A7 Insufficient Attack Protection Lab

In this lab we're going to run an automated attack against our site and then install some protections to protect against those same attacks.

Validating the prerequisites for our scanner

1. First, make sure perl is installed and is at least version 5.10.
   ```
   perl --version
   ```
2. Next, make sure you have a bash shell installed. If not, get one from http://git-scm.com
3. If you have those two things, you're ready to get a vulnerability scanner

Getting uniscan

Any vulnerability scanner will do fine for our purposes here. Unless you have a different choice, let's go get and install uniscan.

4. Go to https://sourceforge.net/projects/uniscan/ This is the online repository for uniscan.
5. Hit the big green Download button on the right.
6. The file downloaded will be called uniscanX.Y.tar.gz. Where the X and Y are version numbers. Copy the file to any temporary directory you like.
7. Open a bash shell and cd to that directory
8. Unzip the file like so
   ```
   gunzip uniscanX.Y.tar.gz
   tar -xvf uniscanX.Y.tar
   ```
   This will create a bunch of files
9. Do a test run of uniscan by running one of these commands ...
   ```
   ./uniscan.pl
   ```
   or
   ```
   perl uniscan.pl
   ```
10. If you get an error saying it can't find a module, edit uniscan.pl and find a line that says
    ```
    use lib "./Uniscan";
    ```
11. Comment out that line and change it to read...
    ```
    #use lib "./Uniscan";
    use FindBin qw ($RealBin);
    use lib $RealBin;
    ```
12. If you get an error about not being able to update, edit uniscan.conf and change the autoupdate setting from 1 to 0.
13. Now hopefully you've got uniscan installed and set up. Let's run a real scan.

Mounting the attack (aka. running a vulnerability scan)

14. Run (without debugging) your web application by hitting Ctrl-F5. This way it won't stop on any breakpoints.
15. Run a full scan against your website by
    ```
    uniscan -u http://localhost:7777 -qewds
    ```
16. Watch in awe as your uniscan looks for any of thousands of known vulnerabilities!
17. When it is finished, look for the output report and open it in a browser. Examine the results.

Installing an intrustion detection system

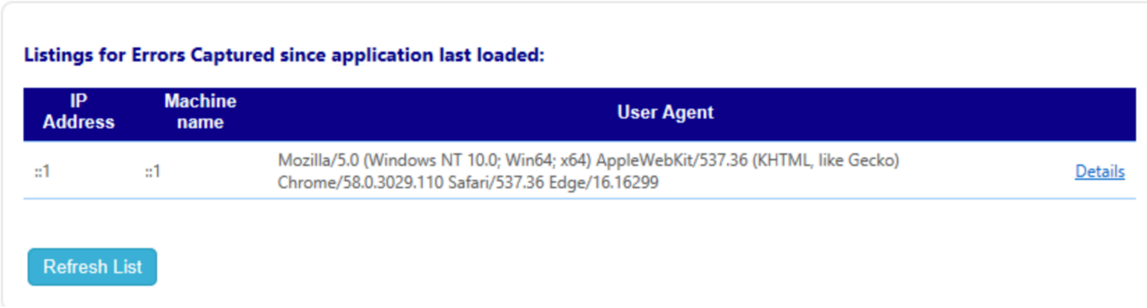
18. As before, any IDS/IPS will be great but if you don't have anything specific in mind, let's install and run Black Azure IP.
19. Stop your website project running. Hit the stop button.
20. Right-click on your *WebSite* project in solution explorer. Choose Manage NuGet Packages.
21. Find and install Azure Black IP.

22. Try to run your site. If it still runs, great. If you have a problem, open web.config and add the validateIntegratedModeConfiguration key and set it to false.

```
<configuration>
    <system.webServer>
        <validation validateIntegratedModeConfiguration="false"/>
    </system.webServer>
</configuration>
```

Checking out the intrusion detection

23. Run your site. Add some products to your cart.
24. Open an new tab and navigate to localhost:7777/DisplayIP.err. You should see no problems.
25. Run a vulnerability scan attack again on your site. Let it run through the scan/attack.
26. Look again at localhost:7777/DisplayIP.err. Now you should see some problems.



Listings for Errors Captured since application last loaded:

IP Address	Machine name	User Agent	
::1	::1	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 Edge/16.16299	Details

(Note: If the uniscan attack is taking too long, you can manually attack your site and see the results. Go to the product search page and search for a product called "<script>alert('foo')</script>". It will throw an error and log the XSS attack attempt.)

When you're able to run an attack and able to detect that attack, you can be finished.