# Brute Force Attack

A lot of the naming conventions in technology are less than ideal. It’s not like there’s a committee that approves or rejects terminology. Usually, someone coins a phrase and it sticks, like a nickname. Take for example the *brute force attack*. It would be more suitably called *the long way home*. There’s nothing fancy about a brute force attack. It’s simply about taking the time, nothing rushed.

The perpetrator of the brute force attack sets up a process by which every possible variable is attempted. Let me give you a physical metaphor. Imagine for some reason a bank allowed you access to a safety deposit box vault, but you were not a bank customer, and further imagine that you were overcome with the spirit of opportunity, and you started trying every safety deposit box to find one that was unlocked. Can you see the person doing that in your imagination? Maybe they start at the top row and go left to right, then drop down to the next row and go right to left, pulling the small handle to every box door. Tedious isn’t it? That’s the basic definition of a brute force attack, except, like all things in technology, it takes place in an invisible digital space that our physical eyes can’t see. We only know it’s taking place through reporting programs — if we have them — or after the fact, when the system has been breached.

Why is it the long way home? Because programs are set up to continually attack the login site with every possible combination. A brute force attack can go on for an indeterminate period. There’s little or no sophistry to it, like kicking or ramming a door until the frame or the hinges crack and access is gained. They don’t end until the login, the open safety deposit box in my visual metaphor, is found. Even *found* is a tad gratuitous of a word; it’s not so much that the information is *found*, more like the access is *revealed* after countless possible variations of the allowable password and login are tried, way past the tolerance of any human effort, which is why programs are set up to do the repetitive task of trying login names and passwords.

## Motive and Method

The simple brute force attack is the most *gamed* in cyber security, meaning a lot of them are games of some sort. The people or groups doing them just do them to see if they can do them. It’s like the Joker in Christopher Nolan’s *Dark Knight* movie; when he finally gets half the mob’s money, and it’s in a big pile in a warehouse, he can think of nothing better to do than burn it. He did everything he did just to do it, to kill time, like a game. The same is true for some of the brute force attacks. Others, of course, are to gain access to a network, to get privileged information, and to use that information to criminal ends, theft, fraud, extortion, you name it.

### Hybrid Brute Force Attack

The hybrid brute force attack is a mixture of the simple brute force attack and a little bit of external logic. What is meant by *external* is acquired information. The attacker begins with a little bit of external information, maybe a dictionary of commonly used passwords in the organization, and this greatly limits the scope of the brute force attack. Another way of looking at it is a brute force attack with delimited parameters. Remember, the straight-up brute force attack relies on a relentless campaign of attempts at the username and login until it yields, beginning with no clues or insights into what those usernames and passwords might be, throwing spaghetti against the wall if you will.

### Dictionary Attack

The dictionary attack employs insider knowledge, like a company dictionary of possible strings or phrases, and this limits the guesswork involved in the usernames or passwords, in that it’s not a cold start. This can save time and, more importantly, exposure, as some systems will have failsafes when too many login attempts have been made, so advance knowledge that narrows the scope could be the difference between success and failure for the hacker. Sometimes the dictionary can contain personal information about the accounts you’re trying to get into or even previously leak passwords from data breaches.

### Rainbow Table Attack

A rainbow table is a pre-computed table for reversing cryptographic hash functions. This is a much more sophisticated brute force attack in which the hacker uses a rainbow table to guess a function up to a certain length consisting of a limited set of characters. Like the dictionary attack, it can greatly speed up the cracking process by avoiding the need to compute the hash function for each guess.

### Reverse Brute Force Attack

The reverse brute force attack goes after a network to which the attacker has data. It’s similar to the dictionary attack but more specific. The hacker might have a collection of old passwords for the organization, but, given that many people reuse passwords, it’s a starting point that greatly limits the brute force attack. Reusing passwords is the single greatest vulnerability in most systems.

## Tools of the Trade

Hackers come equipped with highly specialized tools that have been designed for their trade. Now, in fairness, most of the top-tier tools were created for cybersecurity professionals, but they’ve been repurposed by the bad guys. It’s like a cop losing his gun and that weapon being used in a holdup. One of the best tools is John the Ripper, a free password-cracking software that can automatically detect the type of hashing used in a password. It can perform brute force attacks by combining text and numbers, or it can be used with a dictionary of passwords for dictionary attacks. Rainbow Crack generates rainbow tables to speed up the brute force attack process. Ncrack supports an amazing array of protocols, including RDP, SSH, HTTP(S), SMB, POP3(S), VNC, FTP, and Telnet, and it’s a common tool for brute force attacks. It’s compatible with all major operating systems, including Linux, BSD, Windows, and Mac OS X. These tools are easy to get and fairly easy to learn. The brute force attack is not a high-level attack. It can be executed by almost anyone who puts a little bit of time into learning the tools of the trade.

## Protection

The beginning of defense against a brute force attack is simply knowing that they happen, and, more importantly, how they happen and why they happen. We’ve covered that already. Now let’s get down to a war plan. Here's what Caesar would come up with in his tent if he were camped on the outskirts of a digital battle.

# worksheet

To protect against brute force attacks on networks and systems, you can implement the following prevention techniques:

### Passwords

Don’t piss around with passwords. If you’re in charge of security, or the closest thing to a security chief in your company, enforce password regulations, and enforce here means more than *suggestions* at monthly meetings or weekly team scrums. You want to take a look at the passwords being used. Make sure they live up to the standards you’ve set. There’s no other way. Unfortunately, you can’t take people at their word. Choose a set of protocols for establishing a password. They should include the length of the password, and the mixture of letters, symbols, and numbers. Enforce password changes at a cyclic frequency. Quarterly is too little. Weekly is too much. You’ll drive the team crazy. Six to eight weeks is a good space of time to recycle passwords. It may seem excessive, but when you know how much effort is out there to break your network, it’s about right.

### Attempts

You have to limit the number of login attempts. If you leave it open-ended, it’s just a red carpet inviting brute force attacks to walk into your organization, and sooner or later they will succeed, because they have unlimited time. The limit on attempts will lock out nefarious actors. Anyone who has to try their username or password more than once is either incapacitated or they’re not who they say they are. If there’s any other reason and I’m being a little too hard on team members who might forget their login credentials, remember they can call the network administrator, or walk over to his or her office, explain the situation, and ask for a password reset.

### Two-Factor Authentication

Two-factor authentication doubles the number of systems that a hacker has to get through. Even more powerful is how it sequentially concatenates them in a way that makes the hacker's mission almost impossible. The process is very simple. Mandate that users provide a second form of verification. This can be a code sent to their smartphones right after they enter their passwords. Set a time limit on the input of that code and you're all set. Even if the brute force attack works, the hacker is not going to be able to receive the code. They don’t own the smartphone. As a bonus, you can configure your system to annotate any login attempts that get past the username and password but fail to input the two-factor authentication code. Then the security team can analyze where that attempt came from and identify possible bad actors.

### hCaptcha

Use a captcha tool of which hCaptcha is the best of class. The acronym stands for *Completely Automated Public Turing test to tell Computers and Humans Apart*.

It’s an excellent tool to determine whether the user is a human or a bot. You’re usually asked to select a certain type of image amidst an array of images, for example, select all the squirrels. None of the tools in the brute force attack toolbox can do this, We can rely on the laziness of hackers. We haven't seen any examples of an ancillary system set up to ring a bell, flag an alarm, or send an email, or any ancillary system for that matter that can accommodate the request for human visual interaction, so it’s an excellent defense.