**Creating functional graph**

Penetration testing a complex web application is challenging as you have to keep close attention to small details while not forgetting the big picture. The advice is to study the whole target behavior, then split the tests in smaller tasks and take care of each one.

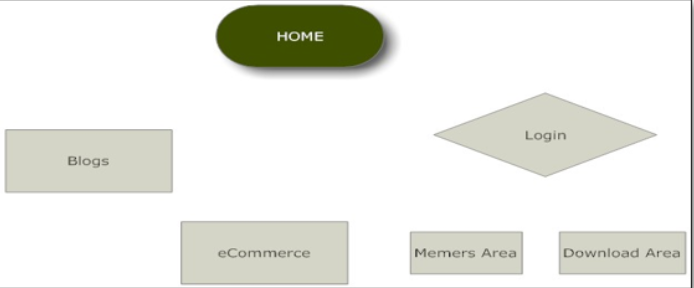
**Study the target**:

In this phase you would use the web browser to study the target under the behavioral point of view. No technical analysis is involved.

The following questions should help guide you:

* What is the purpose of the website/web application?
  + Sell online?
  + Corporate online presence?
  + Blogging?
* What seems to be the core of the website?
  + Selling product?
  + Do the sell memberships? Digital contents?
* Does it require a login to perform certain actions?
* What are the main areas of the website?
  + Blogs.
  + eCommerce area.

The answering to the above questions will help you illustrate the website blocks on paper, as the following example:



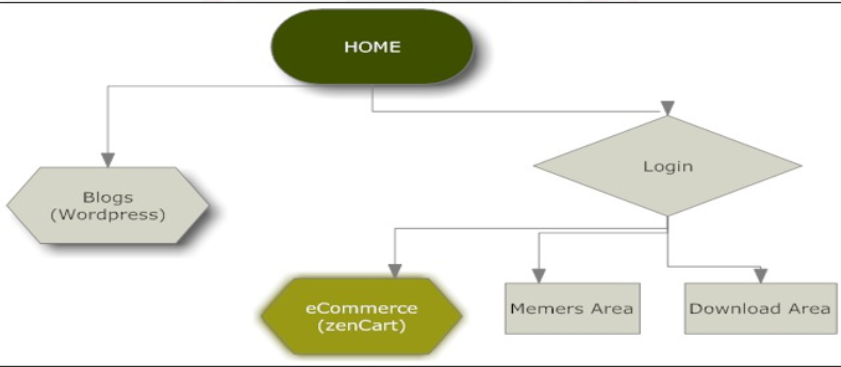
**Study the blocks:**

We will repeat our process for each block more closely.

We want to know if:

* any block uses a third-party application (we will change the shape of block if so).
* Any block can only be accessed through the login (we will create a first path using arrow).

In the following example: Zencart is third-party application and at the same time is the core of the whole website.



**Functional Graph:**

The goal of the functional graph is to visually spot important information at a glance. We will use this functional graph as a basis for further charting of our information and prepare it for the testing part.

**Mapping the attack surface:**

* Client-side validation.
  + Recognizing where the validation occurs will allow us to manipulate the input in order to mount our input.
  + Attacks: SQL injection, Cross-Site scripting or general logic flow.
* Database integration.
  + In this phase, allow us to look for SQL injection.
* File upload and download.
  + If not handled correctly, can lead to number of nasty attack including Remote and Local File Inclusion.
* Display of user supplied data.
  + Lead to XSS.
* Redirections.
  + The difference between the two is not important for our analysis. Remember 3xx code stands for redirect. From the client perspective, the redirection is handled by web browser, it recognizes the 3xx HTTP response code and makes a request to the pages contained in the location header.
* Access control and login protected pages.
  + Login pages will reveal the presence of restricted access areas of the site, we will employ authentication bypass techniques as well as password brute forcing to test the security of the authentication routines in place.
* Error messages.
  + While at this stage, we will not intentionally cause the application generate errors (how it can be a great source of information),

During the process of mapping the attack surface, we have two charting techniques:

1. Tree-based chart is especially good if there are just a few blocks. Its value is in visually spotting information.
2. Table-based chart is what we can actually use in our testing phase, where a test for a given vulnerability will be triggered by a V in the table.