- **WayBackMachine leaking old endpoints for account takeover**

When using WayBackMachine to scan for robots.txt I found an endpoint which was named similar to a past bug I had found. “SingleSignIn” and “DoubleSignIn”. The first initial bug I found **enabled me to supply the endpoint with a user’s ID and it would reveal the email associated with that account**. /SingleSignIn?userid=123. Since the newly discovered endpoint’s name was similar I simply tried the same parameter and checked what happened. **To my surprise, instead of revealing the email it actually logged me into their account!** This is an example of using past knowledge of how a website is working to find new bugs.

- **API Console blocked requests to internal sites but no checks done on redirects**

A very well known website provides a Console to test their API calls as well as webhook events. They were filtering requests to their internal host (such as localhost, 127.0.0.1) but these checks were only done on the field input. Supplying it with https://www.mysite.com/redirect.php which redirected to http://localhost/ bypassed their filtering and allowed me to query internal services as well as leak their AWS keys. If the functionality you are testing allows you to input your own URL then always test for how it handles redirects, there is always interesting behaviour! Features which are designed to send a request from the server should always be built with security in mind, so you should consider instantly, “*What has the developer put in place to prevent malicious activity?*”

- **Leaking data via websocket**

Most developers when setting up websocket functionality won’t verify the website attempting to connect to their WebSocket server. This means an attacker can use something like shown below to connect and send data to be processed, as well as receiving responses. Whenever you see websocket requests, always run basic tests to see if your domain is allowed to connect & interact.