redirect script and see if the target parses the redirect and follows. What happens if you add sleep(1000) before the redirect, can you cause the server to hang and time out? Perhaps their filter is **only** checking the parameter value and **doesn't** check the redirect value and successfully allows you to read internal data. Don't forget to try using a potential open redirect you have discovered as part of your chain if they are filtering external websites.

Aside from looking for features on the website which takes a URL parameter, always hunt for any third-party software they may be using such as Jira. Companies don't always patch and leave themselves vulnerable so always stay up to date with the latest CVE's. Software like this usually contains interesting server related features which can be used for malicious purposes.

File uploads for stored XSS & remote code execution

There is a 99% chance the developer has created a filter as to what files to allow and what to block. I know before I've even tested the feature there will (*or at least should*) be a filter in place. Of course it depends on where they store the files but if it's on their main domain then the very first thing I will try to upload is a **.txt**, **.svg** and **.xml**. These three file types are sometimes forgotten about and slip through the filter. I first test for .txt to check how strict the filter actually is (if it says only images .jpg .png .gif are allowed for example) and then move on from there. As well as this just simply uploading three different image types (.png .gif and .jpg) can give you an indication as to how they are handling uploads. For example, are all photos saved in the same format regardless of the photo type we uploaded? Are they not trusting **any** of our input and always saving as **.jpg** regardless?

The approach to testing file upload filenames is similar to XSS with testing various characters & encoding. For example, what happens if you name the file "zseano.php/.jpg" - the code may see ".jpg" and think "ok" but the server actually writes it to the server as zseano.php and misses everything after the forward slash. I've also had success with the payload zseano.html%0d%0a.jpg. The server will see