# Objective 10 Now Hiring

This objective returns us to web vulnerabilities and finding them with browser webdev tools or Burp Suite.   
Graphical user interface, text

Description automatically generated

But first we need to visit Noxious O. D’or by the golden toilet.

## Terminal IMDS Exploration

Here is Noxious standing by the IMDS Exploration Terminal in Jack’s restroom.

|  |  |
| --- | --- |
| A picture containing text  Description automatically generated | Text  Description automatically generated |

This terminal is a walkthrough of an AWS application called Instance Metadata and User Data (IMDS) which is used to control AWS cloud instances (virtual machines, clusters, and the like.) The knowledge you gain here will be essential for completing Objective 10. The terminal does not require any instruction, other than “do it.”

## Hint after completing IMDS Exploration

There is just one hint from Noxious, which is a link to IMDS documentation.

|  |  |
| --- | --- |
| Graphical user interface  Description automatically generated  <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html> | Text  Description automatically generated |

## Objective 10 Help Wanted

Jack’s site gives you the opportunity to join the Frost Tower team. Because of a vulnerability it also gives you the opportunity to see what happens when a web server running on a cloud instance exposes sensitive information about the site’s internal workings. It also shows a great example of the vulnerability Server-Side Request Forgery (SSRF), which allows the attacker (us) to force the web site to make an HTTP request of our choosing.

## Step 1 question: SSRF

The site we are testing is <https://apply.jackfrosttower.com/>

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated | Graphical user interface, text, application, email  Description automatically generated |

The URL to prove that you have been naughty and have an NLBI report is the avenue to exploit the site. If you insert a URL for a valid image, you get this.  
Graphical user interface, text, application, chat or text message

Description automatically generated

However, if you submit a URL from the IMDS terminal, <http://169.254.169.254/latest/meta-data/>, you get this.  
Graphical user interface, text, application

Description automatically generated

Note that the middle of the page is trying to return a broken document. Hmm, I wonder what is in it.

Open <https://apply.jackfrosttower.com/> in your browser with webdev tools. Have the Network tab open before you submit an application. Submit an application with a simple URL from the IMDS lesson in the NLBI field. Do you get a broken image? Can you find that image in the Network tab and look at its raw contents?

Note: I had better luck in Firefox. Sometimes Chrome did not want to save a copy of the broken image.

### Step 1 answer

I picked <http://169.254.169.254/latest/meta-data/> more or less at random and entered it into the NLBI URL field. I had entered “a” as my first name and when the page came up, I saw a file named a.jpg. Right-click and save the file.

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| Image viewer says that a.jpg is not a jpeg file. Opening the file in a text editor gives interesting results. | ami-id  ami-launch-index  ami-manifest-path  block-device-mapping/ami  block-device-mapping/ebs0  block-device-mapping/ephemeral0  block-device-mapping/root  block-device-mapping/swap  elastic-inference/associations  elastic-inference/associations/eia-bfa21c7904f64a82a21b9f4540169ce1  events/maintenance/scheduled  events/recommendations/rebalance  hostname  iam/info  iam/security-credentials  iam/security-credentials/jf-deploy-role  <snip> |

Apparently, the site tried to display our invalid NLBI URL as an image, and put the results of the IMDS query, <http://169.254.169.254/latest/meta-data/>, into an image file.

### Step 2 question: Get the key

Using what you learned in the IMDS terminal and the SSRF problem you found in Step 1, download the secret access key. Use a different first name on every application you submit, or the site will not give you a new image file.

### Step 2 answer

The elf-deploy-role from the IMDS lesson does not exist on this server. You can walk the table with separate requests for iam and iam/security-credentials, then discover the role is jf-deploy-role and make the final request. Or you can look at the results in the Step 1 answers and go directly to the solution. Request <http://169.254.169.254/latest/meta-data/iam/security-credentials/jf-deploy-role>.

Using “x” as a first name this time, we fill out the form and see this.  
Graphical user interface, text, application

Description automatically generated

Save the file and open it in a text editor (or cat from a terminal).  
Graphical user interface, text, application

Description automatically generated

Nice! The secret access key is "CGgQcSdERePvGgr058r3PObPq3+0CfraKcsLREpX". We could use that to take over Jack’s AWS cloud installation! SSRF can be a nasty vulnerability.