**Walking An Application**

**Task 1 – Walking An Application**

In this room you will learn how to manually review a web application for security issues using only the in-built tools in your browser. More often than not, automated security tools and scripts will miss many potential vulnerabilities and useful information.

Here is a short breakdown of the in-built browser tools you will use throughout this room:

* View Source - Use your browser to view the human-readable source code of a website.
* Inspector - Learn how to inspect page elements and make changes to view usually blocked content.
* Debugger - Inspect and control the flow of a page's JavaScript
* Network - See all the network requests a page makes.

**Task 2 – Exploring The Website**

As a penetration tester, your role when reviewing a website or web application is to discover features that could potentially be vulnerable and attempt to exploit them to assess whether or not they are. These features are usually parts of the website that require some interactivity with the user.

Finding interactive portions of the website can be as easy as spotting a login form to manually reviewing the website's JavaScript. An excellent place to start is just with your browser exploring the website and noting down the individual pages/areas/features with a summary for each one.

An example site review for the Acme IT Support website would look something like this:

|  |  |  |
| --- | --- | --- |
| **Feature** | **URL** | **Summary** |
| Home Page | / | This page contains a summary of what Acme IT Support does with a company photo of their staff. |
| Latest News | /news | This page contains a list of recently published news articles by the company, and each news article has a link with an id number, i.e. /news/article?id=1 |
| New Article | /news/article?id=1 | Displays the individual news article. Some articles seem to be blocked and reserved for premium customers only. |
| Contact Page | /contact | This page contains a form for customers to contact the company. It contains name, email and message input fields and a send button. |
| Customer Login | /customer/login | This page contains a login form with username and password fields. |
| Customers | /customers | This link redirects to /customers/login. |
| Customer Signup | /customer/signup | This page contains a user-signup form that consists of a username, email, password and password confirmation input fields. |
| Customer Reset Password | /customers/reset | Password reset form with an email address input field. |
| Customer Dashboard | /customers | This page contains a list of the user's tickets submitted to the IT support company and a "Create Ticket" button. |
| Create Ticket | /customers/ticket/new | This page contains a form with a textbox for entering the IT issue and a file upload option to create an IT support ticket. |
| Customer Account | /customer/account | This page allows the user to edit their username, email and password. |
| Customer Logout | /customers/logout | This link logs the user out of the customer area. |

**Task 3 – Viewing The Page Source**

The page source is the human-readable code returned to our browser/client from the web server each time we make a request

The returned code is made up of HTML ( HyperText Markup Language), CSS ( Cascading Style Sheets ) and JavaScript, and it's what tells our browser what content to display, how to show it and adds an element of interactivity with JavaScript.

For our purposes, viewing the page source can help us discover more information about the web application.

How do I view the Page Source?

* While viewing a website, you can right-click on the page, and you'll see an option on the menu that says View Page Source.
* Most browsers support putting view-source: in front of the URL for example, view-source:https://www.google.com/
* In your browser menu, you'll find an option to view the page source. This option can sometimes be in submenus such as developer tools or more tools.

Let’s view some Page Source!

Try viewing the page source of the home page of the Acme IT Support website. Unfortunately, explaining everything you can see here is well out of the scope of this room, and you'll need to look into website design/development courses to understand it fully. What we can do, is pick out bits of information that are of importance to us.

At the top of the page, you'll notice some code starting with <!-- and ending with --> these are comments. Comments are messages left by the website developer, usually to explain something in the code to other programmers or even notes/reminders for themselves. These comments don't get displayed on the actual webpage. This comment describes how the homepage is temporary while a new one is in development. View the webpage in the comment to get your first flag.

**Task 4 – Developer Tools – Inspector**

Developer Tools

Every modern browser includes developer tools; this is a tool kit used to aid web developers in debugging web applications and gives you a peek under the hood of a website to see what is going on. As a pentester, we can leverage these tools to provide us with a much better understanding of the web application. We're specifically focusing on three features of the developer tool kit, Inspector, Debugger and Network.

Opening Developer Tools

The way to access developer tools is different for every browser. If you're not sure how to access it, click the "View Site" button on the top right of this task to get instructions to how to access the tools for your browser.

Inspector

The page source doesn't always represent what's shown on a webpage; this is because CSS, JavaScript and user interaction can change the content and style of the page, which means we need a way to view what's been displayed in the browser window at this exact time. Element inspector assists us with this by providing us with a live representation of what is currently on the website.

As well as viewing this live view, we can also edit and interact with the page elements, which is helpful for web developers to debug issues.

On the Acme IT Support website, click into the news section, where you'll see three news articles.

Right-clicking on the premium notice ( paywall ), you should be able to select the Inspect option from the menu, which opens the developer tools either on the bottom or right-hand side depending on your browser or preferences. You'll now see the elements/HTML that make up the website ( similar to the screenshots below ).

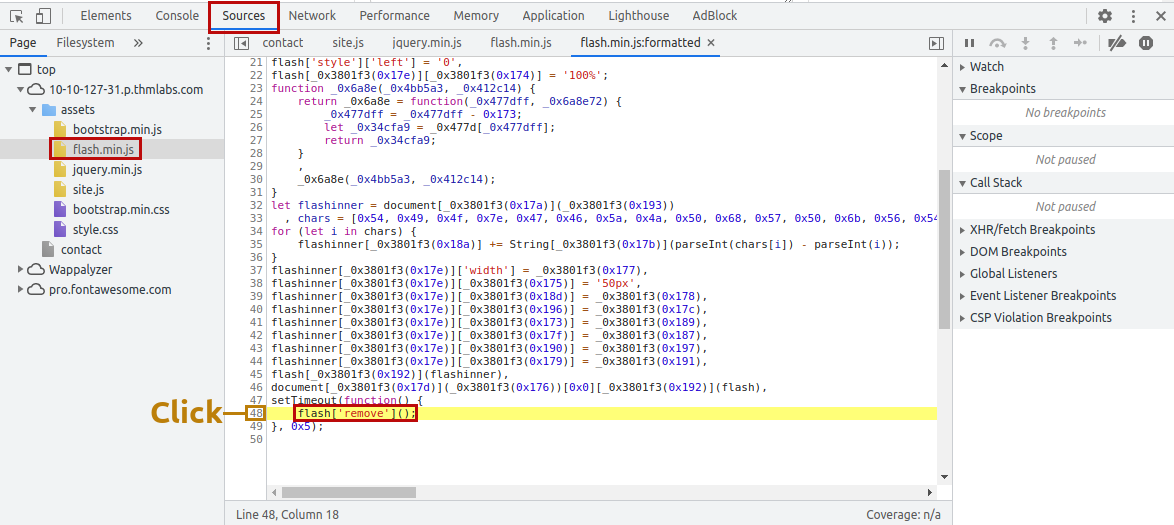


**Task 5 – Developer Tools - Debugger**

This panel in the developer tools is intended for debugging JavaScript, and again is an excellent feature for web developers wanting to work out why something might not be working. But as penetration testers, it gives us the option of digging deep into the JavaScript code. In Firefox and Safari, this feature is called Debugger, but in Google Chrome, it's called Sources.

On the Acme IT Support website, click on the contact page, each time the page is loaded, you might notice a rapid flash of red on the screen. We're going to use the Debugger to work out what this red flash is and if it contains anything interesting. Debugging a red dot wouldn't be something you'd do in the real world as a penetration tester, but it does allow us to use this feature and get used to the Debugger.

In both browsers, on the left-hand side, you see a list of all the resources the current webpage is using. If you click into the assets folder, you'll see a file named flash.min.js. Clicking on this file displays the contents of the JavaScript file.



We can return some of the formattings by using the "Pretty Print" option, which looks like two braces { } to make it a little more readable, although due to the obfustication, it's still difficult to comprehend what is going on with the file. If you scroll to the bottom of the flash.min.js file, you'll see the line: flash['remove']();

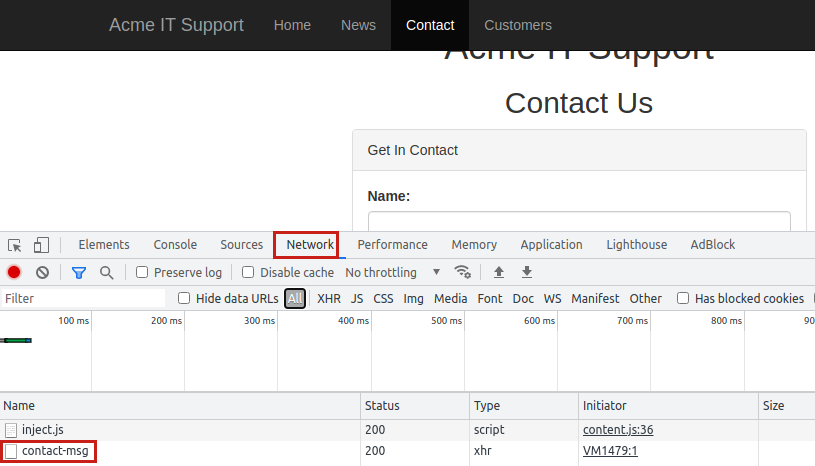
If you click the line number that contains the above code, you'll notice it turns blue; you've now inserted a breakpoint on this line. Now try refreshing the page, and you'll notice the red box stays on the page instead of disappearing, and it contains a flag.

**Task 6 – Developer Tools – Network**

The network tab on the developer tools can be used to keep track of every external request a webpage makes. If you click on the Network tab and then refresh the page, you'll see all the files the page is requesting.

Try doing this on the contact page; you can press the trash can icon to delete the list if it gets a bit overpopulated.

With the network tab open, try filling in the contact form and pressing the Send Message button. You'll notice an event in the network tab, and this is the form being submitted in the background using a method called AJAX. AJAX is a method for sending and receiving network data in a web application background without interfering by changing the current web page.



What is the flag shown on the contact-msg network request?

Flag -