# Brute force attacks and OS hardening

In this reading, you'll learn about brute force attacks. You'll consider how vulnerabilities can be assessed using virtual machines and sandboxes, and learn ways to prevent brute force attacks using a combination of authentication measures. Implementing various OS hardening tasks can help prevent brute force attacks. An attacker can use a brute force attack to gain access and compromise a network.

Usernames and passwords are among the most common and important security controls in place today. They are used and enforced on everything that stores or accesses sensitive or private information, like personal phones, computers, and restricted applications within an organization. However, a major issue with relying on login credentials as a critical line of defense is that they're vulnerable to being stolen and guessed by malicious actors.

## Brute force attacks

A **brute force attack** is a trial-and-error process of discovering private information. There are different types of brute force attacks that malicious actors use to guess passwords, including:

- *Simple brute force attacks.* When attackers try to guess a user's login credentials, it's considered a simple brute force attack. They might do this by entering any combination of usernames and passwords that they can think of until they find the one that works.
- *Dictionary attacks* use a similar technique. In dictionary attacks, attackers use a list of commonly used passwords and stolen credentials from previous breaches to access a system. These are called "dictionary" attacks because attackers originally used a list of words from the dictionary to guess the passwords, before complex password rules became a common security practice.

Using brute force to access a system can be a tedious and time consuming process, especially when it's done manually. There are a range of tools attackers use to conduct their attacks.

## Assessing vulnerabilities

Before a brute force attack or other cybersecurity incident occurs, companies can run a series of tests on their network or web applications to assess vulnerabilities. Analysts can use virtual machines and sandboxes to test suspicious files, check for vulnerabilities before an event occurs, or to simulate a cybersecurity incident.

### Virtual machines (VMs)

Virtual machines (VMs) are software versions of physical computers. VMs provide an additional layer of security for an organization because they can be used to run code in an isolated environment, preventing malicious code from affecting the rest of the computer or system. VMs can also be deleted and replaced by a pristine image after testing malware.

VMs are useful when investigating potentially infected machines or running malware in a constrained environment. Using a VM may prevent damage to your system in the event its tools are used improperly. VMs also give you the ability to revert to a previous state. However, there are still some risks involved with VMs. There's still a small risk that a malicious program can escape virtualization and access the host machine.

You can test and explore applications easily with VMs, and it's easy to switch between different VMs from your computer. This can also help in streamlining many security tasks.

### Sandbox environments

A sandbox is a type of testing environment that allows you to execute software or programs separate from your network. They are commonly used for testing patches, identifying and addressing bugs, or detecting cybersecurity vulnerabilities. Sandboxes can also be used to evaluate suspicious software, evaluate files containing malicious code, and simulate attack scenarios.

Sandboxes can be stand-alone physical computers that are not connected to a network; however, it is often more time- and cost-effective to use software or cloud-based virtual machines as sandbox environments. Note that some malware authors know how to write code to detect if the malware is executed in a VM or sandbox environment. Attackers can program their malware to behave as harmless software when run inside these types of testing environments.

## Prevention measures

Some common measures organizations use to prevent brute force attacks and similar attacks from occurring include:

- **Salting and hashing:** Hashing converts information into a unique value that can then be used to determine its integrity. It is a one-way function, meaning it is impossible to decrypt and obtain the original text. Salting adds random characters to hashed passwords. This increases the length and complexity of hash values, making them more secure.
- **Multi-factor authentication (MFA) and two-factor authentication (2FA):** MFA is a security measure which requires a user to verify their identity in two or more ways to access a system or network. This verification happens using a combination of authentication factors: a username and password, fingerprints, facial recognition, or a one-time password (OTP) sent to a phone number or email. 2FA is similar to MFA, except it uses only two forms of verification.
- **CAPTCHA and reCAPTCHA:** CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. It asks users to complete a simple test that proves they are human. This helps prevent software from trying to brute force a password. reCAPTCHA is a free CAPTCHA service from Google that helps protect websites from bots and malicious software.
- **Password policies:** Organizations use password policies to standardize good password practices throughout the business. Policies can include guidelines on how complex a password should be, how often users need to update passwords, and if there are limits to how many times a user can attempt to log in before their account is suspended.

## Key takeaways

Brute force attacks are a trial-and-error process of guessing passwords. Attacks can be launched manually or through software tools. Methods include simple brute force attacks and dictionary attacks. To protect against brute force attacks, cybersecurity analysts can use sandboxes to test suspicious files, check for vulnerabilities, or to simulate real attacks and virtual machines to conduct vulnerability tests. Some common measures to prevent brute force attacks include: hashing and salting, MFA and/or 2FA, CAPTCHA and reCAPTCHA, and password policies.