* Some common vulnerability.

1. Testing for default credentials.
2. Testing for default content
3. Bug bounty of web app using dangerous http methods.
4. Testing for proxy functionality.
5. Testing for virtual hosting misconfiguration.
6. Testing for web server software bugs.
7. Testing for web application firewall.

* Testing for default credentials.

- there is a web application using some open saurce or comercial software installed on server.
- server admin just forgot to remove predefined authentication credentials they have not change it.
- let's asume I have cisco router and I have not change default credentials of that router and I connected admin interface with web application.

* Testing for default content

Where to look for it?

1. phpinfo.php (Critical files)
2. debug method
3. Sample functionality desined to demonstrate certain common Task.
4. powerfull function left accessible
5. server manuals (documents) license that may lead some other attack through that information.

* Bug bounty of web app using dangerous http methods.

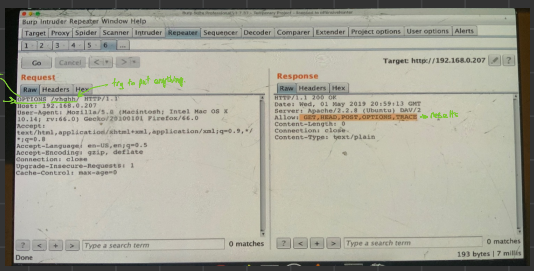
Options: used to request available methods on a server.

Get : retriere the information that is requested
Head : retrieve only header information.
Post : send request with entity enclosed in data
put : store the enclosed entity on a server.
delete : remove the resource from server.
Trace : diagnose the communication.
Connect : to create tunnel with a proxy.

* put, delete and connect methods are not required on most of the servers. It is dangerous to have those methods enable on an application because this can significantly impact their security.

* Many ways to detect HTTP methods:

- using option methods.



- or using Curl

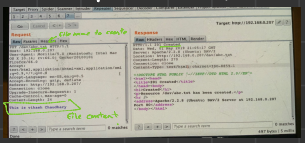
↳ >curl -X OPTIONS -i 'Target'

* exploitation of dangerous methods:

① see what http methods are available by make options request.

② put method → Create/upload files on server.

③ delete methode → delete files on server.



Use Restclient addons to see header response.

* Application server as proxy

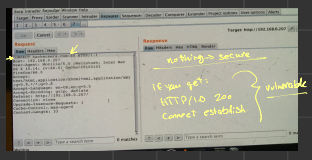
iF Application server as proxy, you can performe various attacks.

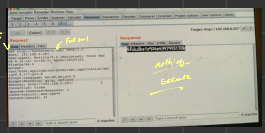
Attacker may use the server to attack third party systems on internet

Two main tiknik to find this vulnerability.

① Using GET method by giving full url in a http request body. ↘

② use CONNECT method





* Web Server Software bugs:

① scan with nmap >nmap -sV ip

② find service vulnerabilities (searchsploit, exploit-db, cve...)

③ use Nessus also for scan