

**Brittany Walker**

Old school web journal. Sisyphean pursuit of skill.

## Advent of Cyber – Brute Forcing Passwords with Hydra



Our story continues. At AntarctiCrafts, Detective Frost-eau found that critical systems had been locked down and the password to the IT server room door had been changed. Remember our exploit in Day One?




I've never spent much time on brute-force software before, but I'm excited to start! Hydra's real strength is that it's *parallelized*, meaning that it supports multiple connections simultaneously. This reduces the amount of time needed to crack a password. Makes sense. If you're limited to

one guess at a time, that'll take way longer than making multiple guesses simultaneously. Hydra works online and supports more than 50 network protocols. Impressive.

We need to get to the backup tapes and get business back online. We have to get gifts delivered by Christmas! To get past the new password, we're going to learn about

1. Factors in password complexity.
2. Generating password combos with crunch.
3. Trying those combos with Hydra.

Password complexity is calculated with a simple formula. The base number (possible characters) raised to the power of the password's length. For example, if the possible characters in a PIN number are the 10 standard digits (0-9) and it's a 4 digit PIN, the number of possible combos is  $10^4$ , which evaluates to 10,000. For a password pulling out some stops (10 digits, 26 uppercase and 26 lowercase English letters) even a 4-character password would be one of  $62^4$ , or 14,776,336 possible passwords.

Password Length	Allowed Characters	Number of Possible Passwords	
4	Uppercase, lowercase, and digits	14,776,336	
6	Uppercase, lowercase, and digits	56,800,235,584	
8	Uppercase, lowercase, and digits	218,340,105,584,896	
10	Uppercase, lowercase, and digits	839,299,365,868,340,224	
12	Uppercase, lowercase, and digits	3,226,266,762,397,899,821,056	
14	Uppercase, lowercase, and digits	12,401,769,434,657,526,912,139,264	
16	Uppercase, lowercase, and digits	47,672,401,706,823,533,450,263,330,816	

14-odd million sounds like a lot, but as the challenge points out, computers are *fast*. If you can try 1,000 passwords a second, you'll have the right password in 4 hours if you have to try every possible combination. It's also likely that you won't have to – it may take only 2 hours. Of course, many systems will lock you out after a few missed guesses, and few will let you make

attempts that fast. If you managed to find a password hash to compare against, though, you don't need to access the site directly. You can run your tests locally as fast as you like.

Most password cracking software allows you to set time between attempts for precisely those reasons. Slows you down, but you'll get there eventually. The only way to decrease the likelihood of having passwords brute forced is by increasing complexity. This can be done by specifying a minimum length, character variety, and so on.

Let's see if our intrepid AntarctiCrafts hacker had that kind of foresight!

### Step One – Let's Generate a Password List!

Fortunately for us, the PIN pad on the IT server room door only has 16 characters – the hexadecimal digits. We'll prepare a list of passcodes using crunch, a utility made to generate wordlists for passing to your password cracker. Say that five times fast.

Crunch is pre-installed on the Attack Box, so we're going to split our screen here and get to work. I found [this](#) incredible answer from Kamil Maciorowski on how crunch accepts input, but for brevity, these are the basics:

```
crunch <min-len> <max-len> [<charset string>] [options]
```

[...]

**charset string**

You may specify character sets for crunch to use on the command line or if you leave it blank **crunch** will use the default character sets. The order **MUST BE** lower case characters, upper case characters, numbers, and then symbols. If you don't follow this order you will not get the results you want. You **MUST** specify either values for the character type or a plus sign. [...]

The command provided below says that we want a minimum length and maximum length of 3 – essentially, we only want three-character passwords. We copy the character set from the PIN pad exactly, and we save the output (using the -o flag) to 3digits.txt

```
crunch 3 3 0123456789ABCDEF -o 3digits.txt
```

### Step Two: Using the List

Automation is where it's at! Can you imagine manually entering 4,096 different combos by hand?

That's where Hydra comes in. To use it, we need to understand how passwords are checked by the web application. We do this by viewing the source HTML of the site we're interested in. In this case, we use the POST method from HTTP and send the PIN to [machine IP]:8000/login.php with the name *pin*.

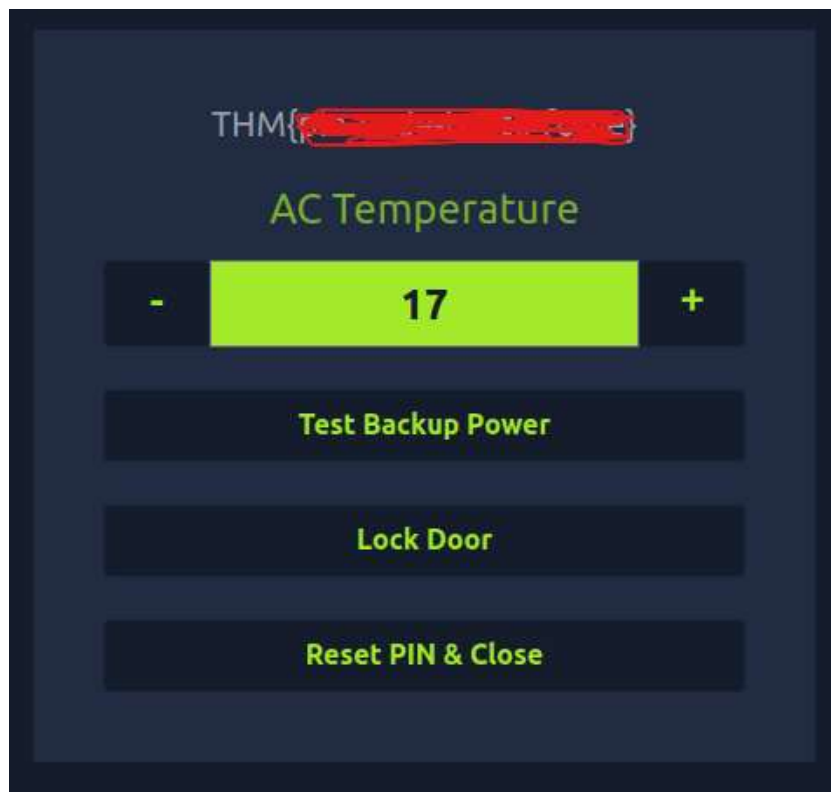
Gorgeous. To brute force this site, we'll use the provided command.

**hydra =l " -P 3digits.txt -f -v [machine IP] http-post-form "/login.php:pin=^PASS^:Access denied" -s 8000**

(all one line, of course.)

- **-l** indicates no login username – we just need a pin here.
- **-P 3digits.txt** specifies the name of the password list file we're using.
- **-f** stops Hydra when a working password is found. We just need one!
- **-v** means verbose. It means more output, and can help flag issues.
- **[machine IP]** here will be different for everyone, since we're using the VM in TryHackMe. Replace this with the IP address provided when you start the machine!
- **"/login.php:pin=^PASS^:Access denied"** is actually a three-part statement separated by the colon (:) – the first part is the page to submit to, the ^PASS^ command is replaced by each password guess in turn with the name *pin*, and "Access denied" tells Hydra what to expect as a result if the attempted password is incorrect.
- **-s 8000** indicates the target port number.

Once we run the command, it only takes a few short minutes to crack the control system and capture our flag. Using the Firefox browser inside the AttackBox, we navigate to [machine IP]:8000/pin.php and enter our PIN.



Click that unlock button to get your flag, and voila.

Day 4 promises to bring more brute-forcing fun – I'll see you on the next writeup!

Have you used Hydra or crunch before? John the Ripper is also on my must-try-soon list. If you have your own password-cracking software favorites, drop them in the comments below.

#### Advertisements

Occasionally, some of your visitors may see an advertisement here, as well as a [Privacy & Cookies banner](#) at the bottom of the page. You can hide ads completely by upgrading to one of our paid plans.

UPGRADE NOW

DISMISS MESSAGE

## Sponsored Content