Authorization

The process of authorization refers to the rights, permissions and privileges granted to an authenticated user. Authorization is the last step in any identity management system: Having claimed and proved an identity access right (or permissions need to be assigned), for any given resource. Authorization can be managed through controls based on: user roles, user or resource attributes, policies etc. An Identity Access Management (IAM) policy store can be used to hold the authorization policies and comparing them to the entities access request.

Authorization is where the principles of least privilege and need-to-know apply. These principles both have a common theme, although they are slightly different. Least privilege only grants a person the minimum level of access required for them to perform their job function – perhaps read-only or guest-level access. Need-to-know is often based on classification of information that only grants access to information when required or according to the user's clearance. An example of this is to only display the last four digits of a credit card number to a person working for a merchant that accepts credit card payments. This prevents theft or misuse of the credit card by the employee.

A core element of authorization is the principle of separation of duties (also known as segregation of duties). Separation of duties is based on the security practice that no one person should control an entire high-risk transaction from start to finish. Separation of duties breaks the transaction in separate parts and requires a different person to execute each part of the transaction. For example, Bill may submit an invoice for payment, but it has to be approved by Sam prior to payment; or Bill may submit a proposal for a change to a system configuration but Sam will review and need to approve the change before it can be implemented. These steps can prevent fraud or detect an error in the process before implementation.

It could be that Sam will sometimes input an invoice for payment, but he would not be able to approve the invoices he inputs. This is mutual exclusivity: Sam can perform both operations (input and approval), but not on the same invoice.

However, if Sam and Bill work together to bypass the separation of duties they could jointly commit fraud. This is called collusion.

Another implementation of separation of duties is dual control. This would apply at a bank where there are two separate combination locks on the door of the vault. Some personnel know one of the combinations and some know the other, but no one knows both combinations. Two people have to work together to open the vault, thus the vault is under dual control.

Maintenance

Identity management is not a one-time static process. Users credentials and needs will change: name changes, job functions, geographical locations etc. Therefore, ID management should be seen as a “living process” and requires constant maintenance. This maintenance includes (but is not limited to) a variety of functions including: user, group and password management etc. The creation and maintenance of identities is one of the key phases and starts when a new identity request is received (for example: from HR when a new employee joins the organization), once created the other administrative functions are addressed, such as group membership, location, temporal (timed) access and so on. These components set the users privilege levels. Some accounts will have a limited lifespan (guest and temporary users) and those lifespans will need to be assigned and monitored. Once the account lifespan expires these accounts will need to be archived in accordance with corporate policies.

Bear in mind that sooner or later identities will change, as users move job functions, promotions etc. this all falls under the maintenance phase. Build in a de-provisioning phase and again remember that archiving process as the activities carried out on old accounts may need to be investigated at some point.

The control of some accounts may need to be delegated to other departments, thus allowing the workload of managing the account identities to be spread across the organization.

All of these processes require a dedicated individual or team and in larger organizations this can represent a large overhead to an organization. An approach often adopted to reduce this overhead is self-service. The self-service approach removes the need for dedicated teams to create and manage identities (accounts) but instead puts the focus on the users themselves.

Discussion: Protecting Password

What are some ways to ensure that passwords are only reset for the correct people?

Passwords

There are many opinions about what makes a strong password. Most often a longer password is considered stronger than a shorter one, and the use of special characters, numbers and upper and lowercase letters may provide additional strength. Nevertheless, a well-publicized breach occurred where an organization allowed upper- and lowercase letters but actually converted everything to lowercase.

The other question is the length of time a password should be used before it must be changed. The main point to remember is the need to balance security with functionality: if the password rules are too stringent then users will almost certainly begin to work around the access controls, for example by writing passwords down or using the same password on multiple systems.

When a user has entered a password incorrectly multiple times then it may be advisable to lock out the UserID being accessed. The point at which the account is locked is called the threshold or clipping level. The clipping level is set to allow for normal human error, but to lock the account if it is experiencing a brute force attack.

Ownership

Another authentication factor is based on ownership. This has been in use for many decades: a person's identity could be authenticated through a letter, passport, driver's license, or badge. Today we also see ownership through the use of tokens (hardware- and software-based), smart cards, key fobs and other hardware or software mechanisms.

Tokens

Many tokens and smart cards are used to create a dynamic, or one-time password – a password that is only valid for a single login and then is not subject to a replay attack. The creation of a one-time password may be based on timing, where the password changes every minute or two; or the passwords may be based on events, where the password changes every time the user pushes a button on the device or uncovers a hidden value (scratch card).

Some ownership-based systems are synchronous (such as the timing and event-based systems), while others are based on a challenge-response scheme (asynchronous). A challenge response scheme operates on the principle that the access control server sends a challenge to the user. The user must then reply (respond) back with the correct response to the challenge. This response could be generated by a token or may just be the challenge encrypted using the user's password.

Tokens come in many types and are used to prove one's identity electronically. An example is a bank ATM card, which the customer uses to access the bank account associated with the card. The token is used in place of a password – or addition to it, as in the case of a PIN that the customer must enter each time the card is used – to verify the customer's identity. The token acts like an electronic key to access something. All tokens contain some secret information that are used to prove identity.

Ownership-based systems have the weakness that the loss or failure of the device may result in denial of service for the user; or a stolen device could be used by an unauthorized person to log into the organization.

Uses of Tokens Information

Tokens contain either passwords or some form of cryptographic function that authenticates the holder of the token as a valid entity. Tokens fall into one of four main categories.

- **Static:** This type of device actually contains a password. The password is transmitted to the authentication server at logon. Transmitting any password would leave a system open to a replay attack.
- **Synchronous:** Using an in-built algorithm and a timer this device generates a password which is constantly changing. This change is based on either the time or an event.
    - Using time-based tokens, the password changes at a timed interval, typically every 60 seconds. This works by placing a secret value (a key) on the token and then using that value to encrypt the current time.
    - Typically, the steps involved are:
        - The user reads the value from the token and enters the value, together with a PIN number, into the login window.
        - The authentication server calculates its own comparative value based on the synchronized time value and the respective access control subject's PIN. If the compared values match, access is granted.
        - RSA's SecurID is a popular choice for synchronous tokens.
    - Event based tokens create a new password it is each time the token is activated. Each activation will require a password or PIN number thus adding additional security to the device.

- In both cases the token and the authentication server must be synchronised, because the token is battery powered (usually with a non-removable battery) over time the token and the authentication server will slip out of sync and will need to be re-synced in the server software.
- **Asynchronous:** This device uses the cryptographic concept of a one-time pad to generate the one-time use password.

Challenge response: This device uses the elements of a PKI (Public Key Infrastructure). The token contains the entities private key. This is used to decrypt a random challenge sent from the authentication server, which has been encrypted with a copy of the tokens public key. If the token is capable of completing the decryption phase it sends the decrypted value back to the authentication server. Given that the public/private keys are mathematically related this process provides authentication.

Asynchronous Token

An asynchronous token, such as the event-driven, asynchronous token from Secure Computing called the SafeWord, eToken PASS, provides a new one-time password with each use of the token. It can be configured to expire on a specific date, but its lifetime depends on its frequency of use.

The token can last between five and 10 years and effectively extend the time period typically used in calculating the total cost of ownership in a multifactor authentication deployment. In the use of an asynchronous one-time password token, the access control subject typically executes a five-step process to authenticate identity and have access granted:

1. The authentication server presents a challenge request to the access control subject.
2. The access control subject enters the challenge into the token device.
3. The token device mathematically calculates a correct response to the authentication server challenge.
4. The access control subject enters the response to the challenge along with a password or PIN.
5. The response and password or PIN is verified by the authentication server and if correct, access is granted.

The use of a PIN, together with the value provided from the token, helps to mitigate the risk of a stolen or lost token being used by an unauthorized person to gain access.

Mathematical-Algorithm-Based One-Time Passwords

Hash chains (the repeated use of a hashing function) can be used to generate a complex series of one-time passwords from a secret shared key. Given that each password is unique, it is very unlikely that unauthorized user would be unable to determine what the password expected by the authentication server would be, even if the attacker had access to previous password.

Disconnected Tokens

Disconnected tokens require no connection either logical or physical.

When using time-synchronization the synchronization is done before the token is distributed to the client. As discussed earlier, these device display a value (the password) which is entered via the login.

Connected Tokens

Connected tokens are physically connected to the computer with which the user is being authenticated.

With these types of tokens, the time synchronization occurs when the token is inserted into an input device. The token automatically transmits the authentication information when a connection is made.

These types of devices do require an input device to be installed, either built-in (to the keyboard etc.) or plugged in via a port USB or Firewire (1394).

Contactless Tokens

Using an in-built antenna RFID and NFC devices enable the use of contactless connectivity to transmit login credentials or payment information. While this may be more convenient for the user than the devices previously mentioned they do present the addition problem of snooping or cloning.

Bluetooth and Mobile Device Tokens

Bluetooth tokens are often combined with a USB token, thus working in both a connected and a disconnected state. Bluetooth authentication works when closer than 32 feet (10 meters). If the Bluetooth is not available, the token must be inserted into a USB input device to function. A mobile device such as a smartphone or tablet computer can also be used as an authentication device, providing secure two-factor identification.

Smart Cards

Typically, smart cards are credit-card size, contain a tamper-resistant security system, are managed by a central administration system, and require a card reader device, such as the typical card reader on an ATM or fuel pump at a gasoline station. There are contact and contactless smart cards and readers.

- A contact card reader requires physical contact with the card reader. There are two primary types of contact for these readers: landing and friction. A landing contact requires physical contact with the contacts (landing zone) on the card when it is placed within the reader. Typical standards for landing contact readers include ISO 7816. Landing contact readers are popular in physical access applications. A friction contact requires that the card landing contacts are swiped against the contact reader. Typical friction card readers are those used in credit card transactions at merchants.
- Contactless card readers are quickly gaining in popularity and typically rely on radio-frequency identification (RFID) technology to facilitate reading. The additional security mechanisms found in contactless card applications can include challenge/response-based encryption safeguards to reduce the risk of “card skimming” whereby the account information is stolen in an otherwise legitimate transaction.

Radio Frequency Identification (RFID)

Radio-frequency identification (RFID) is the wireless non-contact use of radio-frequency electromagnetic fields to transfer data for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Some tags are powered and read at short ranges, typically a few meters, via magnetic fields. Others use a local power source such as a battery or else have no battery but collect energy from the interrogating EM field, and then act as a passive transponder to emit microwaves or UHF radio waves. Battery-powered tags may operate at hundreds of meters. Unlike a bar code, the tag does not necessarily need to be within line of sight of the reader and may be embedded in the tracked object.

Some common problems with RFID are reader collision and tag collision.

- Reader collision occurs when the signals from two or more readers overlap. The tag is unable to respond to simultaneous queries. Systems must be carefully set up to avoid this problem; many systems use an anti-collision protocol (also called a singulation protocol). Anti-collision protocols enable the tags to take turns in transmitting to a reader.
- Tag collision occurs when many tags are present in a small area; but since the read time is very fast, it is easier for vendors to develop systems that ensure that tags respond one at a time.

Since the tags can be read without being swiped or obviously scanned (as is the case with magnetic strips or barcodes), anyone with an RFID tag reader can read the tags embedded in clothes and other consumer products without knowledge. For example, customers could be scanned before entering a store to see what they are carrying. A customer might then be approached by a clerk who knows what is in the customer's backpack or purse, and the clerk can suggest accessories or other items. For various reasons, RFID reader/tag systems are designed so that distance between the tag and the reader is kept to a minimum. However, a high-gain antenna can be used to read the tags from much further away, leading to privacy problems.

One of the main concerns with RFID tags is that their contents can be read by anyone with an appropriately equipped scanner – even after it has been taken out of the store. One technology that has been suggested is a zombie RFID tag, a tag that can be temporarily deactivated when it leaves the store. The process would work like this: a customer brings a purchase up to the register, the RFID scanner reads the item, the customer pays for it, and as the customer leaves the store, a special device sends a signal to the RFID tag to "die." That is, it is no longer readable. The "zombie" element arises if a customer brings an item back to the store. A special device especially made for that kind of tag "re-animates" the RFID tag, allowing the item to re-enter the supply chain.

Characteristic

Characteristic-based authentication has also been in use for thousands of years as fingerprints were used to validate contracts in ancient Mesopotamia, just as they are used to log onto smartphones and laptops today. This form of authentication is commonly called biometrics ((bio meaning life, and metrics meaning measures). Biometrics today takes two primary forms, physiological and behavioral.

A biometric authentication solution entails two processes – enrollment and verification.

- During the enrollment process, the user's registered biometric code is either stored in a system or on a smart card that is kept by the user.
- During the verification process, the user presents his or her biometric data to the system so that the biometric data can be compared with the stored biometric code.

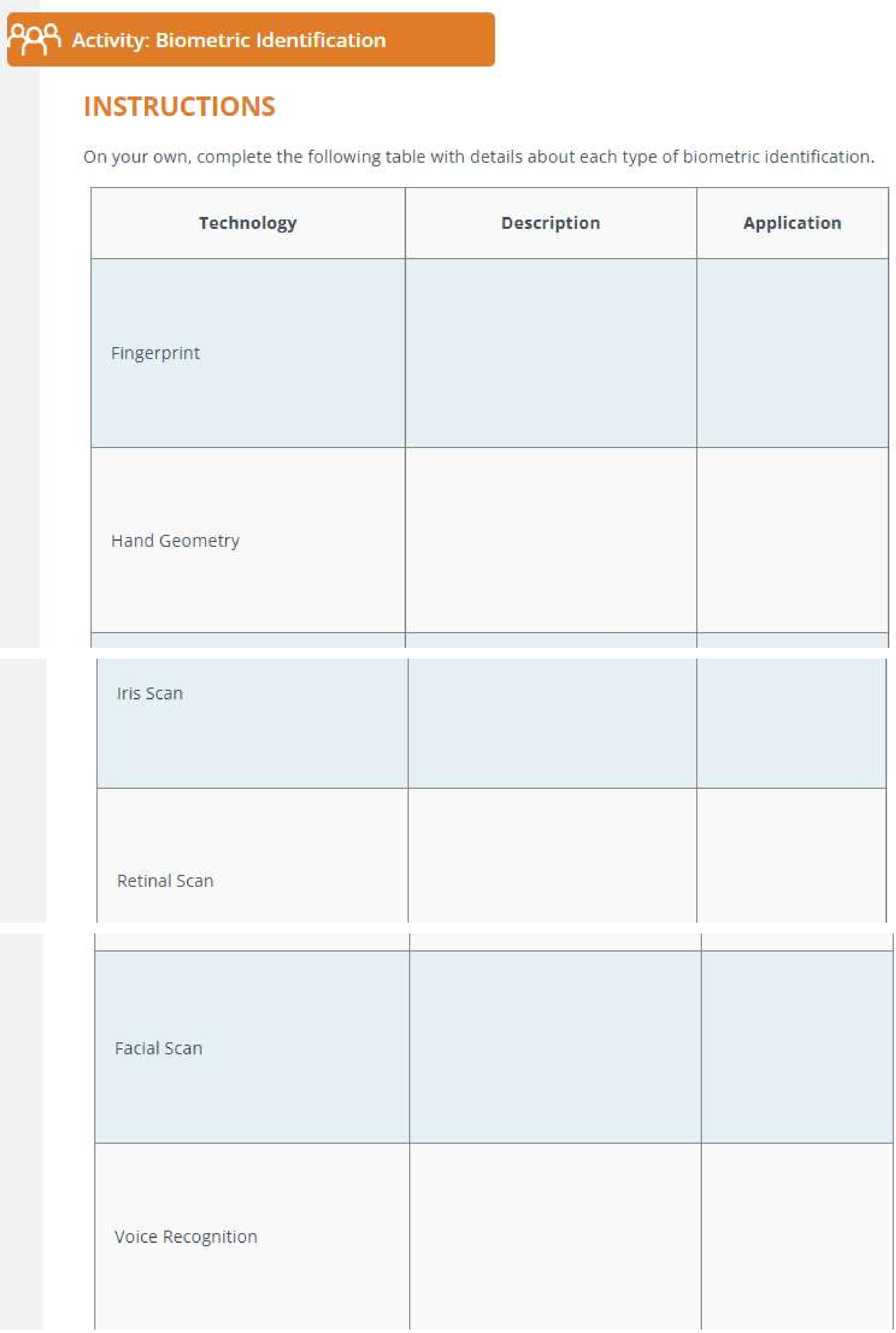
User verification is carried out either within the smart card (on-card matching), or in the system outside the card (off-card matching). The on-card matching algorithm protects the user's stored biometric code; as such, the code is not necessarily transferred to the outside environment. Even though the biometric data may not be considered to be secret, it is personally identifiable information (PII) and the protocol should not reveal it without the user's consent.

Physiological systems measure the characteristics of a person such as a fingerprint, iris scan (the colored portion around the outside of the pupil in the eye), retina scan (the pattern of blood vessels in the back of the eye), palm scan, and venous scans that look for the flow of blood through the veins in the palm. Some of the biometrics devices will actually combine processes together – such as checking for pulse and temperature on a fingerprint scanner – to resist counterfeiting.

Behavioral systems measure how a person acts by measuring voiceprints, signature dynamics and keystroke dynamics. A keystroke dynamics system measures behaviors such as the delay rate (how long a person holds down a key) and transfer rate (how rapidly a person moves between keys) as the person types.

Biometric systems are considered to be highly accurate but they are also plagued by challenges. Biometric systems are rather expensive to implement and maintain due to the cost of purchasing equipment and registering all the users. Users may also be uncomfortable with the use of biometrics, considering them to be an invasion of privacy, a risk of disclosure of medical information (retina scans can disclose medical conditions). A further drawback is the challenge of sanitization of the devices.

The implementation of biometric systems can also be a good example of the challenge of finding the ideal setting, or "sweet spot," for the sensitivity of the device. The sensitivity of biometric devices can usually be adjusted, and this will affect the accuracy of the device.



Activity: Biometric Identification

INSTRUCTIONS

On your own, complete the following table with details about each type of biometric identification.

Technology	Description	Application
Fingerprint		
Hand Geometry		
Iris Scan		
Retinal Scan		
Facial Scan		
Voice Recognition		

Biometric Implementation Issues

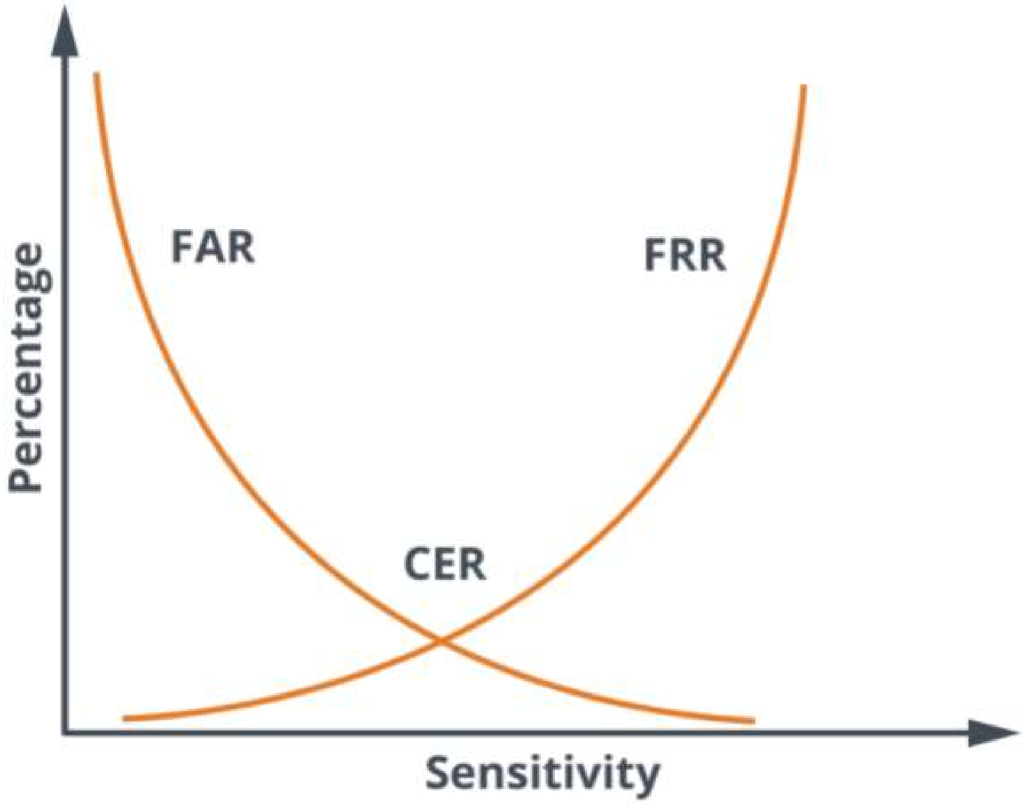
User acceptance is one of the most critical factors in the success of any biometric-based implementation. To minimize the risk of improper use, which can cause failed access, the device should not cause discomfort or concern and must be easy to use.

Biometric Accuracy

Biometric accuracy is measured by two distinct rates:

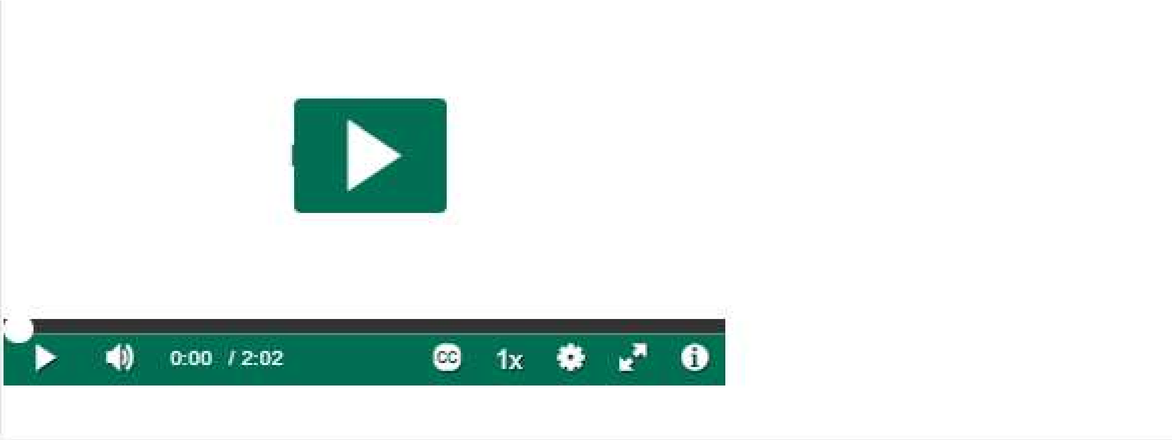
- False Rejection Rate (FRR), or Type I error
- False Acceptance Rate (FAR), or Type II error

The actual methodologies of the measurement of accuracy may differ in each type of biometric device, but simply put, a good comparative accuracy factor can be obtained by looking at the intersection point at which the Type I error rate equals the Type II error rate. This value is known as the crossover error rate (CER). The biometric device accuracy increases as the crossover value becomes smaller.



In reusable password authentication, the access control subject had to remember a password, possibly a difficult one. In token-based authentication, the access control subject had to retain possession of the token device. In biometric, characteristic-based authentication, the actual access control subject is the authentication device.

[redacted]



Strong Authentication

Each of the authentication techniques we have looked at has advantages and disadvantages, and no one method is good enough on its own to protect the assets of the organization. The solution is to use a combination of several authentication techniques together – for example, a smart card with a PIN to combine ownership with knowledge. An implementation of two of the same factor (e.g., facial recognition software with a fingerprint) is not usually considered to be two factor since both authentication techniques are the same. Most security solutions today recommend the use of at least two-factor or three-factor authentication.

Next-Generation Identification (NGI) Database

The accuracy of biometrics has been found to improve when multiple techniques are combined. To that end, and to increase the accuracy of suspect identification, the FBI started work on an NGI database in 2011 (first pilots launched in 2013) that will combine the use of fingerprint, face, iris, and palm matching capabilities to improve overall accuracy. The NGI system will offer state-of-the-art biometric identification services and provide a flexible framework of core capabilities that will serve as a platform for multimodal functionality. The projected NGI system capabilities are as follows:

1. **Quality Check Automation:** The QC Automation capability has eliminated the manual review of the majority of fingerprint transactions. Approximately 15% still require a manual review. To explain how QC affects response times, the average processing time with QC automation is approximately 0.7 seconds, as opposed to a manual QC processing time of 16.1 seconds.
2. **Interstate Photo System Enhancements:** The IPS will allow for easier retrieval of photos and include the ability to accept and search for photographs of scars, marks, and tattoos. In addition, this initiative will also explore the capability of facial recognition technology.
3. **Advanced Fingerprint Identification Technology:** Advanced Fingerprint Identification Technology will provide faster, more efficient IAFIS identification processing, increased search accuracy, improved latent processing services, and allow for seamless searches of ten-flat fingerprint impressions for noncriminal justice purposes.

4. **Enhanced IAFIS Repository:** Modifications will be made to incorporate multimodal biometric identification capabilities for future needs. There will also be an Iris Repository developed, which will provide for the submission of iris data, provide retrieval capability, provide iris search capability, and provide iris maintenance capability.
5. **FBI National Palm Print System:** The NGI Program will include the capability for the IAFIS to accept, store, and search palm print submissions from local, state, and federal law enforcement and criminal justice agencies. The National Palm Print System will provide a centralized repository for palm print data that can be accessed nationwide.

Department of Defense (DoD) Electronic Transmission Specification (EBTS)

The Department of Defense (DoD) Electronic Biometric Transmission Specification (EBTS) was developed by the Biometrics Identity Management Agency (BIMA) to transport and store biometric data and associated DoD-relevant information. This information is transferred from biometric collection devices to a BIMA storage, matching, and distribution point. Biometric matching services are provided by BIMA to the DoD and its information-sharing partners using Automated Biometric Identification System (ABIS).

ABIS is a central biometric storage and matching engine that responds to DoD EBTS match request transactions. ABIS sends biometric matching results and distributes biometric and associated information. ABIS transactions can be used to exchange information in one of several traditional formats, or in an Extensible Markup Language (XML) format. The DoD EBTS XML schema can support fast, efficient transactions using an analogous Abstract Syntax Notation One (ASN.1) schema that can transfer and exchange information in both compact binary and XML markup formats.

The latest version, DoD EBTS 3.0, was published as an emerging standard in 2011. It is based on the American National Standards Institute/National Institute of Standards and Technology (ANSI/NIST) Information Technology Lab (ITL) standard. ITL and DoD EBTS transactions are not signed objects. These standards rely on an optional ITL Type-98 Information Assurance record to protect selected content in environments where use of the record is mandated. Best ITL guidance calls for the Type-98 record to contain a SignedData message, such as the version defined in the X9.73 CMS standard.

Along with DoD EBTS v3.0, two authoritative data products, the Baseline Application Profile v1.0 and Integrated Data Dictionary v5.0 are also provided. The Baseline Application Profile v1.0 defines a standard set of transactions (TOTs) and is the first Application Profile for DoD EBTS v3.0. The Integrated Data Dictionary v5.0 defines all the data elements and the characteristics (e.g., field size, character type, valid values) referenced by both DoD EBTS and application profiles.

Activity: Biometric Characteristics

INSTRUCTIONS

Answer the following questions.

1. What are three types of behavioral biometrics?
2. What are the two steps in biometric authentication?

Activity Answers: Biometric Characteristics

Answers

1. What are three types of behavioral biometrics?

    a. Keystroke

    b. Signature analysis

    c. Voice recognition

2. What are the two steps in biometric authentication?

    a. During the enrollment process, the user's registered biometric code is either stored in a system or on a smart card that is kept by the user.

    b. During the verification process, the user presents his or her biometric data to the system so that the biometric data can be compared with the stored biometric code.

Denial of Service

The problem with access controls is that the controls affect legitimate users a lot more frequently than they stop unauthorized users from gaining access. The organization's helpdesk has to deal with forgotten passwords on a daily basis and this costs the organization time and money – both for the helpdesk personnel and the loss of productivity for the user who is locked out of the systems. Organizations have also fallen victim to attacks where the attacker would cause a mass lockout of all the user accounts of a system.

As an example, an attacker who knew that many users select weak or predictable passwords conducted an intriguing attack on a bank in China. He selected a word that was quite possibly going to be used as a password and then tried to log into a user's account at the bank. He knew that if he tried several times then that user's account would be locked out, so instead of trying multiple passwords on one account he wrote a script that would try to log into each account number of the bank using the password he had chosen. As a result, he found that several dozen users of the bank's online banking system were using the word he chose. Each time he was able to log in he then transferred money from the victim's account to his own.

Entitlement

All entities having their identity defined using interoperable identifiers allows for rich risk-based decisions to be made. This is "entitlement" — a set of rules, defined by the resource owner, for managing access to a resource (asset, service, or entity) and for what purpose. The level of access is conditioned not only by your identity but is also likely to be constrained by a number of further security considerations.

For example, other factors could include your company policy, your location (i.e., are you inside your secure corporate environment, connected via a hotspot or from an Internet café, etc.) or time of day.

Entities

There are five types of "entity" that require digital identity:

- People
- Devices
- Organizations
- Code
- Agents

For example, a laptop is a device that needs identity. Potentially, this device is a company-owned laptop and, therefore, will have a "corporate laptop" persona involving an organization identity. The laptop is running code (include data in this term), and this code needs to be trusted, therefore, necessitating both identity and attributes. Finally, there are agents — someone or something you give authority to act on your behalf. For example, you may give your personal assistant the authority to use specified attributes of your business credit card and frequent flyer personas to book your travel, but your assistant would use his/ her identity. Identity needs to encompass all these entities to ensure a trusted transaction chain.

Accounting

Accounting (sometimes called auditing) is not the last phase in the identity management lifecycle process since it happens throughout each of the phases described previously. Accounting is the process of creating a record or logging all activity on the system. Each time a person logs in or out, attempts to access an application or a customer record, a log entry is created for later review. This is one reason that a UserID must be unique since that is the only way to associate a particular action with the person or process initiating that action.

Logs should be protected to prevent the deletion of log entries or the deletion of the log file entirely. Sometimes protection is done by writing a log off to a separate system that even the administrator cannot access. These logs may be needed in case of an investigation or to prove compliance with laws or regulations.

Reviewing logs can be a time-consuming process and it can be hard to justify log review when there are other more pressing issues, but logs can be reviewed using tools to filter out critical data. Moreover, log review may alert the organization to activity that may be ongoing as part of an attack, even if the attack has not yet been successful. The challenge is to log the correct data and in the correct place to enable identification of a problem. As one security manager observed after a breach that had lasted for six weeks, he found that the logs contained almost nothing of value since they were logging the wrong things. Feeding log data into a security information and event management (SIEM) system may also facilitate the capturing and analysis of log data by correlating data from many sources and gaining a larger perspective of the relevance of the log data.

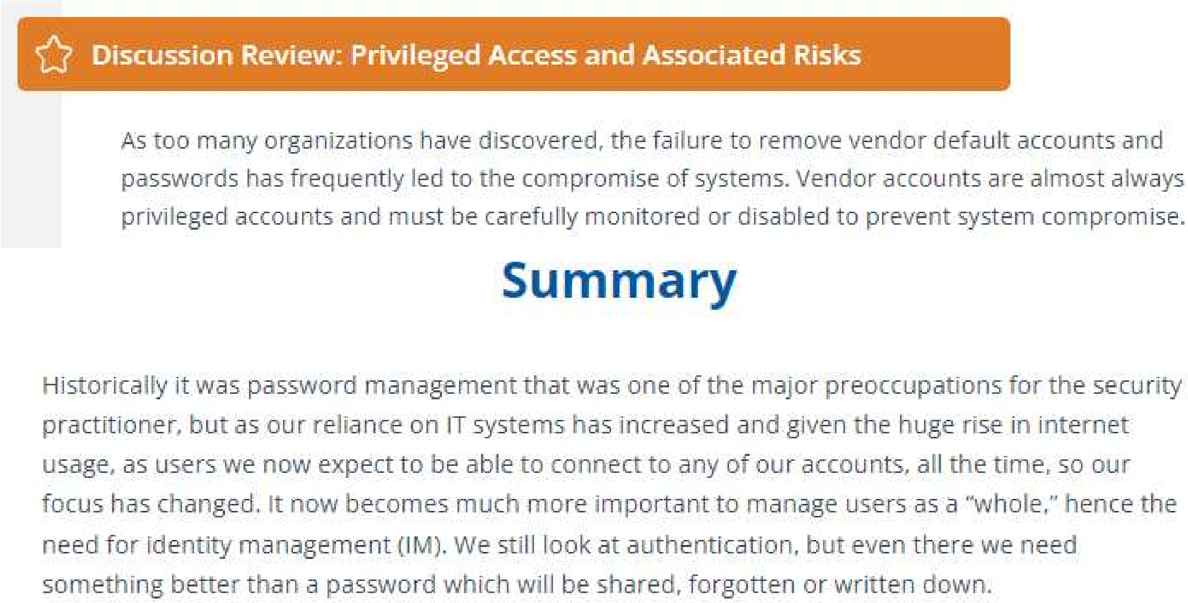
Privileged Access

One of the challenges of access control is that there are many different levels of access that need to be managed. It was much simpler to manage access decades ago, when the only people on our systems were employees, but now we need to ensure that each user has the level of access they should have, and that they do not have access permissions beyond what they should have.

Some users of our systems require privileged access: administrators, maintenance personnel, and management. These people require a level of access that poses a risk to the system itself or the data it manages. Misuse of privileged access can disable the system, delete data, remove security controls and lead to a security breach. Least privilege for such personnel is still a high level of privilege.

Discussion: Privileged Access and Associated Risks

What are some steps that can be taken to reduce the risk associated with users that have privileged access to systems or equipment?



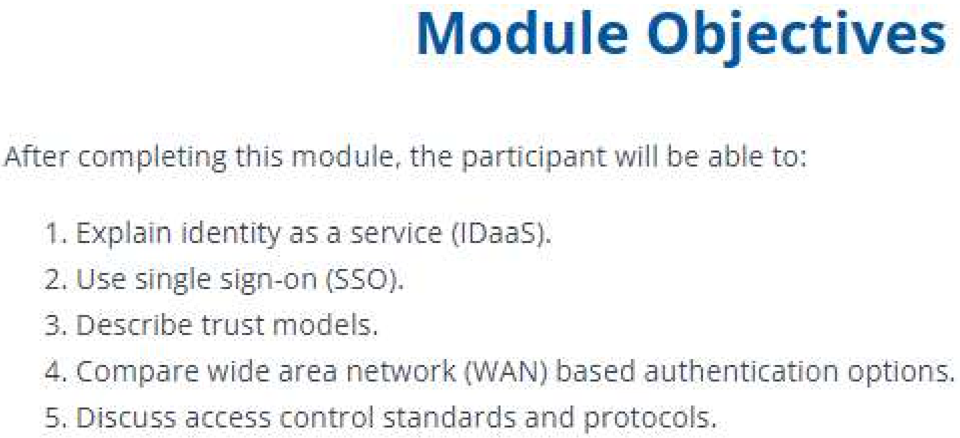
Discussion Review: Privileged Access and Associated Risks

As too many organizations have discovered, the failure to remove vendor default accounts and passwords has frequently led to the compromise of systems. Vendor accounts are almost always privileged accounts and must be carefully monitored or disabled to prevent system compromise.

Summary

Historically it was password management that was one of the major preoccupations for the security practitioner, but as our reliance on IT systems has increased and given the huge rise in internet usage, as users we now expect to be able to connect to any of our accounts, all the time, so our focus has changed. It now becomes much more important to manage users as a "whole," hence the need for identity management (IM). We still look at authentication, but even there we need something better than a password which will be shared, forgotten or written down.

Module 4: Implement and Maintain Authentication Methods - Identity and Access Management



Module Objectives

After completing this module, the participant will be able to:

1. Explain identity as a service (IDaaS).
2. Use single sign-on (SSO).
3. Describe trust models.
4. Compare wide area network (WAN) based authentication options.
5. Discuss access control standards and protocols.

Overview

Identity and Access Management (IAM) Implementation

The implementation of access management contains its own challenges. Audits in many organizations often reveal that the identity management processes used are flawed, resulting in many users who have access permissions that they have accumulated over the years that are not aligned with their current business needs. This is a problem where privacy regulations require accountability and tracking of access permissions, and it can lead to financial penalties, security breaches, and embarrassment for the organization.

The idea of an identity and access management (IAM) system is to automate the process and reduce the administrative overhead, while improving reporting and the ability to monitor the access levels granted to users. Some of the features of IAM systems include an automated process for users to request and be granted access to systems, a streamlined process for new users and for password resets.

Decentralized Access Control Management

System administrators and the local LAN administrator originally handled all access control locally. When a user needed access, the manager had to request access from the local IT staff, who would then arrange that access—often days later. This was an inefficient process but since each system and each LAN operated as a separate entity in the early days, it was really the only way to manage access.

As systems and networks merged and were integrated across departments and buildings, the local management of access became untenable. Administrators could not keep up with the workload and effectively manage access for a mobile and diverse workforce. So automation and integration of access permissions were essential.

There are advantages to decentralized access management. Locally managed access is more responsive to local needs in that a manager can quickly have access adjusted as workloads and staffing require. Access rules and procedures are subject to local direction and oversight, and the responsibility to manage access permissions is managed by the managers closest to the users.

Centralized Access Control Management

Many organizations have migrated to a centralized access control framework, and this has many advantages. With a centralized department that manages access for all staff regardless of work location, access is managed in a consistent manner. Defined procedures can be set up that ensure access is only granted in compliance with centrally defined policies, privacy laws, and standards. This avoids the problems with decentralized access management where various local managers can do things their own way in an inconsistent and unpredictable manner.

The deployment of centralized access management leads to better reporting capabilities and visibility for senior management into compliance with legislation.

The deployment of centralized access management has also led to the use of single sign-on (SSO) solutions that enable a user to use a single set of identification and authentication credentials to gain access to multiple systems. These are discussed below.

However, there is a drawback. Local offices may be frustrated with a procedure-driven centralized process that is slow and unbending. It may not be possible for a local manager to have access controls changed quickly (perhaps because a member of staff is absent) when the centralized process can only promise to address an access change request in 48 hours.

Single Sign-on (SSO)

Single sign-on (SSO) is a very generic term that refers to many very different technologies and solutions ranging from the older script-based single sign-on solutions of the 1990s through to Identity as a Service (IDaaS) in the current cloud environment.

As computer systems became more common in the 1990s, the problem of managing multiple IDs became apparent. The average user was managing 12–15 UserIDs, each with their own password rules and expiration dates. The problem continued getting worse by an order of magnitude as systems and UserIDs proliferated. So the idea of SSO was born. With SSO, users would only have to remember a single UserID and password to be granted access to all of their systems.

Early password-based systems were rather insecure. Many of them actually transmitted passwords in cleartext across a network. Examples of this were Password Authentication Protocol (PAP), which was designed to allow a user to connect to a system – often to connect to their local Internet Service provider (ISP)) over a modem-based telephone line. This was understandable over a telephone-based network since telephone lines were a little harder to intercept than today's internet, but when those services moved onto the internet, the use of PAP continued, making the level of security unacceptably low.

Other early systems were based on scripts where a front-end program would simulate the steps a user would take to log in to an application. The user would now enter the application using the script that logged them in automatically. The risk again was that many of these scripts also passed the passwords in plaintext over the network.

We use many different SSO products today. Many of them will store a user's passwords for them so that the user does not have to remember all of their passwords for multiple systems. The greatest disadvantage of these solutions is the risk of compromise of the password storage area. These password storage areas may be protected by a weak password or passphrase chosen by the user and easily compromised. We have seen many cases where a person's private data was stored by a cloud provider but easily accessed by unauthorized persons through password compromise.

Popular SSO solutions today include Federated Identity Management, web portals, OAUTH, and other similar solutions that allow a person to log in to a common application and then be able to access multiple other systems or applications based on their initial login. This uses protocols such as SAML (Security Assertion Markup Language) to pass security assertions or login credentials between enterprises that trust one another. (Since this course is careful to avoid being vendor-specific, these solutions have not been named here, but we have all seen them as we can access a website using credentials linked to an email or social media provider.)

SSO Risks

Single point of failure: With all of the user's credentials stored on a single authentication server, the failure of that server can prevent access for those users to all applications for which it had provided authentication services.

Single point of access: Because SSO affords a single point of access, it is more prone to mass denial-of-service attacks whereby entire groups of users can be denied access to systems by attacking the single point of access.

Kerberos

Kerberos, described in RFC 1510, was originally developed by the Massachusetts Institute of Technology (MIT) and quickly became a popular network authentication protocol for indirect (third-party) authentication services. It is an interesting example of how SSO can work and is built into many products today.

Kerberos uses symmetric encryption (explained later) to encrypt the exchange of messages between the Users, Key Distribution Centers (KDC), and the Applications. This provides confidentiality of the communications but does not protect against other forms of attack, such as a brute force on the passwords or compromise of the data stored in the KDC.

It is designed to provide strong authentication using that symmetric cryptography. It is an operational implementation of key distribution technology and affords a key distribution center, authentication service, and ticket granting service. Hosts, applications, and servers all have to be "Kerberized" to be able to communicate with the user and the ticket granting service.

Kerberos Applications

Kerberos is based on a centralized architecture, thereby reducing administrative effort in managing all authentications from a single server. Furthermore, the use of Kerberos provides support for:

Authentication: A user is who he/she claims to be.

Authorization: What can a user do once properly authenticated?

Confidentiality: Keep data secret.

Integrity: Data received is the same as the data that was sent.

Nonrepudiation: Determines exactly who sent or received a message.

Kerberos Process

The process in the use of Kerberos is substantially different from indirect authentication technologies and is considerably more complex.

The following is a simplified explanation of the Kerberos process that was adapted for use here from the book *Applied Cryptography: Protocols, Algorithms, and Source Code in C* by Bruce Schneier (New York, NY: Wiley, 1993).

1. Before an access control subject can request a service from an access control object, it must first obtain a ticket to the particular target object; hence, the access control subject first must request from the Kerberos Authentication Server (AS) a ticket to the Kerberos Ticket Granting Service (TGS). This request takes the form of a message containing the user's name and the name of the respective TGS.
2. The AS looks up the access control subject in its database and then generates a session key to be used between the access control subject and the TGS. Kerberos encrypts this session key using the access control subject's secret key. Then, it creates a Ticket Granting Ticket (TGT) for the access control subject to present to the TGS and encrypts the TGT using the TGS's secret key. The AS sends both of these encrypted messages back to the access control subject.
3. The access control subject decrypts the first message and recovers the session key. Next, the access control subject creates an authenticator consisting of the access control subject's name, address, and a time stamp, all encrypted with the session key that was generated by the AS.

4. The access control subject then sends a request to the TGS for a ticket to a particular target server. This request contains the name of the server, the TGT received from Kerberos (which is already encrypted with the TGS's secret key), and the encrypted authenticator.
5. The TGS decrypts the TGT with its secret key and then uses the session key included in the TGT to decrypt the authenticator. It compares the information in the authenticator with the information in the ticket, the access control subject's network address with the address from which the request was sent, and the time stamp with the current time. If everything matches, it allows the request to proceed.
6. The TGS creates a new session key for the user and target server and incorporates this key into a valid ticket for the access control subject to present to the access control object server. This ticket also contains the access control subject's name, network address, a time stamp, and an expiration time for the ticket – all encrypted with the target server's secret key – and the name of the server. The TGS also encrypts the new access control subject target session key using the session key shared by the access control subject and the TGS. It sends both messages to the access control subject.

7. The access control subject decrypts the message and extracts the session key for use with the target access control object server. The access control subject is now ready to authenticate himself or herself to the access control object server. He or she creates a new authenticator encrypted with the access control subject target session key that the TGS generated.
   To request access to the target access control object server, the access control subject sends along the ticket received from Kerberos (which is already encrypted with the target access control object server's secret key) and the encrypted authenticator. Because this authenticator contains plaintext encrypted with the session key, it proves that the sender knows the key. Just as important, encrypting the time of day prevents an eavesdropper who records both the ticket and the authenticator from replaying them later.

8. The target access control object server decrypts and checks the ticket and the authenticator, also confirming the access control subject's address and the time stamp. If everything checks out, the access control object server now knows the access control subject is who he or she claims to be, and the two share an encryption key that they can use for secure communication. (Since only the access control subject and the access control object server share this key, they can assume that a recent message encrypted in that key originated with the other party.)
9. For those applications that require mutual authentication, the server sends the access control subject a message consisting of the time stamp plus 1, encrypted with the session key. This serves as proof to the user that the access control object server actually knew its secret key and was able to decrypt the ticket and the authenticator.

Considerations

To provide for the successful implementation and operation of Kerberos, the following should be considered:

- Overall security depends on a careful implementation.
- Requires trusted and synchronized clocks across the enterprise network.
- Enforcing limited lifetimes for authentication based on time stamps reduces the threat of a malicious hacker gaining unauthorized access using fraudulent credentials.
- The Key Distribution Center must be physically secured.
- The Key Distribution Center must be isolated on the network and should not participate in any non-Kerberos network activity.
- The Authentication Server can be a critical single point of failure.

Kerberos is an excellent example of the greatest risk associated with most SSO solutions: the central and single point of failure, the KDC. Kerberos requires that all devices on the network use Kerberos to communicate, and if the KDC is unavailable then no one can communicate. Also, a compromise of the KDC would enable an attacker to have unlimited access to everything! This requires replication and physical protection of the KDC.

Kerberos Tools

Kerberos is available in many commercial products, and a free implementation of Kerberos is available from MIT. The tools used to support Kerberos in a Windows Server 2003 environment can be found as part of the Windows Server 2003 Resource Kit Tools. The Microsoft Windows Server 2003 Resource Kit Tools are a set of tools to help administrators streamline management tasks such as troubleshooting operating system issues, managing Active Directory, configuring networking and security features, and automating application deployment. The tools are:

- Kerbtray.exe: Kerberos Tray displays ticket information via a GUI. It allows a user to view and purge the ticket cache.
- Klist.exe: Kerberos List is a command-line tool that is used to view and delete Kerberos tickets granted to the current logon session. To use Kerberos List to view tickets, a user must run the tool on a computer that is a member of a Kerberos realm. When Kerberos List is run from a client, it shows the:
    - Ticket-granting ticket (TGT) to a Kerberos Key Distribution Center (KDC) in Windows
    - Ticket-granting ticket (TGT) to Ksserver on UNIX

- Ksetup.exe: Kerberos Setup is a command-line tool that can be used to configure Kerberos interoperability. Including (but not limited to):
    - Set up local account to Kerberos V5 account mappings
    - Set the computer's password in the Kerberos realm
    - Change a user's password in a Kerberos V5 realm

Network Ports Used During Kerberos Authentication

SERVICE NAME	UDP	TCP
DNS	53	53
Kerberos	88	88

Identity as a Service (IDaaS)

The emergence of cloud computing has created many new opportunities, cost savings, and benefits for organizations and introduced new challenges related to access control and privacy. When data is stored and processed by a cloud service provider (CSP), there are some challenges with ensuring that the necessary access controls and permissions are maintained over sensitive data. This requires service-level agreements and ways to provide assurance to management of compliance with laws and agreements, as we will see elsewhere in the course, but it also requires clear definitions of access controls and limits on the ability of users, administrators (especially the administrators working on behalf of the CSP), managers, and customers in regard to data access and modification.

One solution for managing access control over both cloud-based and non-cloud-based systems is through Identity as a Service (IDaaS). This is an Identity and Access Management (IAM) service provided by a third party to manage access permissions. In essence, IDaaS is a cloud-based SSO.

The IDaaS provider also handles the provisioning and deprovisioning components associated with identity management.

Gartner (a global research and advisory company) states that the core aspects of IDaaS are:

- **IGA:** Provisioning of users to cloud applications and password functionality.
- **Access:** User authentication, single sign-on, and authorization, supporting federation standards such as SAML.
- **Intelligence:** Identity access log monitoring and reporting.

Some of the top performers in the IDaaS space that are part of Gartner's Magic Quadrant:

- Centrify
- Okta
- Windows Active Directory Federated Services

The main risks associated with identity and access management are based on improperly maintained access permissions where a user has a level of access that is not aligned with current job responsibilities, and access failure through the failure of the access control solution or compromise of the access control system.

Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) is a protocol that supports the access and authorization required to link disparate organizations.

It uses the XML-based framework for describing and exchanging security information between online business relationships. The security information is maintained in SAML assertions that work between trusted security domain boundaries. The SAML standard follows a prescribed set of rules for requesting, creating, communicating and using SAML assertions.

SAML has three roles and four primary components.

SAML roles:

1. User or principal
2. Service provider or relying party
3. Identity provider (IdP)

SAML components:

1. **Assumptions** – defines how SAML attributes, authentication and authorization requests-response protocol messages can be exchanged between systems using common underlying communication protocols and frameworks.
2. **Bindings** – defines how SAML assertions and protocols message exchanges are conducted with response/request pairs.
3. **Protocols** – defines what protocols are used, including HTTP and SOAP.
4. **Profiles** - defines specific sets of rules for a use case for attributes, bindings and protocols for a SAML session.

Open Authorization (OAuth)

Internet Engineering Task Force (IETF) RFC 6749 states:
The Open Authorization (OAuth) 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service or by allowing the third-part application to obtain access on its own behalf.

OAuth standard has four roles:

- **Resource owner** – An entity capable of granting access to a protected resource. When the resource owner is a person, the entity is referred to as an end-user.
- **Resource server** – The server hosting the protected resources which is capable of accepting and responding to protected resource requests using access tokens.
- **Client application** – An application making protected resource requests on behalf of the resource owner and with its authorization. The term "client" does not imply any implantation characteristics (e.g. whether the application executes on a server, a desktop or other device).
- **Authorization server** – The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.

Trust

Trust is essential for many data exchanges and e-commerce transactions. The challenge is to know whom we can trust and to establish a way to build trust. When we are communicating over networks it is difficult to prove whom we are talking to and to ensure that the data we are being provided is legal, authentic, and complete.

When we call a person on the phone, they will often answer with their name or the name of the organization we have reached. This provides us with some assurance that we have reached the right person. We do the same on the internet. When we reach a website, such as the website of our bank, they often send us a certificate that can be used to validate we are on the correct website and not a website masquerading as the bank. The certificate also contains the public key of the bank and using that key, we know we can communicate confidentially with the bank. This process where we initiate the connection but where the responding website authenticates to us is called reverse authentication: the respondent authenticates to the originator. If the bank also asks the client that was reaching out to them for a certificate so that both parties had to exchanges certificates, this would be mutual authentication. Node authentication, where the calling device has to provide their MAC or IP address to the called site, is known as forward authentication. These three forms of authentication – reverse, mutual, and forward – allow us to establish trust and engage in e-commerce transactions with a level of assurance that we are communicating with the correct organization and not a spoofed website.

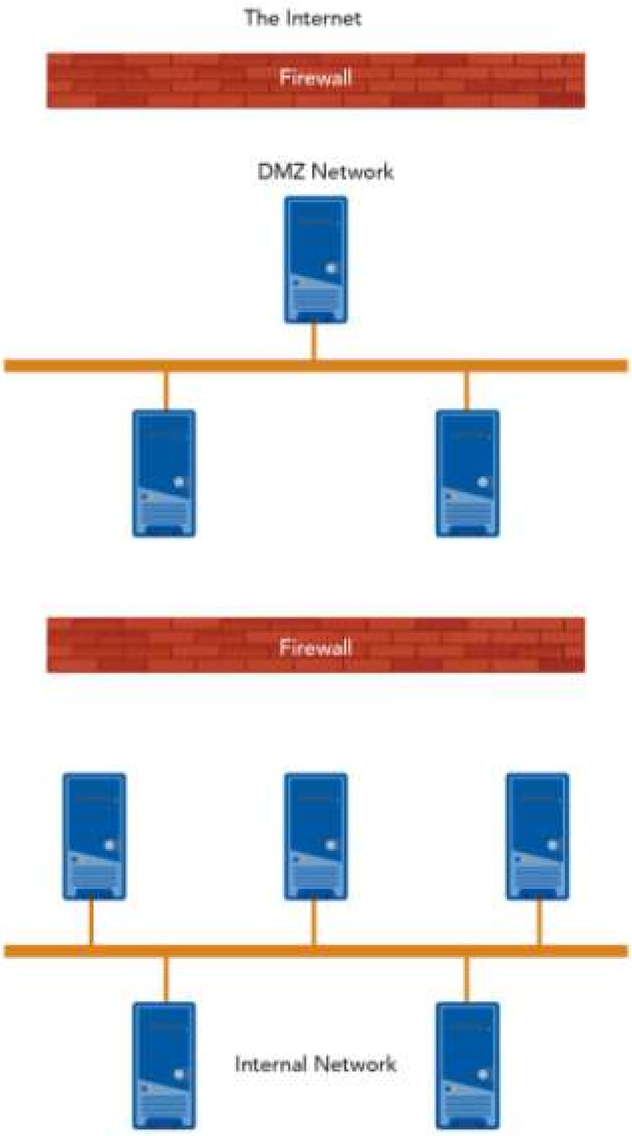
Comparing Internetwork Architectures

Computers are connected together using networks, and different types of networks provide different levels of trust. Primarily, there are four types of trust architectures:

- **Internet:** The internet is a global system of interconnected computer networks that use the standard internet protocol suite (TCP/IP) to link several billion devices worldwide. It encompasses millions of private, public, academic, business, and government packet switched networks, linked by a broad array of electronic, wireless, and optical networking technologies. The terms "internet" and "World Wide Web" are often used interchangeably in everyday speech; it is common to speak of "going on the internet" when invoking a web browser to view web pages. However, the World Wide Web (or the web) is actually just one of a very large number of services running on the internet. The web is a collection of interconnected documents (web pages) and other web resources linked by hyperlinks and URLs. In addition to the web, a multitude of other services are implemented over the internet, including email, file transfer, remote computer control, newsgroups, and online games. All of these services can be implemented on any intranet accessible to network users.
- **Intranet:** An intranet is a network based on TCP/IP protocols (an internet) belonging to an organization, usually a corporation, accessible only by the organization's members, employees, or others with authorization. Intranets utilize standard network hardware and software technologies such as Ethernet, WiFi, TCP/IP, web browsers, and web servers. An organization's intranet typically includes internet access but is firewalled so that its computers cannot be reached directly from the outside.

- **Extranet**: An extranet is a computer network that allows controlled access from the outside for specific business or educational purposes. Extranets are extensions to, or segments of, private intranet networks that have been built in many corporations for information sharing and e-commerce. In a business-to-business context, an extranet can be viewed as an extension of an organization's intranet that is extended to users outside the organization – usually partners, vendors, and suppliers – in isolation from all other internet users. An extranet is similar to a DMZ in that it provides access to needed services for channel partners, without granting access to an organization's entire network.
- **Demilitarized zone (DMZ):** A DMZ is a computer host or small network inserted as a "neutral zone" between a company's private network and the outside public network (see Figure 2.7). It prevents outside users from getting direct access to a server that has company data. In a typical DMZ configuration, a separate computer (or host in network terms) receives requests from users within the private network for access to websites or other companies accessible on the public network. The DMZ host then initiates sessions for these requests on the public network. However, the DMZ host is not able to initiate a session back into the private network.

The DMZ can only forward packets that have already been requested. In a DMZ configuration, most computers on the LAN run behind a firewall connected to a public network such as the Internet. One or more computers also run outside the firewall, in the DMZ. Those computers on the outside intercept traffic and broker requests for the rest of the LAN, adding an extra layer of protection for computers behind the firewall.


The Internet
Firewall
DMZ Network
Firewall
Internal Network

One-way Trust

A one-way trust is a unidirectional authentication path that is created between two domains. This means that in a one-way trust between Domain A and Domain B, users in Domain A can access resources in Domain B. However, users in Domain B cannot access resources in Domain A. Some one-way trusts can be either a non-transitive trust or a transitive trust, depending on the type of trust that is created.

Two-Way Trust

In a two-way trust, Domain A trusts Domain B, and Domain B trusts Domain A. This means that authentication requests can be passed between the two domains in both directions. Some two-way relationships can be either non-transitive or transitive depending on the type of trust that is created.

Transitive Trust

Transitive trust is the extension of trust between two parties to areas outside or beyond the original trust relationship. It can be defined in two ways:

1. If Domain A trusts Domain B and Domain B trusts Domain C, then Domain A could have transitive trust with Domain C.
2. The other approach is used in some directory-based products where a subdirectory will inherit trust from the parent directory.

Manage Network Access Control

This approach will not let a device connect to a network until it has been verified and proven to be compliant with corporate security policies. This can be done using Network Access Control (NAC) devices. When a person tries to connect a device to the network, it is placed into isolation (quarantine) until it has been checked for compliance with security baselines, such as the presence of an anti-virus product, up-to-date patches on applications and operating systems, and appropriate security settings. NAC is especially desirable with a mobile workforce that returns to the office after being connected to untrusted networks (e.g., hotels, coffee shops).

Network Access Protection

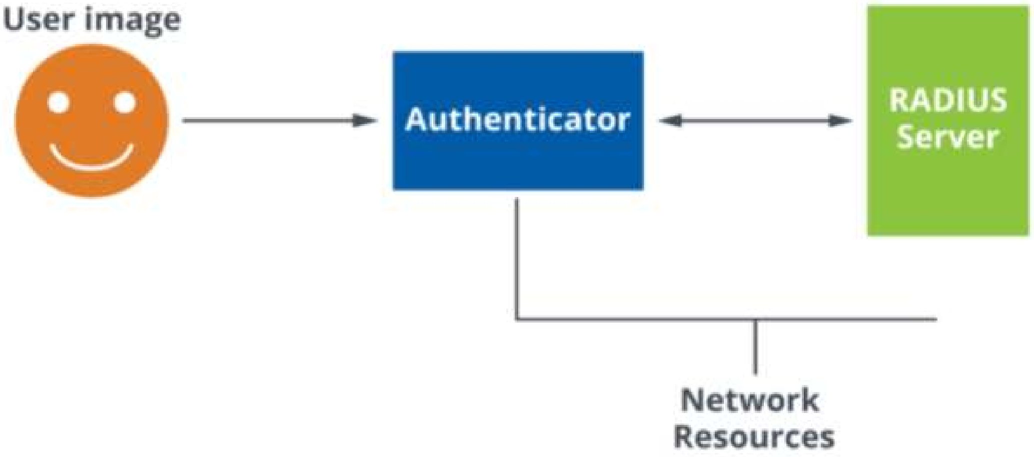
Network Access Protection was a Microsoft technology designed to enforce the NAC concept. Based on the "health" read security setting, configurations and so forth, a client would be allowed to join the network or not.

The system was of course based upon a security policy which defined the minimum-security settings required by the organization. If the client failed the examination, it could be directed to a Health Remediation Server that make and correction, such as applying patches, before releasing the device from quarantine.

Microsoft deprecated the technology in Windows Server 2016.

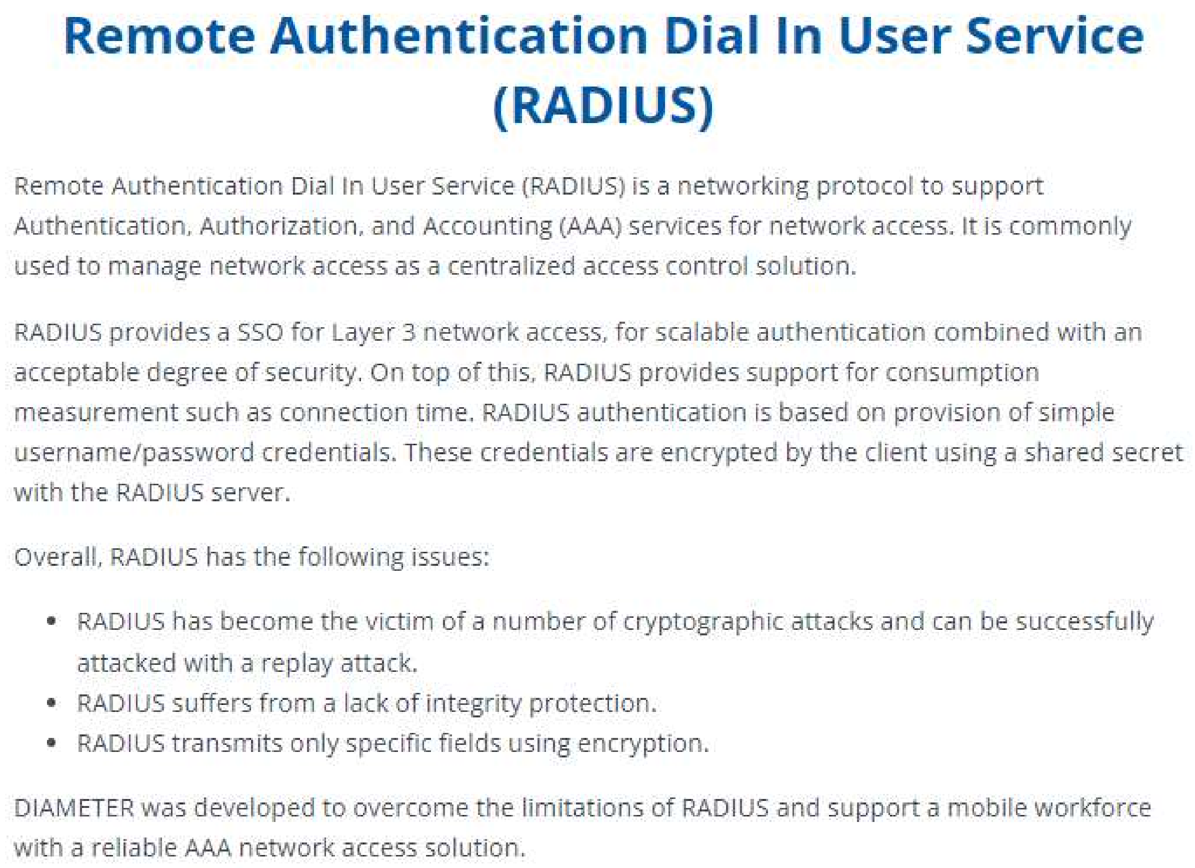
802.1x Standard

A method of controlling access to networks is through the IEEE 802.1x standard (see the figure below). This is a port-based network access control standard that uses Extensible Authentication Protocol over LAN (EAPOL) to enforce security policies.



802.1x requires three components:

- **The supplicant (user)** - provides the authenticator with their logon credentials (username and password) using EAP.
- **The authenticator (a network device)** - then communicates with an authentication server using RADIUS or Diameter.
- **The authentication server** - validates the credentials of the user and notifies the authenticator that access can be granted (or denied) to the requested network resources to the supplicant.



Remote Authentication Dial In User Service (RADIUS)

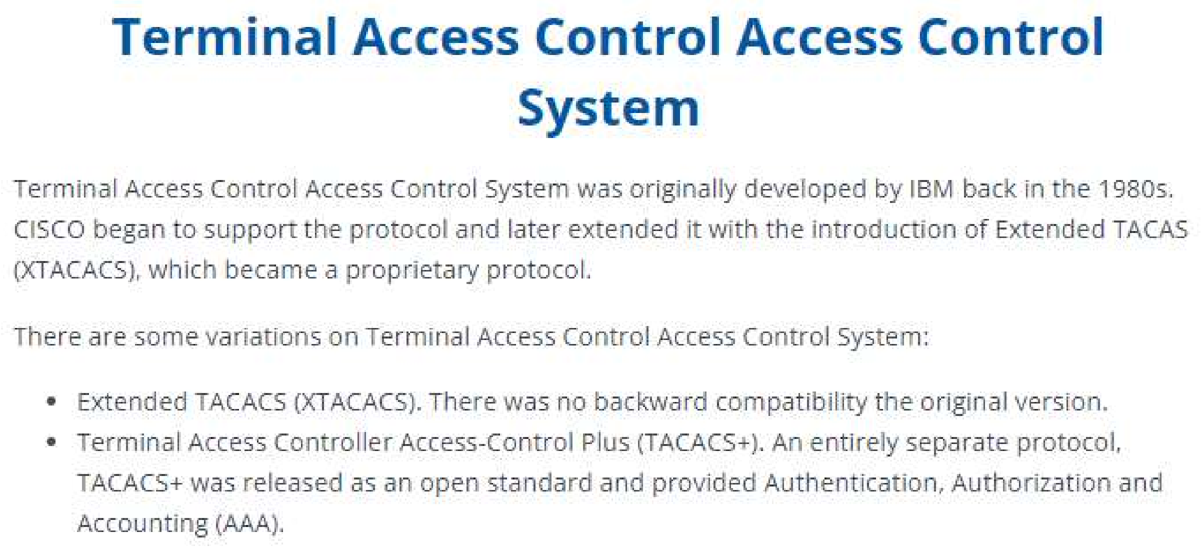
Remote Authentication Dial In User Service (RADIUS) is a networking protocol to support Authentication, Authorization, and Accounting (AAA) services for network access. It is commonly used to manage network access as a centralized access control solution.

RADIUS provides a SSO for Layer 3 network access, for scalable authentication combined with an acceptable degree of security. On top of this, RADIUS provides support for consumption measurement such as connection time. RADIUS authentication is based on provision of simple username/password credentials. These credentials are encrypted by the client using a shared secret with the RADIUS server.

Overall, RADIUS has the following issues:

- RADIUS has become the victim of a number of cryptographic attacks and can be successfully attacked with a replay attack.
- RADIUS suffers from a lack of integrity protection.
- RADIUS transmits only specific fields using encryption.

DIAMETER was developed to overcome the limitations of RADIUS and support a mobile workforce with a reliable AAA network access solution.



Terminal Access Control Access Control System

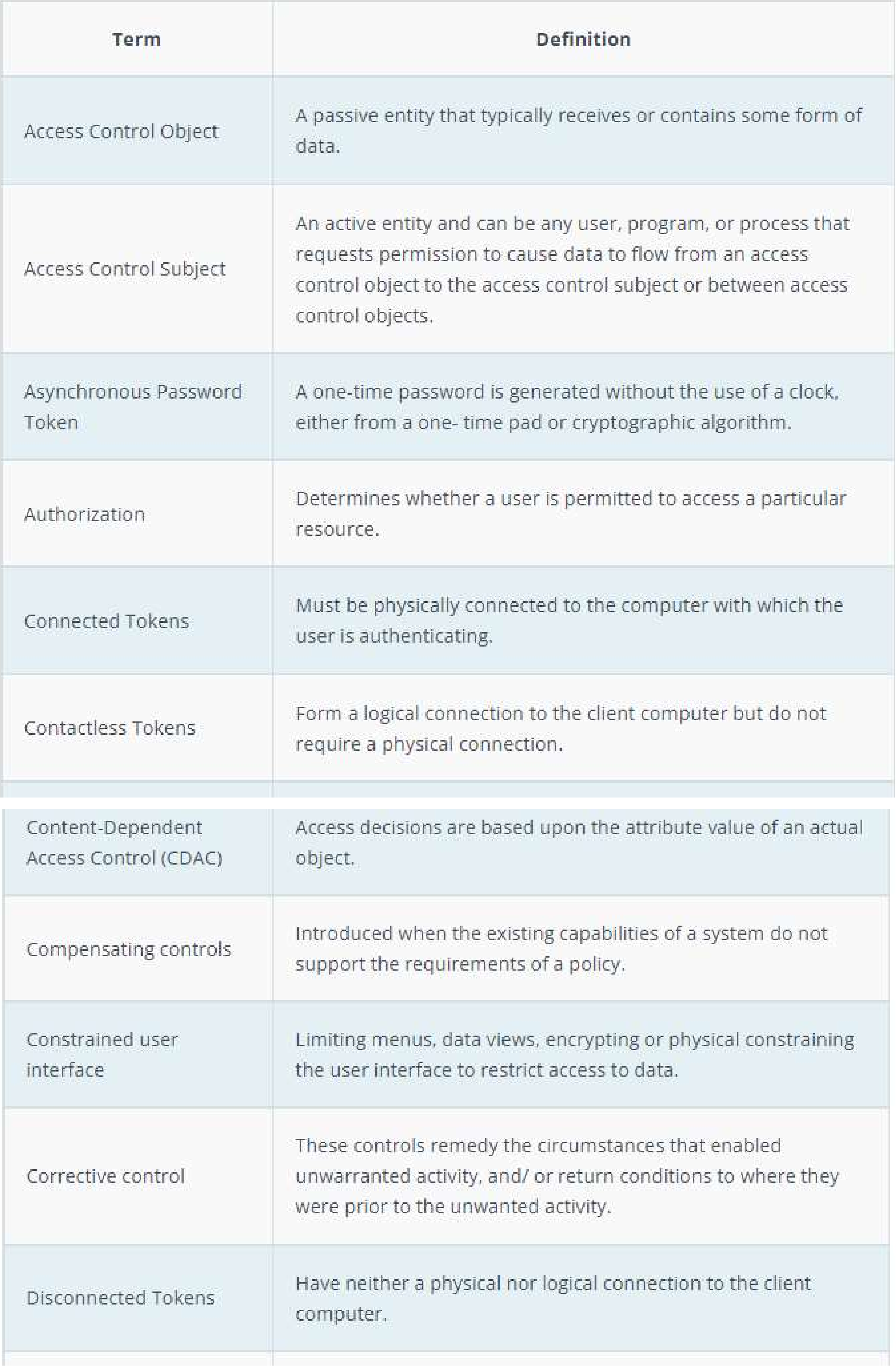
Terminal Access Control Access Control System was originally developed by IBM back in the 1980s. CISCO began to support the protocol and later extended it with the introduction of Extended TACAS (XTACACS), which became a proprietary protocol.

There are some variations on Terminal Access Control Access Control System:

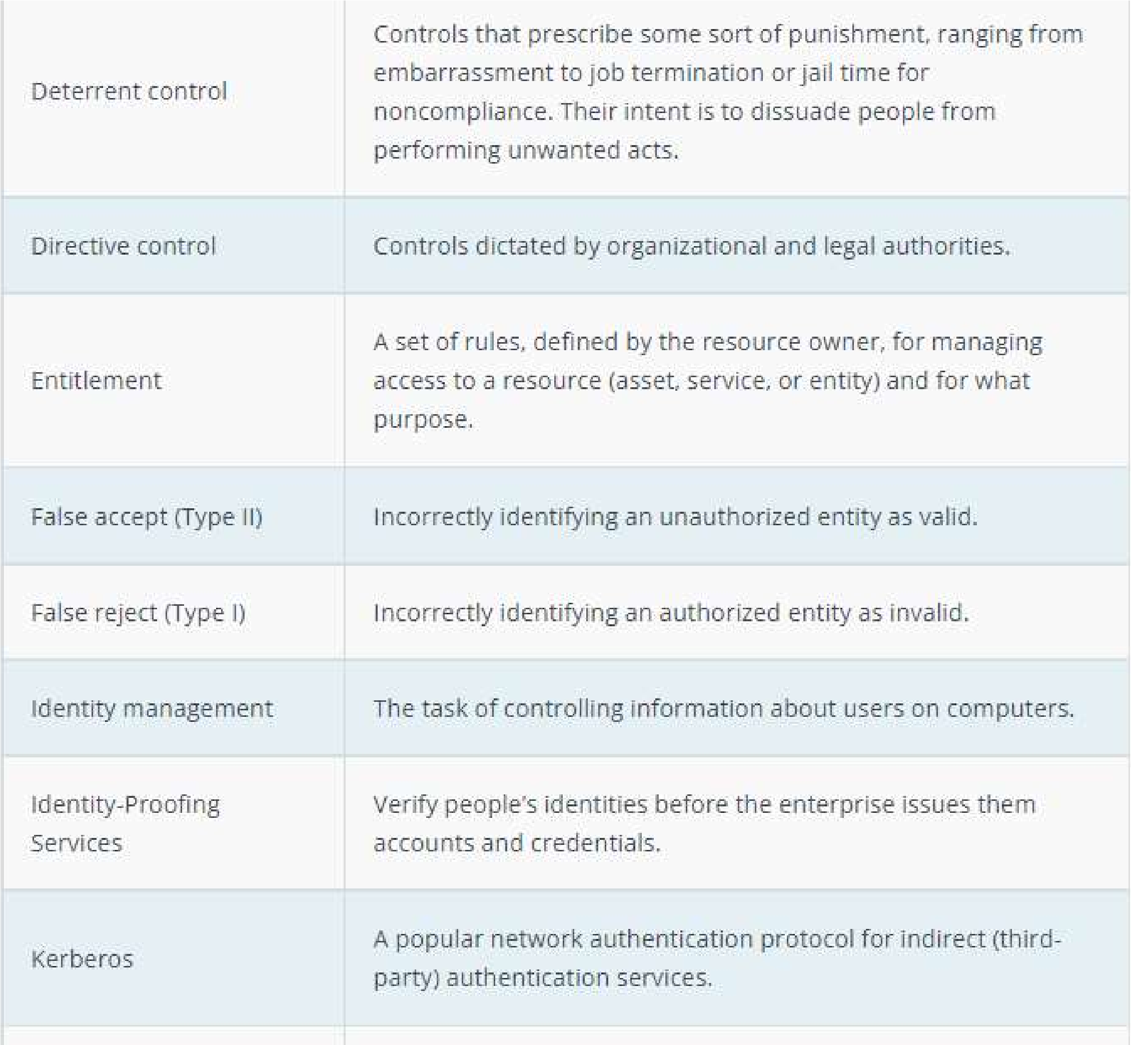
- Extended TACACS (XTACACS). There was no backward compatibility the original version.
- Terminal Access Controller Access-Control Plus (TACACS+). An entirely separate protocol, TACACS+ was released as an open standard and provided Authentication, Authorization and Accounting (AAA).

Chapter 2 Review

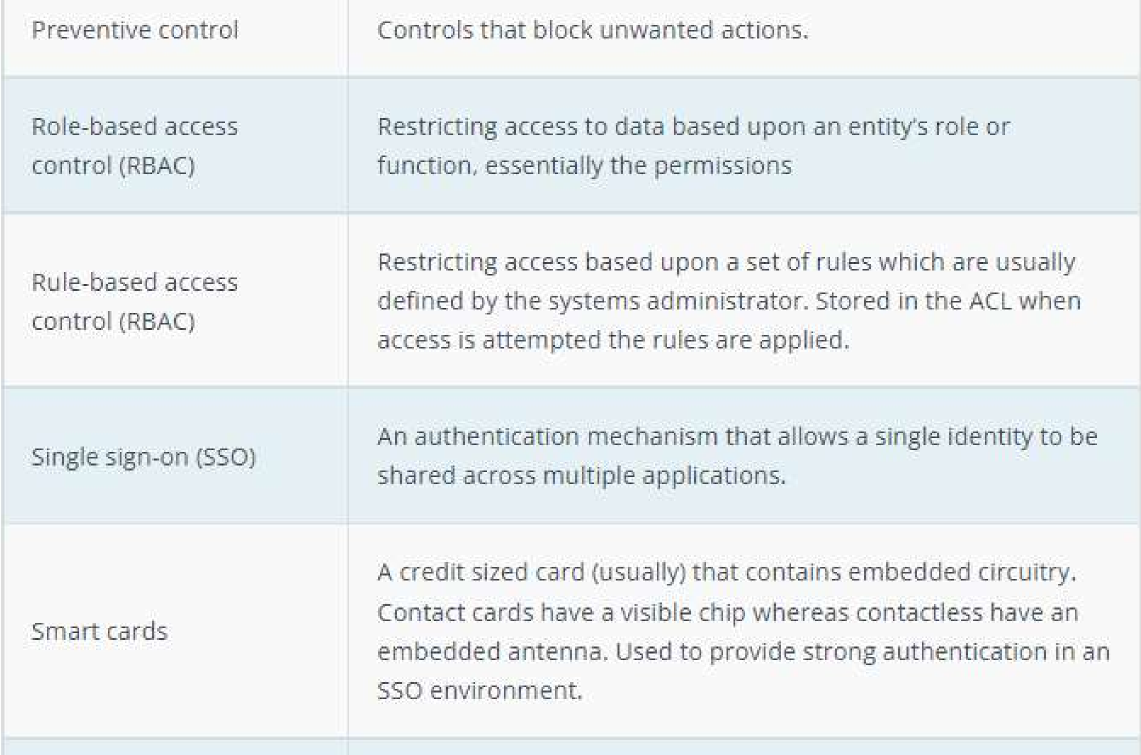
Terms and Definitions



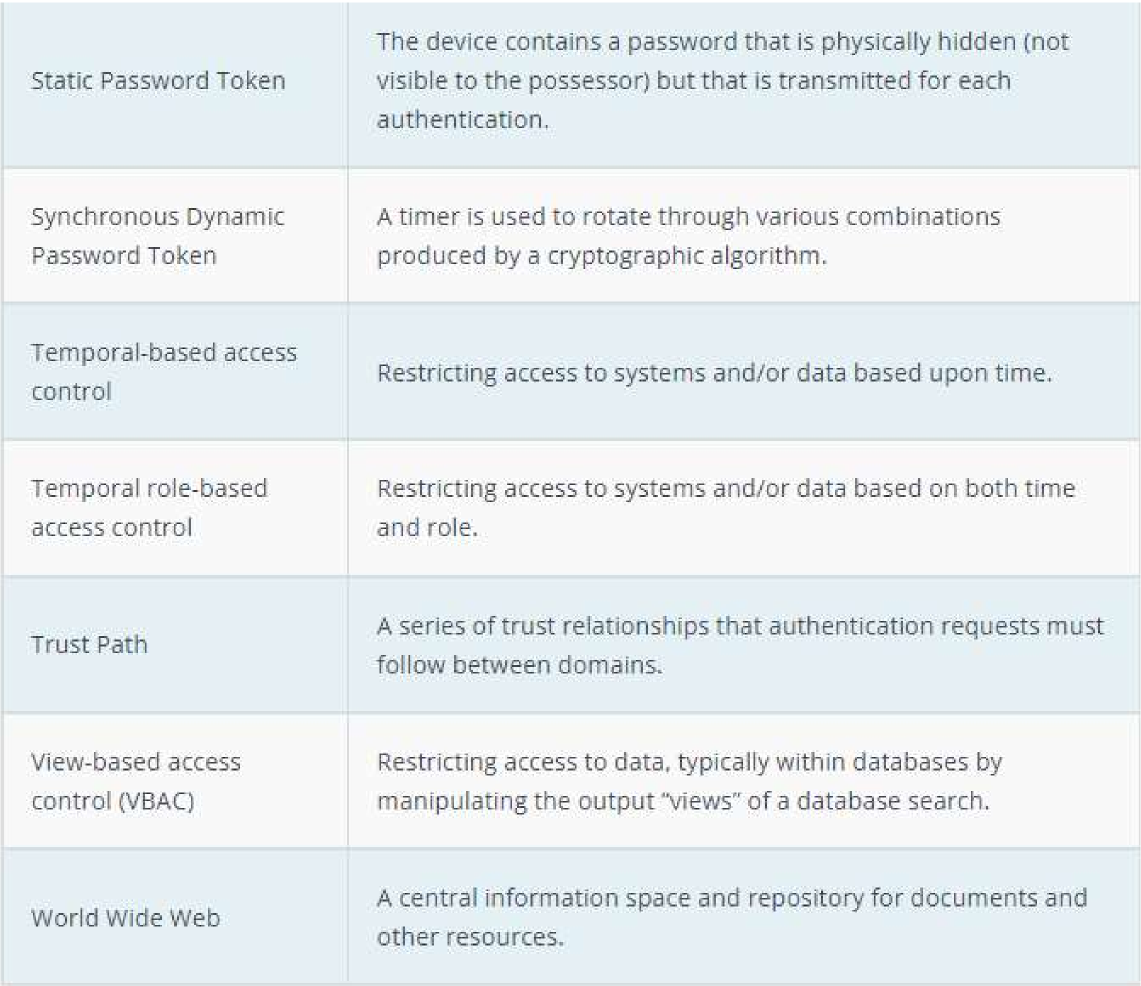
Term	Definition
Access Control Object	A passive entity that typically receives or contains some form of data.
Access Control Subject	An active entity and can be any user, program, or process that requests permission to cause data to flow from an access control object to the access control subject or between access control objects.
Asynchronous Password Token	A one-time password is generated without the use of a clock, either from a one- time pad or cryptographic algorithm.
Authorization	Determines whether a user is permitted to access a particular resource.
Connected Tokens	Must be physically connected to the computer with which the user is authenticating.
Contactless Tokens	Form a logical connection to the client computer but do not require a physical connection.
Content-Dependent Access Control (CDAC)	Access decisions are based upon the attribute value of an actual object.
Compensating controls	Introduced when the existing capabilities of a system do not support the requirements of a policy.
Constrained user interface	Limiting menus, data views, encrypting or physical constraining the user interface to restrict access to data.
Corrective control	These controls remedy the circumstances that enabled unwarranted activity, and/ or return conditions to where they were prior to the unwanted activity.
Disconnected Tokens	Have neither a physical nor logical connection to the client computer.



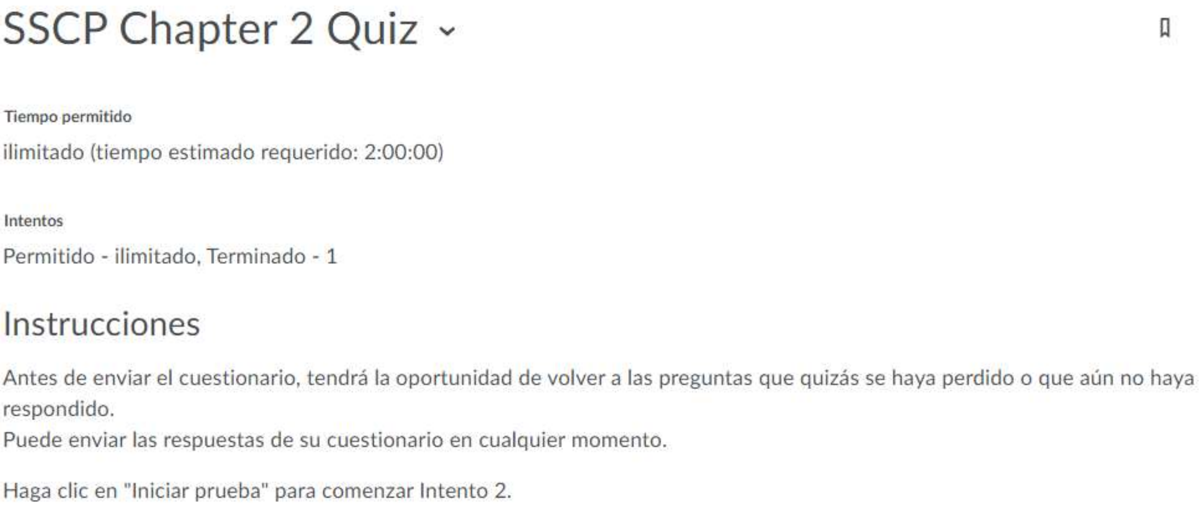
Deterrent control	Controls that prescribe some sort of punishment, ranging from embarrassment to job termination or jail time for noncompliance. Their intent is to dissuade people from performing unwanted acts.
Directive control	Controls dictated by organizational and legal authorities.
Entitlement	A set of rules, defined by the resource owner, for managing access to a resource (asset, service, or entity) and for what purpose.
False accept (Type II)	Incorrectly identifying an unauthorized entity as valid.
False reject (Type I)	Incorrectly identifying an authorized entity as invalid.
Identity management	The task of controlling information about users on computers.
Identity-Proofing Services	Verify people's identities before the enterprise issues them accounts and credentials.
Kerberos	A popular network authentication protocol for indirect (third-party) authentication services.



Preventive control	Controls that block unwanted actions.
Role-based access control (RBAC)	Restricting access to data based upon an entity's role or function, essentially the permissions
Rule-based access control (RBAC)	Restricting access based upon a set of rules which are usually defined by the systems administrator. Stored in the ACL when access is attempted the rules are applied.
Single sign-on (SSO)	An authentication mechanism that allows a single identity to be shared across multiple applications.
Smart cards	A credit sized card (usually) that contains embedded circuitry. Contact cards have a visible chip whereas contactless have an embedded antenna. Used to provide strong authentication in an SSO environment.



Static Password Token	The device contains a password that is physically hidden (not visible to the possessor) but that is transmitted for each authentication.
Synchronous Dynamic Password Token	A timer is used to rotate through various combinations produced by a cryptographic algorithm.
Temporal-based access control	Restricting access to systems and/or data based upon time.
Temporal role-based access control	Restricting access to systems and/or data based on both time and role.
Trust Path	A series of trust relationships that authentication requests must follow between domains.
View-based access control (VBAC)	Restricting access to data, typically within databases by manipulating the output "views" of a database search.
World Wide Web	A central information space and repository for documents and other resources.



SSCP Chapter 2 Quiz

Tiempo permitido

ilimitado (tiempo estimado requerido: 2:00:00)

Intentos

Permitido - ilimitado, Terminado - 1

Instrucciones

Antes de enviar el cuestionario, tendrá la oportunidad de volver a las preguntas que quizás se haya perdido o que aún no haya respondido.
Puede enviar las respuestas de su cuestionario en cualquier momento.

Haga clic en "Iniciar prueba" para comenzar Intento 2.