up, the assignment score gets halved with every 24 hours the assignment is late for example, someone submitting a project 47 hours late after having used all her late days will get (1/2)2 = 1/4 of 4the grade. If a project has more than one part, each part is considered a separate assignment for late days for example, if you submit part 1 72 hours late and part 2 24 hours late, then part 1 gets full credit and part 2 gets 50% credit.

Note that no late hours can be used on the last programming project.