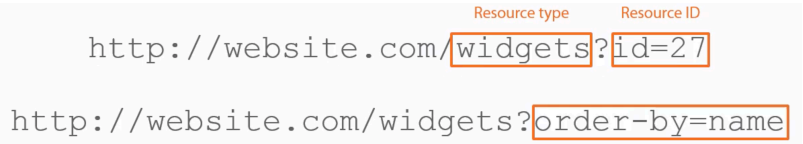
Module - 13
Hacking Web Application

- Presentation (Browser)
- Logic (Web Server)
- Data (database)

 # Query String :



 # Routing :

 # HTTP Verbs :

```
GET http://website.com/widgets/27

POST http://website.com/widgets

PUT http://website.com/widgets/27

DELETE http://website.com/widgets/27
```

 # Protections Offered. By the Browsers :

- Phishing sites
- Invalid SSL certificate
- Weak Cryptography
- Mixed Content
- Reflected Cross Site Scripting
- Security Headers

 # What the Browser can't Defend Against :

- Parameter Tempering (cookies, forms, headers, query Strings)
- Persistence Cross Site Scripting
- Attackers making direct HTTP requests

 # Reconnaissance / Footprinting :

1. Spidering : Crawling through the website following all the links that exists in that site and creating a map
   Netsparker

2. Forced Browsing : Tools : Burpsuite :
   Discover the resource which may not be accessible just by following the links

3. Directory Traversal :

4. Banner Grabing with Wget :
   comand : wget —server-response [Website]

5. Server Fingerprinting with Nmap :
   nmap -T4 -O -A -v [Website]

6. Development Artifact with Acunetix : Similar tool as netsparker , Takes the URL and goes to find as many risks as possible.

7. Automatically Generated Processes : API are being used, Resources are requested over HTTP that process input and return result., WSDL (Web Service Definition Language) It explains the method that a service implements

8. Discovering of Framework Risks : Like Jenkins , Drupal

9. Identifying Vulnerable Target : use Shodan

 # Tampering of Untrusted Data :

 # Understanding Untrusted Data :

Something that comes from an externality, Something that is outside the control of the system

- Request Body
- Request Header
- Query String
- URL (Route)
- HTTP Verbs
- External Services

 # Parameter Tampering :

 # Hidden Field Tampering :

 # Mass Assignment Attack :

Mass assignment vulnerabilites occur when a user is able to initialize or overwrite server-side variables for which are not intended by the application

Add the “isAdmin=true” after your name and password field

 # Cookie Poisioning :

Change the data in cookies

 # Insecure Direct Object Reference

http://xyz.com/shoes/name/1

now you change the ending parameter 1 to 2 and you get the details of the number 2 then this is IDOR

 # Defending Against Tampering :

1. Assume Malicious Intent
2. Verify on the server
3. Client Persistence is Dangerous
4. Whitelisting Allowable Behavious

 # Attack Involving the Client :

1. Reflected Cross Site Scripting :
2. Stored Cross Site Scripting (Persistence)
3. DOM Based XSS

 # Defending Against XSS

- Always Validate Untrusted Data
- Encoded for the right context
- Flag cookies as HTTPOnly

 # Insufficient Transport Layer Security :

Insufficient Transport Layer Protection is a security weakness caused by applications not taking any measures to **protect** network traffic. ... Exposed data and session IDs can be intercepted, which means the application is vulnerable to exploit

 # Cross Site Request Forgery (CSRF) :

Forces an end user to execute unwanted actions on an app they're already authenticated on

 # Attacks against Identity Management and Access Controls :

1. Bad Identity Management : (this password belong to this username :))
2. Identity Enumeration : Invalid account : Bad way , We have sent the mail to XYZ@ABC.com this the correct way
3. “Remember me” Feature : when you click on the remember me the vulnerable website will store the username and password in session cookie.
4. Missing Functional Level Access Controls :
5. Insufficient Access Controls : Accessing other account
6. Privilege Escalation : can be done by parameter tampering

 # Denial of Service Attacks :

A malicious attempt to make a network resource unavailable

 # Types of Denial of Service :

1. Targeted at an Individual
- Attempts to disrupt a single user
- Specially targeted based on the individual Profile
- Exploit feature such as password reset and login
- May also include DDoS

2. Targeted at an entire Service
- Attempt to disrupt the entire service
- Results in an impact to all users
- frequently executed via DDoS
- May leverage a service to do so

 # Exploiting Password Reset : repeating the password reset feature to hang the user

Countermeasure : Never Lock Out Immediately
Always email a reset link

 # Exploiting Account Lockout : “you have entered the wrong password for 5times and your account has been locked for 1 hr”

Less Invasive Defense :

Slow the rate at which login attempts can occur
Allow the user to “log in” via email
Verify the Identity via an independent Channel
- SMS
- Phone

 # DDoS Attacks :

1. Crowd Sourcing a DDoS Attacks :
LOIC : Low Orbit Ion Cannon

2. DDoS as a Service :
3. Features at Risk of a DDoS Attacks :
- Login, registration or change password facilities where a slow hashing algorithm is used
- Any Process that directly send email via SMTP (i.e. password Reset)
- Resources with database connection (potential connection pool DDoS)
- Other high-load feature
  - Running a report
  - Searching a large data set

 # Other DDoS Attacks :

- Amplification Attack
- DNS reflection attack
- NTP Attack
- SNMP Attack
- SYN Flood Attack

 # Defending against DDoS Attacks :

- Dedicated infrastructure perimeter defenses
- DDoS mitigation as a service
- Blocking incoming sources of traffic
- “Beat it with bandwidth”
- DDoS Diversion techinique

 # Other Attacks on the Server :

 # Security Misconfiguration :

- Improper Handling of errors / Exceptions
- from this we can get the information about the website, version, framework

Conversion failed when converting the nvarchar value 'passw0rd' to data type int.

 # Understanding Salted Hashes :

Registration
User enter the password in clear text
Now the server generated a salt of random bits
now the password is combined with salt and then hash it … now this hash is stored in database
Rainbow tables are used to crack salted hashes

 # Insecure cryptographic Storage

 # Unvalidate Redirects and Forwards :

It exploiting the trust that individual have in the domain of website in order to serve malicious content

Defence :

It should only be able to redirect to local resources , You should not be able to redirect to the external Website
create a whitelist on order to redirect to the external sites.

 # Exposed Exceptions logs :

Vulnerability in Web Servers :

- Usually XML or JSON response instead of HTML
- Usually Bearer tokens and / or auth headers instead of cookies
- Increasingly semantic use of HTTP Verb

They are Also Slightly difference Risks

- They are less visible so often less well tested
- They Often depend on custom clients for certification validation
- They are frequently dangerous assumptions made when the developer also build the client…