200 XP

Use authentication and authorization in your network

9 minutes

Only authenticated and authorized users or services should access network resources like servers and applications. It's your job to make sure that access is conditional on authentication and authorization.

In this unit, we'll explore how to use network authentication to authenticate users and services in your network. We'll also look at network authorization as a method to check whether a particular user or service has access to the resource.

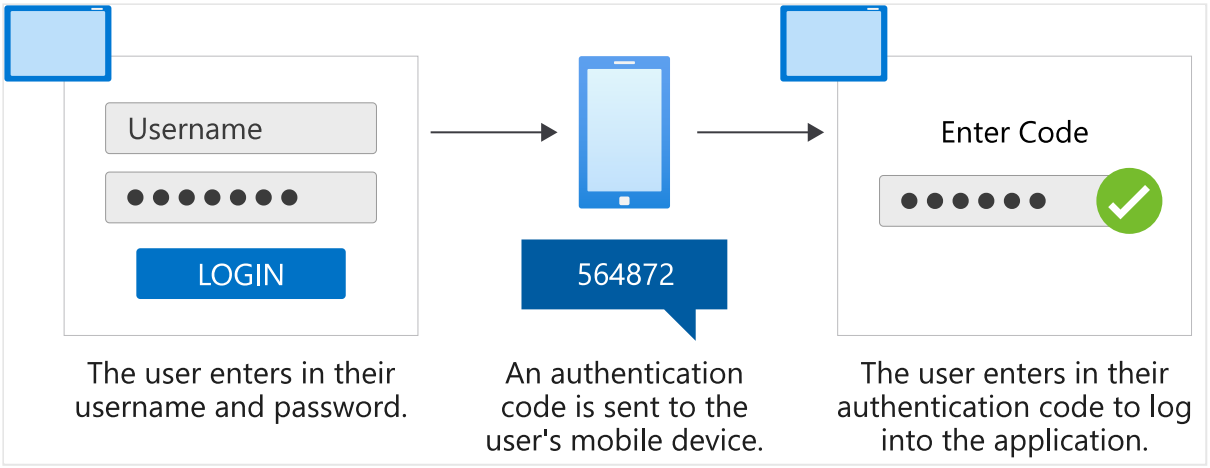
Network authentication

Configure network authentication to verify that users are who they claim to be. Authentication is how the network separates legitimate from suspicious access. Use different methods to implement network-level authentication. Let's have a look at some of these authentication types.

Password authentication

Password authentication is the most familiar form of authentication. The user enters a secret value, only known to them, to gain access to the network. Secure passwords need to meet criteria, like having lowercase and uppercase characters, along with numbers and symbols (such as `?`, `%`, or `$`). We also recommend that you make passwords as long as possible.

Two-factor authentication



Two-factor authentication is a mechanism that allows users to verify an authentication attempt. The user needs to provide a one-time code sent to their device as a confirmation of authentication. For example, they might receive a code through a text message, or a code generated through an app on their phone, like Microsoft Authenticator.

Token authentication

Token-based authentication is similar to two-factor authentication. However, instead of using a cell phone, which may become compromised, a company may choose to use a device purpose-built for authentication. The device can be a USB enabled device or a smart card the user uses for successful authentication. When using token-based authentication, the company should make sure the user returns the device if they no longer require access.

Biometric authentication

Biometric authentication uses the user's physical attributes for authentication. These attributes are uniquely human characteristics, like fingerprints, facial features, or voice. However, biometric-based authentication can be costly to implement because of the specific type of scanners required to handle this information. User privacy concerns could also pose problems.

Transactional authentication

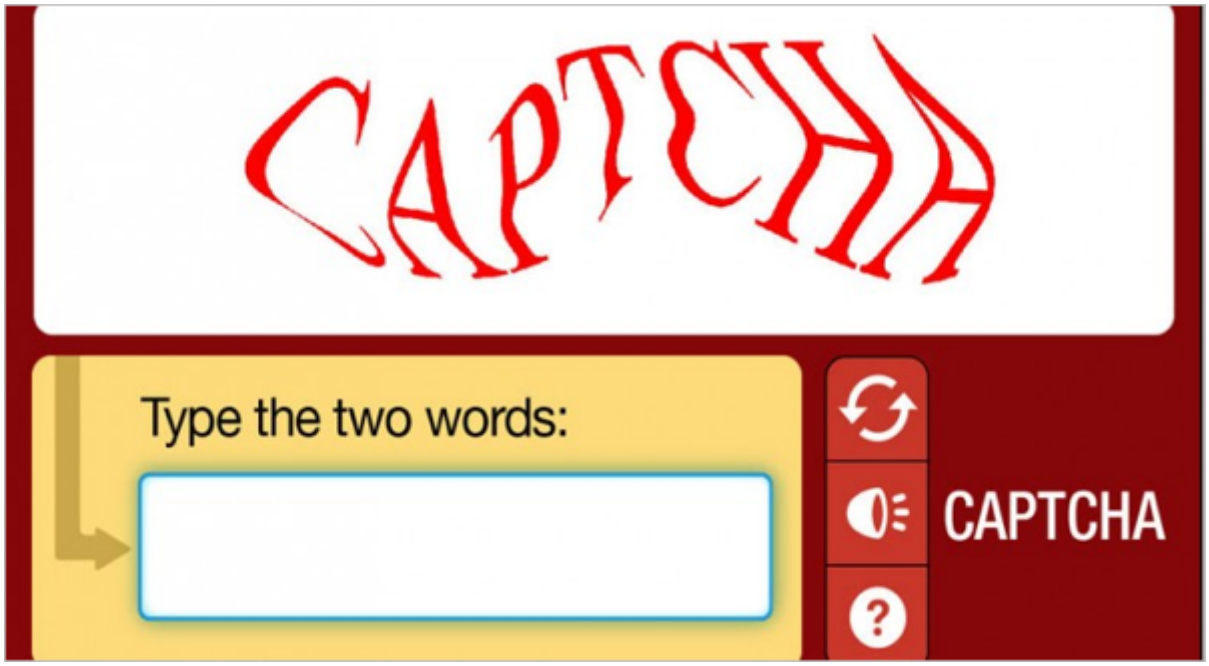
You may not always want to rely only on information provided by a user. Instead, transactional authentication lets us scrutinize the characteristics of the user. You could, for example, expect users to regularly access the network from the United States during work hours. However, if there's a sign-in from the other side of the world at midnight, then the user's account would be flagged, and the system can prompt the user for additional verification steps before authentication.

Transactional authentication gives an additional layer of protection for your network.

Computer recognition authentication

Computer recognition authentication looks at the device being used to access the network. A small piece of software is installed on the device with first-time use. This software holds a cryptographic device marker. When the user signs in, the device marker is checked to see if they're using the authorized device. Computer recognition authentication is especially useful if users are only allowed to sign in from a single device. This method could make things difficult if users regularly switch devices.

CAPTCHA



The Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is used to verify whether the entity attempting access to a system is a human.

Attackers can create applications that are capable of automating the steps to log into accounts. A CAPTCHA presents an obfuscated image of a scenario, letters, or numbers, and the user is asked to explain what they see. Compared to humans, applications have difficulty identifying distorted photos, letters, and numbers. Humans can typically make out what is shown in a distorted image.

However, keep in mind that this method might present difficulties for users with a vision-impairment.

Single sign-on

Single sign-on lets users enter their credentials once to allow authentication across multiple applications and tools. For example, a user could sign in to their mail application and

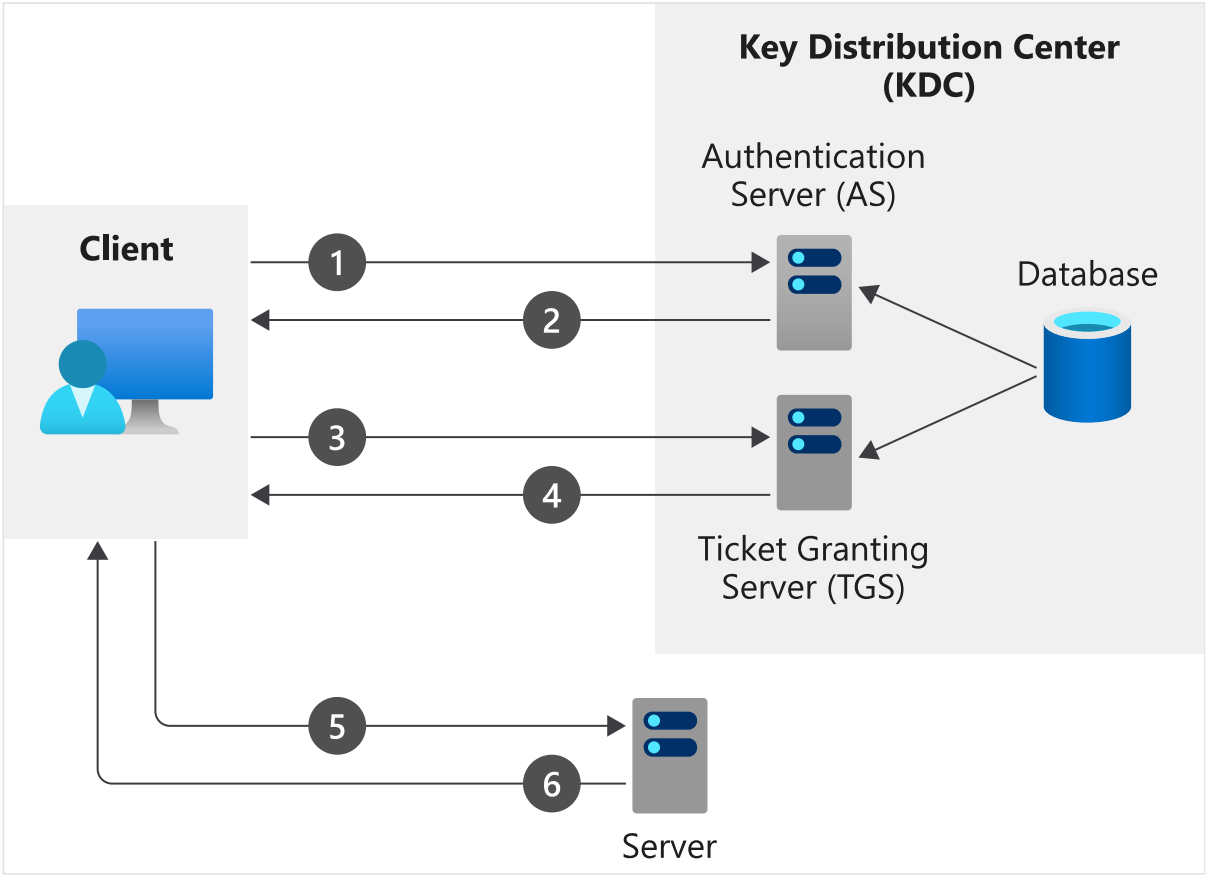
automatically become authenticated across tools they use to manage network security and storage. Single sign-on saves time for users. This method carries the risk that a single sign-on could also help an attacker gain access to several platforms, tools, and applications by successfully accessing one of them.

Authentication protocols

An authentication protocol is a shared set of rules for how information is exchanged between electronic devices. Two of the most commonly used authentication protocols are Kerberos and Transport Layer Security/Secure Sockets Layer (TLS/SSL).

Kerberos

Kerberos is an authentication protocol used across different operating systems. Windows uses Kerberos as its default authentication protocol. Linux and Mac OSs can also use Kerberos.



Kerberos authentication protocol relies on a trusted server called a Key Distribution Center (KDC). A KDC consists of a few components:

- An authentication server that authenticates and issues tickets to principals, like a user or service.
- A database that holds information about the principals and their secret keys.

- Another server that grants service tickets based on the initial tickets that the principals present.

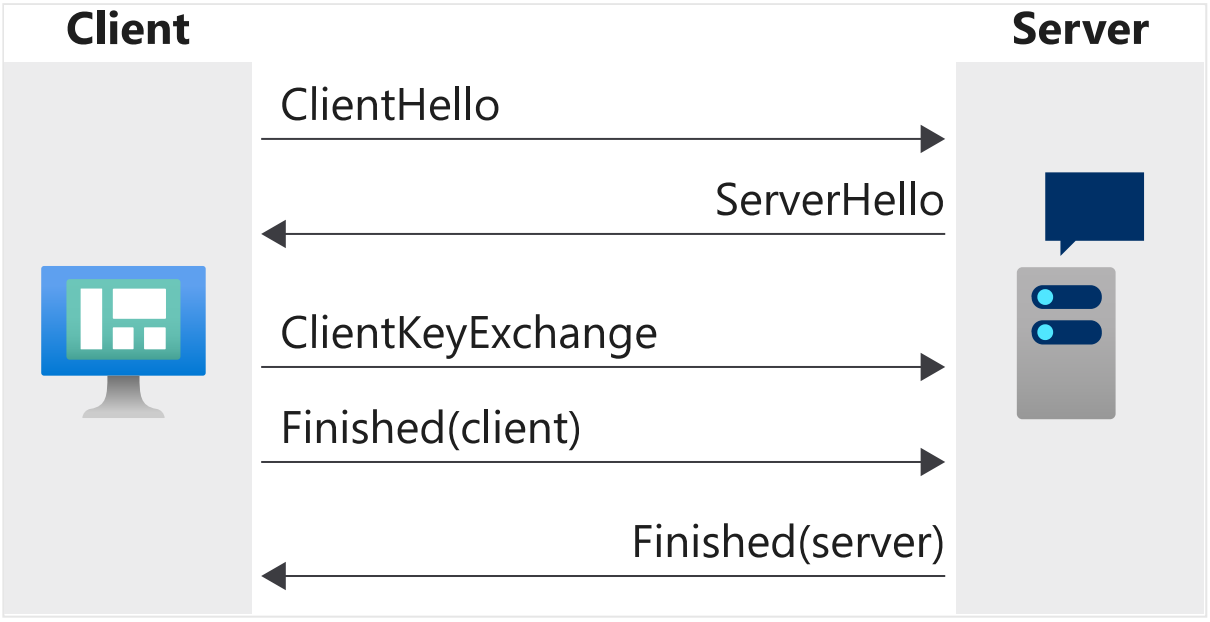
In Kerberos, principals get tickets that grant them service tickets from the KDC. They use those service tickets to access resources, services, or applications. This process remains invisible to the user.

TLS/SSL

TLS and the older SSL are both protocols for encrypting information sent over the internet. Because the data is encrypted, attackers can't view what is sent through TLS/SSL.

You'll often see a padlock on the browser when a site makes use of a secure connection. This symbol means the site is using a secure TLS/SSL session with the browser. TLS/SSL is also used for file transfers, voice-over-IP, and email.

SSL is the predecessor of TLS and is deprecated. We'll often find the two terms used interchangeably. The protocols work as follows:



1. The client sends a "ClientHello" message to the server. This message includes information like the SSL/TLS version, and the cryptographic algorithms that the client supports.

2. The server sends a "ServerHello" message back that includes the algorithm it has chosen from the list of algorithms supported by the client. The message also includes a session ID, the server's digital certificate, and its public key.

3. The client uses the digital certificate to verify the server's identity with a certificate authority, so the client can be sure it's dealing with a trusted server.

4. A client key exchange happens, where the client sends a shared key that's encrypted with the server's public key to the server.
5. The client sends a "finished" message that's encrypted with the shared key.
6. The server sends its own "finished" message that is encrypted with the shared key. From this point, the client and the server can continue to exchange messages that are encrypted with the shared encrypted key.

Network authorization

When authentication is completed successfully, we'll need to ensure the authenticated user or client is authorized to access the resources or services they're requesting. Authorization can be granular. For example, a particular database user might have permission to access and make changes to a single database. But the user couldn't access any other database because they don't have the permissions.

Permissions can include read, write, delete, and more. Use the right permissions for the right user or client. If a user or client switches roles, you can change their permissions to match the new level of access. Give each user or client the least number of permissions needed to get the job done. Never give a user or client any permissions they don't need.

Differences between authentication and authorization

Authentication	**Authorization**
Confirms whether the user or client is who they claim to be.	Confirms whether the user or client can perform an action against a resource or service.
Asks for credentials like username or password.	Checks permissions attached to the account in the background, and sometimes indicates which permission you need.
Must happen before authorization.	Happens after successful authentication.
For example, an HR member signs in to the HR app.	The HR member attempts to delete a user from the wrong department by accident. The action is denied because they don't have the right permissions for that department.

Check your knowledge

1. What role does authorization play during a sign-in event?

- (●) Authorization is the process of determining whether the authenticated user or client has access to specific resources.
  - ✔ **After the user or client is confirmed to have the right permissions, they can access the resource.**
- ( ) Authorization is the process of determining whether a particular user or client is the author of specific resources.
- ( ) Authorization is the process of determining whether a client or user is who they claim to be.

2. Which of following happens in the handshake process of SSL/TLS protocol?

- (●) The server sends a "ServerHello" message back. This message includes a session ID, the server's digital certificate, and its public key.
  - ✔ **The server and the client exchange a number of messages like this before communication can start flowing officially between the two.**
- ( ) The server sends a "ServerHello" message back. This message includes a session ID, the client's digital certificate, and a public key.
- ( ) The server sends a "ClientHello" message back. This message includes a session ID, the server's digital certificate, and its public key.

Next unit: Understand firewalls and network security

Continue >