Acceptance clause: by completing this project the student accepts that he/she is responsible for what he/she will do and that he/she will **not use** these tools outside the virtual one created for that purpose. The student also accepts the legal repercussions that some of these actions may be subject to if these actions are carried out on external sites.

Project 6: Brute Force Attack

Level: 5

Difficulty: high

Environment: Linux Machines (VM)

Objective: Install and analyze the indicated tools to create a FBA on worker1 (DO NOT USE outside the virtual network).

Task: The student will have to collect information about FBA tools, and configure Router to execute a FBA at worker1. After it, the user will protect worker1 using iptables or mitigating FBA with some specific tools (p.e. Fail2ban).

Necessary: Start with project 4 to learn about firewalls.

In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. This is known as an exhaustive key search.
A brute-force attack is a cryptanalytic attack that can, in theory, be used to attempt to decrypt any encrypted data (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier.
When password-guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because a brute-force search takes too long. Longer passwords, passphrases and keys have more possible values, making them exponentially more difficult to crack than shorter ones.
Brute-force attacks can be made less effective by obfuscating the data to be encoded making it more difficult for an attacker to recognize when the code has been cracked or by making the attacker do more work to test each guess. One of the measures of the strength of an encryption system is how long it would theoretically take an attacker to

mount a successful brute-force attack against it. Brute-force attacks are an application of brute-force search, the general problem-solving technique of enumerating all candidates and checking each one. More info: https://en.wikipedia.org/wiki/Brute-force_attack

Tools to be used:

Medusa: http://foofus.net/?page_id=51

Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer. The goal is to support as many services which allow remote authentication as possible. This software need common passwd that can be obtained from https://wiki.skullsecurity.org/Passwords

Hydra: https://wiki.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)

Hydra is a classic, fast network logon cracker. It is commonly used as a network logon cracker. The tool is great since it's both fast and have built-in support for many different protocols. See hydra-gtk also.

Dirb: https://wiki.owasp.org/index.php/Category:OWASP_DirBuster_Project

DIRB - URL Bruteforcer: DIRB is a Web Content Scanner. It looks for hidden Web Objects. It basically works by launching a dictionary based attack against a web server and analizing the response. DIRB main purpose is to help in web application auditing.

Fail2ban: https://www.fail2ban.org/wiki/index.php/Main_Page

Fail2ban scans log files (e.g. /var/log/apache/error_log) and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc. Generally Fail2Ban is then used to update firewall rules to reject the IP addresses for a specified amount of time, although any arbitrary other **action** (e.g. sending an email) could also be configured. Out of the box Fail2Ban comes with **filters** for various services (apache, courier, ssh, etc).
Fail2Ban is able to reduce the rate of incorrect authentications attempts however it cannot eliminate the risk that weak authentication presents. Configure services to use only two factor or public/private authentication mechanisms if you really want to protect services.

psad (http://www.cipherdyne.org/psad/): PSAD is a collection of four lightweight system daemons designed to work with iptables to detect port scans. It features are: a set of highly configurable danger thresholds (with sensible defaults provided); verbose alert messages that include the source, destination, scanned port range, beginning and end times, TCP flags, and corresponding Nmap options; reverse DNS information; alerts via email; automatic blocking of offending IP addresses via dynamic firewall configuration. When combined with fwsnort and the iptables string match extension, PSAD is capable of detecting many attacks described in the Snort rule set that involve application layer data. Can be used with hydra to perform the attack.

Disclaimer: All materials, links, images, formats, protocols and information used in this document are property of their respective authors, and are shown for educational and non-profit purposes, except those that are assigned under licenses for use or free distribution and/or published for this purpose. (Articles 32-37 of Law 2/2019, Spain)

Any actions and or activities related to the code provided is solely your responsibility. The misuse of the information in this document can result in criminal charges brought against the persons in question. The authors will not be held responsible in the event any criminal charges be brought against any individuals misusing the information in this document/tools to break the law.

The examples, tools, tests or any other activity shown or suggested in this document should NEVER be carried out outside the private network created for this purpose, since criminal and/or punishable responsibility may be incurred.