All this text wasn’t written by me (Tiago a.k.a TigaxMT).

All of it was transcribed by me from HackerOne video lessons.

All credits and thanks go to them.

Unchecked Redirects

An unchecked redirect is when a web application performs a redirect to an arbitrary URL outside the application.

This was seen in level4, where the form parameter deletes and votes allowed any URL to be used.

WHO CARES?

Many people responde with “Why does this matter?” when they learn about unchecked redirects. They seem completely useless at the start.
But what if you have a page that is using referer checks for authorization, for instance?
(Please, never do that)

SCENARIO

A simpler scenario is this:

An attacker creates an identical clone of your site, but instead of authenticating against your database, she just dumps login credentials to a file, then redirects back to your site.

With the use of unchecked redirect in the right place, such an attacker could send victims a link to your site, which then sends them to the evil site steal their credentials.

Unless the victims look at the URL after the redirect, they’ll never notice the problem.

DETECTING

Any time you see a redirect, look for the origin of the destination. Often, these will come from a user session in a way that is unexploitable. If, however , you find that it’s coming from the browser in a unsafe way – for instance, as part of a CSRFable POST request – you probably have an exploitable case.

MITIGATION

The mitigation is straightforward, but easy to get wrong.
One way to fix unchecked redirect is to not allow protocol specification in the destination. That is, remove instance of ‘http://’ an the like. This will mean that – at worst – a redirect can only cause a 404.

Another common mitigation is to do away with the redirect destination entirely, by constructing it on the server side from data the client sends.

For instance, in level4 it would be acceptable to send a destination ID along with the delete/vote call, to let the server know where you need to get back to.
This is, as always, a much safer option.