DAY 3 : Web Security Owasp

—Identify Application Entry Points :

Testing the HTTP methods :

1) Requests :

- ✓ Identify GETs are used and where POSTs are used.
- ✓ Identify all parameters used in POST requests.
- ✓ Within POST requests pay attention to hidden parameters.Hidden parameters aren't seen unless a proxy is used or HTML source code is viewed.
- ✓ Pay attention to any additional or custom type headers such as (debug: false).
- ✓ Identify all parameters in query string usually in pair format such as foo=bar. Also note that many parameters in one query string seperated by a &, \~ , : or any other special character or encoding.

2) Response :

- ✓ Identify new cookies are set (set-cookie),modified and added.
- ✓ Identify redirect status code, 400 status code particularly 403 forbidden and 500 internal server error code during normal responses(unmodified requests).
- ✓ Where interesting header is used. For ex : Server: BIG-IP indicates that the site is load balanced.

Testing for application entry points :

1)This example shows a GET request that would purchase an item from an online shopping application.

```
GET /shoppingApp/buyme.asp?CUSTOMERID=100&ITEM=z101a&PRICE=62.50&IP=x.x.x.x
HTTP/1.1

Host: x.x.x.x

Cookie: SESSIONID=Z29vZCBqb2IgcGFkYXdhIG15IHVzZXJuYW1lIGlzIGZvbyBhbmQgcGFzc3d
vcmQgaXMgYmFy
```

All the parameters of the request such as CUSTOMERID, ITEM, PRICE, IP and the Cookie which could just be encoded parameters or parameters used for session state.

2)This example shows a POST request that would log you into an application.

```
POST /example/authenticate.asp?service=login HTTP/1.1

Host: x.x.x.x

Cookie: SESSIONID=dGhpcyBpcyBhIGJhZCBhcHAgdGhhdCBzZXRzIHByZWRpY3RhYmxlIGNvb2t
pZXMgYW5kIG1pbmUgaXMgMTIzNA==;CustomCookie=00my00trusted00ip00is00x.x.x.x00

user=admin&pass=pass123&debug=true&fromtrustIP=true
```

It can be noted that the parameters are sent in several locations:

1. In the query string: `service`
2. In the Cookie header: `SESSIONID`, `CustomCookie`
3. In the request body: `user`, `pass`, `debug`, `fromtrustIP`

—Map Execution Path Through Application :

Test Objective :

Map the application and understand principal workflows.

How To Test :

There are several ways for testing and measurement of code leaverge :

1)Path :

Test each of the path Through an application that include combinatorial and boundary value analysis testing for each decision path.

2)Data Flow (Taint analysis) :

Test the assignment of variable via external interaction. Focuses on mapping the flow,Transformation and use of data throughout an application.

Automatic spidering :

Automatic spidering is a tool to find new resources (URL) on a specific site automatically. ZAP offers a lot of spidering options.