# *Web Programming IV (420-H40-HR)*

# *Assignment 3*

Date assigned: Thursday, March 31

Date due: Thursday, April 21 (completed, see below for interim milestones)

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Securing web servers and server environment
* Research, Analyze and present Man In the Middle Attacks

To do:

There is a group component to this assignment. The groups are as follows:

|  |  |  |
| --- | --- | --- |
| **Group 1** | **Group 2** | **Group 3** |
| Marc | Ruitao | Brahim |
| Charles-Etienne | Xavier | Michael |
| Justin | Jeremy | Kristina |
|  | Ayon |  |

Part 1 – Research and Analyze – MITM (Group)

Research Man in the Middle attacks. It’s not enough to show the man in the middle, but explain how the middle-man got there.

As a team

1. Pick a MITM attack topics. You may research additional MITM variants (get approval from your professor). The initial list to pick from (you may add more once approved by your professor), i.e. pick 1 attack:

|  |
| --- |
| DNS Spoofing |
| ARP Spoofing |
| Browser Proxy, Proxy trojans, Man in the Browser |
| Public, poorly secured WIFI |
| HTTPS spoofing |
| SSL Hijacking, SSL Stripping |

1. For your variant(s). You are to:
   1. Create a power point presentation showing:
      1. Title/Cover page
      2. Interception mechanism (how the “man” gets inserted into the middle). Make this relatable to the audience (“what would you do?”)
      3. Diagram – Attack
      4. Diagram – Defense
      5. Role play – demonstrate Attack and one possible Defense
      6. Examples/cases of this in real life.
      7. Pause and ask for questions from audience
      8. List of references, further reading, sources
      9. Credits page (just like the movies)
   2. Present the presentation back to the class. It should be a maximum of 10 minutes long.
   3. Role play, act out the attack for the class. Overacting and being in character is encouraged. Use of costumes, props, puppets or stuffed animals are also encouraged.

Part 2 – Securing your site (Individual)

1. Host your solution to Assignment 2 on your local web server (not CSDEV or a shared web server, but one that you and only you control).
2. Ensure that you have implemented intrusion logging and session timeouts in your PHP application. Hint [here](https://stackoverflow.com/questions/3068744/php-session-timeout/3068766#3068766). Sessions should timeout if idle for 10 minutes and redirect back to the login page. Provide a list of intrusions that your code generated from actual testing and give a brief explanation of what triggers it:

|  |  |
| --- | --- |
| **Captured Intrusion Logs** | **Explanation** |
| Apr 17 2022 05:54:54pm – ANTICSRF: Tried login but other client does not have the right token (Removed because I don’t want it for testing zap) | When the form generates an ANTICSRF token for the login form and someone does not have the same token set. |
| Apr 17 2022 06:05:55pm - ANTICSRF: Tried either posting or editing player as dum but does not have the right token | When the form generates an ANTICSRF token for the edit/add form and someone does not have the same token set. |
| Apr 17 2022 06:16:30pm - ANTICSRF: Tried deleting image of dum but does not have the right token | When the form generates an ANTICSRF token and you someone does not have the same token set |
| Apr 17 2022 06:37:58pm - Player made an error in input field(s) when trying to log in. | When the Person is in the login menu and makes an error in the input fields |
| Apr 17 2022 06:56:17pm - The old password for dummy@dum.com by dum was not entered correctly | When the person (dum) wants to change the password of someone (dummy@dum.com)but fails entering the wrong credential |

1. Are Man in the Middle attacks are still possible in your deployed setup? If so, explain why and how to defend against it.

Yes, they can still exist, if it were public that is. Because we are using IIS express, we are only sending requests to our own server so it’s not like we have to check for attacks like those. If it were public on the other hand, It would be very hard to check to check for proxies because it is impossible to detect them. Defending against it on the server side is very hard because you can’t really check if a proxy is there. On the other hand, on the client side you can do different things. You can refuse to login to a free wifi hotspot so someone does not inject between you and your server, put a vpn so that every request is encrypted and your location is protected.

1. What configuration was set to your web application server to make it more secure. Explain how each of these items make it more secure.

|  |  |  |
| --- | --- | --- |
| **Configuration Item** | **Setting** | **Rationalization** |
| Request filtering | Get and post verb allowed only | Allowing only the get and post verbs so that the server does not run into a verb that it does no recognize. |
| Content-Security-Policy | default-src'self'; frame-ancestors 'none'; form-action 'self'; | So that the default fetches from self and not other sources, frame-ancestors none so someone does not use a Iframe and set the form action to self so that the action does not come from unwanted |
| X-Content-type-options | nosniff | So that people with older browsers can’t sniff out activities of users |
| Maximum-allowed content length | 250000 | I posted a very big player  (inputs with lots of characters) and an image to check what a player with lots of information would give to my server. I won’t need much more than 250000 to keep a margin of error and in case someone posts a big image |
| Maximum URL length | 80 | Because this is an all in one form, I put my maximum url length at 80 because There’s no way that it would be bigger than that. |
| Maximum query length | 260 | Kept it as 260 so someone does not enter something too big or unexpected. I still kept it as 260 in case someone has a very long email |

1. Run OWASP ZAP against your installed web application. Include an initial data file and instructions in a README on how to run your tests. Must be repeatable.
2. Include the generated report file as part of your submission.

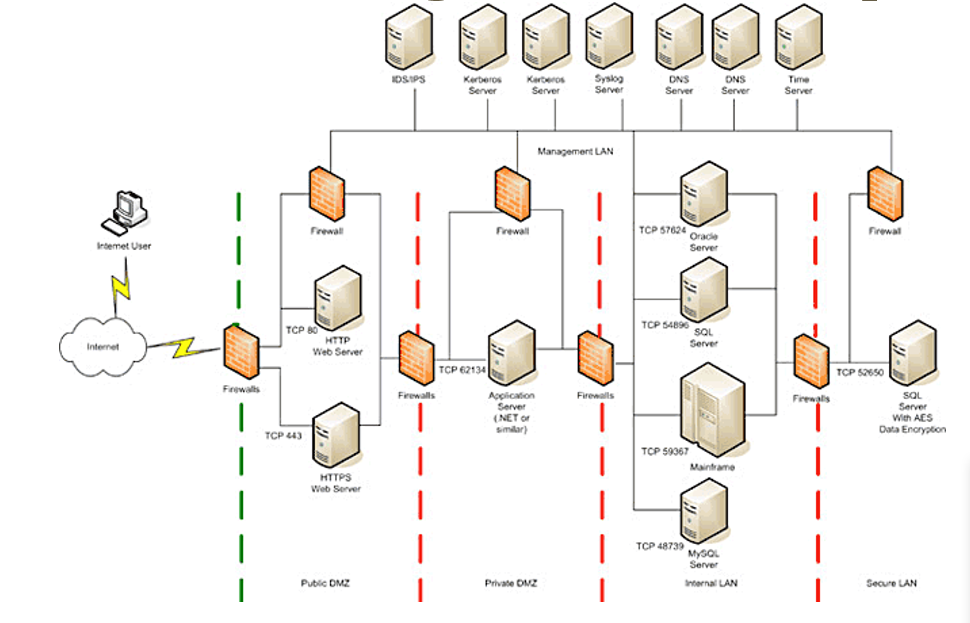
Graphical user interface, text, application, email

Description automatically generated

1. Assume your application had a database back end (not just a simple file) and your deployment followed the architecture shown in Appendix I. Describe:
   1. What web server side configuration can be done to secure your application further if you were to deploy this in a real production environment.(i.e. things you could not do because we’re in an academic environment).

Remove Directory browsing so that people don’t have access to your directories, put your server on a separate partition so someone does not try to make a directory traversal attack. I’d Also probably invest in a certificate and make it so that my website only accepts encrypted requests (https). I’d make sure to follow the least privilege methodology of things and defense in depth. Filter request so that my firewalls makes decisions on what to take as a requests. I’d also make sure that my program is secure against attacks such as sql injection so that not one is able to get to my database and just do whatever they want.

* 1. Propose the full architecture/deployment diagram. Include defense in depth and least privilege solutions and any other security concerns.



First of all, I’d make my defense in depth like this. Firewalls blocking requests that are outside the scope of what the user can do during that context prevents attacks from happening. And if something breaks, it’s not the full server that goes down. Separate the client from the server as much as possible. I’d also opt for least priviledge so that people don’t try anything shady. Don’t give freedom to people who might destroy your site

. I’d Also probably invest in a certificate and make it so that my website only accepts encrypted requests (https).

**Marking Scheme**

The key milestones are:

Declare your group topic(s) (April 1)

Hand it in (April 7)

Present to the class (April 8)

|  |  |  |
| --- | --- | --- |
| **Part 1 - MITM research and presentation** |  |  |
| Presentation |  |  |
| Interception mechanism explained |  | 6 |
| Diagrams are clear and shows how the attack works, how the defense breaks it up |  | 6 |
| Examples |  | 4 |
| References, Sources, Credits |  | 4 |
| Role Play |  |  |
| Story is clear, content is understandable |  | 5 |
| Design/Creativity/Style |  | 6 |
| Execution - educational,rehearsed/smooth |  | 5 |
|  |  |  |
|  |  |  |
| **Part 2 - Securing your site** |  |  |
| A02 hosted |  |  |
| Intrusion Logs |  | 8 |
| Session timeouts & redirect to login page |  | 8 |
| MITM concerns |  | 4 |
| Available web server configuration settings |  | 8 |
| Final OWASP ZAP report on full site |  | 10 |
| Full secure proposal |  |  |
| Defense in Depth, Least Privilege, Other |  | 10 |
|  |  |  |
|  |  |  |
| Organization - Handed in properly, formatting, code standards |  | 5 |

**To submit**

When you have completed the assignment, zip the files (Source code for your project with code updates, this document, and powerpoint for your group presentation) as YourUserName\_H40A03 and save the zip file to Moodle.

# Appendix I

**Typical 3 tier web application deployment**

