Lesson 1 Access Control Fundamentals

Introduction

A user first must be identified as an authorized user, such as by logging in with a user name and password to a laptop computer. Because that laptop connects to the corporate network that contains critical data, it is important also to restrict user access to only the software, hardware, and other resources for which the user has been approved. These two acts—authenticating only approved users and controlling their access to resources—are important foundations in information security.

This chapter introduces you to the principles and practices of controlling access. You will first examine access control terminology, the four standard control models, and their best practices. Then you will investigate implementing access control. Finally, you will explore authentication services, which are used to verify approved users.

Lesson Proper

Access Control

As its name implies, access control is granting or denying approval to use specific resources; it is controlling access. Physical access control consists of fencing, hardware door locks, and mantraps to limit contact with devices. In a similar way, technical access control consists of technology restrictions that limit users on computers from accessing data, it is a fundamental component of data security that dictates who's allowed to access and use company information and resources. Through authentication and authorization, access control policies make sure users are who they say they are and that they have appropriate access to company data. Access control can also be applied to limit physical access to campuses, buildings, rooms, and data centers. Access control has a set of associated terminology used to describe its actions. There are four standard access control models as well as specific practices used to enforce access control.

Access Control Terminology

A user accessing a computer system would likewise present credentials or identification, such as a user name, when logging on to the system. Checking the delivery person's credentials to be sure that they are authentic and not fabricated is authentication. Computer users, likewise, must have their credentials authenticated to ensure that they are who they claim to be, often by entering a password, fingerprint scan, or other means of authentication. Authorization, granting permission to take the action, is the next step. Likewise, once users have presented their identification and been authenticated, they can be authorized to log in to the system. computer users are granted access only to the specific services, devices, applications, and files needed in order to perform their job duties

Action	Description	Scenario example	Computer process
Identification	Review of credentials	Delivery person shows employee badge	User enters user name
Authentication	Validate credentials as genuine	Gabe reads badge to determine it is real	User provides password
Authorization	Permission granted for admittance	Gabe opens door to allow delivery person in	User authorized to log in
Access	Right given to access specific resources	Delivery person can only retrieve box by door	User allowed to access only specific data

Figure 1 Basic steps in access control

Other terminology is used to describe how computer systems impose technical access control:

- Object. An object is a specific resource, such as a file or a hardware device.
- Subject. A subject is a user or a process functioning on behalf of the user that attempts to access an object.
- Operation. The action that is taken by the subject over the object is called an operation.
  For example, a user (subject) may attempt to delete (operation) a file (object). Individuals are given different roles in relationship to access control objects or resources.
  These roles are summarized in Figure 2.

Role	Description	Duties	Example
Owner	Person responsible for the information	Determines the level of security needed for the data and delegates security duties as required	Determines that the file SALARY.XLSX can be read only by department managers
Custodian	Individual to whom day-to-day actions have been assigned by the owner	Periodically reviews security settings and maintains records of access by end-users	Sets and reviews security settings on SALARY.XLSX
End-user	User who accesses information in the course of routine job responsibilities	Follows organization's security guidelines and does not attempt to circumvent security	Opens SALARY.XLSX

Figure 2. Roles in Access Control

Access Control Models

Consider a network system administrator who needs to act as an access control custodian. One afternoon she must give a new employee access to specific servers and files. With of thousands of files scattered across a multitude of different servers, and with the new employee being given different access privileges to each file (for example, he can view one file but not edit it, and in a different file he can edit but not delete), controlling access could prove to be a daunting task. However, this job is made easy by the fact that the hardware and software have a predefined framework that the custodian can use for controlling access. This framework is called an access control model and is embedded in the software and hardware. The custodian can use the appropriate model to configure the necessary level of control. There are four major access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role Based Access Control (RBAC), and Rule Based AccessControl (RBAC).

- **Discretionary Access Control (DAC)** The Discretionary Access Control (DAC) model is the least restrictive. With the DAC model, every object has an owner, who has total control over that object. Owners can create and access their objects freely. In addition, the owner can give permissions to other subjects over these objects. For example, with DAC, Amanda could access the files EMPLOYEES.XLSX and SALARIES.XLSX as well as paste the contents of EMPLOYEES.XLSX into a newly created document MY_DATA.XLSX. She also could give Abby access to all of these files but allow Brian to only read EMPLOYEES. XLSX. DAC is used on operating systems such as most types of UNIX and Microsoft Windows. These controls can be configured so that another user can have full or limited access over a file, printer, or other object. DAC has two significant weaknesses. *First*, although it gives a degree of freedom to the subject, DAC poses risks in that it relies on decisions by the end-user to set the proper level of security. As a result, incorrect permissions might be granted to a subject or permissions might be given to an unauthorized subject. A *second weakness* is that a subject's permissions will be "inherited" by any programs that the subject executes. Attackers often take advantage of this inheritance because end-users frequently have a high level of privileges. Malware that is downloaded onto a user's computer that uses the DAC model would then run at the same high level as the user's privileges.
- **Mandatory Access Control (MAC)** The opposite of DAC is the most restrictive access control model known as Mandatory Access Control (MAC). MAC assigns users' access controls strictly according to the custodian's desires. This is considered the most restrictive access control model because the user has no freedom to set any controls. This

model is typically found in military settings in which security is of supreme importance. There are two key elements to MAC:

- **Labels.** In a system using MAC, every entity is an object (laptops, files, projects, and so on) and is assigned a classification label. These labels represent the relative importance of the object, such as confidential, secret, and top secret. Subjects (users, processes, and so on) are assigned a privilege label (sometimes called a clearance).
- **Levels.** A hierarchy based on the labels is also used, both for objects and subjects. Top secret has a higher level than secret, which has a higher level than confidential.

MAC grants permissions by matching object labels with subject labels based on their respective levels. To determine if a file can be opened by a user, the object and subject labels are compared. The subject must have an equal or greater level than the object in order to be granted access. For example, if the object label is top secret, yet the subject has only a lower secret clearance, access is denied. Subjects cannot change the labels of objects or other subjects in order to modify the security settings There are two major implementations of MAC. The first is called the lattice model. A **lattice** is a type of screen or fencing that is used as a support for climbing garden plants. Different "rungs" on the MAC lattice model have different security levels, and subjects are assigned a "rung" on the lattice just as objects are. There can even be multiple lattices placed beside each other to allow for different groups of labels. For example, one subject label lattice could use the clearances confidential, secret, and top-secret while a corresponding subject label lattice could use public, restricted, and top clearance. The rungs of each subject lattice would still align with the rungs on the object security lattice Another implementation of MAC is the **Bell-LaPadula (BLP) model.** Although this model is very similar to the lattice model, it contains an additional restriction not found in the original lattice model. This protection prevents subjects from creating a new object or performing specific functions on objects that are at a lower level than their own. For example, a user with a clearance secret should not have the ability to open a document at the secret level and then paste its contents to a newly created document at the confidential level. A variation of the BLP model is the Biba Integrity model, which goes beyond the BLP model and adds protecting data integrity in addition to confidentiality. Microsoft Windows uses a MAC implementation called Mandatory Integrity Control (MIC). Based on the Biba model, MIC ensures data integrity by controlling access to securable objects. A security identifier (SID) is a unique number issued to the user, group, or session. Each time a user logs in, the system retrieves the SID for that user from the database, and then uses that SID to identify the user in all subsequent interactions with Windows security. Windows links the SID to an integrity level. Objects such as files, processes, services, and devices are assigned integrity levels—low, medium, high, and system—that determine their levels of protection or access. In order to write to or delete an object, the integrity level of the subject must be equal to or greater than the object's level. This ensures that processes running with a low integrity level cannot write to an object with a medium integrity level. This can be seen in practice through a Window's feature known as **User Account Control (UAC).** The standard user (lower level) who attempts to install the software (higher level) is first required by UAC to enter the higher-level administrative password before being allowed to proceed (which elevates the action to the higher level). As an additional check, an administrative user also must confirm the action (yet he does not need to enter the administrative password, In this way, UAC attempts to match the subject's privilege level with that of the object.

- **Role Based Access Control (RBAC)** The third access control model is Role Based Access Control (RBAC), sometimes called Non-Discretionary Access Control. RBAC is considered a more "real-world" access control than the other models because the access under RBAC is based on a user's job function within an organization. Instead of setting permissions for each user or group, the RBAC model assigns permissions to particular roles in the organization and then assigns users to those roles. Objects are set to be a certain type, to which subjects with that particular role have access. For example, instead of creating a user account for Ahmed and assigning specific privileges to that account, the role Business_Manager can be created based on the privileges an individual in that job function should have. Then Ahmed and all other business managers in the organization can be assigned to that role. The users and objects inherit all of the permissions for the role.
- **Rule Based Access Control (RBAC)** The Rule Based Access Control (RBAC) model, also called the Rule-Based Role-Based Access Control (RB-RBAC) model or automated provisioning, can dynamically assign roles to subjects based on a set of rules defined by a custodian. Each resource object contains a set of access properties based on the rules. When a user attempts to access that resource, the system checks the rules contained in that object to determine if the access is permissible. Rule Based Access Control is often used for managing user access to one or more systems, where business changes may trigger the application of the rules that specify access changes. For example, a subject on Network A wants to access objects on Network B, which is located on the other side of a router. This router contains the set of access control rules and can assign a certain role to the user, based on her network address or protocol, which will then determine whether she will be granted access. Similar to MAC, Rule Based Access Control cannot be changed by users. All-access permissions are controlled based on rules established by the custodian or system administrator.

Best Practices for Access Control

Enforcing technical access control using the access control models is only one means of providing security. In addition, establishing a set of "best practices" for limiting access also can help secure systems and their data. These practices include separation of duties, job rotations, least privilege, implicit deny, and mandatory vacations.

- **Separation of Duties** News headlines such as "County Official Charged with Embezzlement" appear all too frequently. Often this fraud results from a single user being trusted with a set of responsibilities that place the person in complete control of the process. For example, one person may be given total control over the collection, distribution, and reconciliation of money. If no other person is involved, it may be too tempting for that person to steal, knowing that nobody else is watching and that there is a good chance the fraud will go undetected. To counteract this possibility, most organizations require that more than one person be involved with functions that relate to handling money, because it would require a conspiracy of all the individuals in order for fraud to occur. Likewise, a foundational principle of computer access control is not to give one person total control. Known as separation of duties, this practice requires that if the fraudulent application of a process could potentially result in a breach of security, the process should be divided between two or more individuals. For example, if the duties of the owner and the custodian are performed by a single individual, it could provide that person with total control over all security configurations. It is recommended that these responsibilities be divided so that the system is not vulnerable to the actions performed by a single person.
- **Job Rotation** Another way to prevent one individual from having too much control is to use job rotation. Instead of one person having sole responsibility for a function, individuals are periodically moved from one job responsibility to another. Employees can rotate either within their home department or across positions in other departments. The best rotation procedure involves multiple employees rotating across many positions for different lengths of time to gain exposure to different roles and functions.
  Job rotation has several advantages:
  - It limits the amount of time that individuals are in a position to manipulate security configurations.
  - It helps to expose any potential avenues for fraud by having multiple individuals with different perspectives learn about the job and uncover vulnerabilities that someone else may have overlooked.

- Besides enhancing security, job rotation also can reduce "burnout," increase employee satisfaction, provide a higher level of employee motivation, enhance and improve skills and competencies leading to promotional advancement, and provide an increased appreciation for peers and decreased animosity between departments.

Job rotation, however, also has disadvantages. In some cases employees may not be in a specific job long enough to develop proficiency, and productivity may be lost in the time it takes to train employees in new tasks. Also, job rotation is often limited to less specialized positions. For these reasons, job rotation may not always be practical.

- **Least Privilege** Consider the rooms in a large office building, each of which has a door with a lock. Different classifications of employees can be provided different keys to open doors based on their jobs. For example, a typical office worker would not be given a key that opens every door in the building. There simply is no need for this classification of worker to have access to the contents of every room. If that key were lost or stolen, the thief could easily enter any office at any time to remove its contents. Instead, a typical office worker would be provided only a key that opens the door to his office because that is all that is needed for the worker to do his job. A member of the building's security staff, on the other hand, would have a key that could open any office because her job function would require it. Limiting access to rooms in a building is a model of the information technology security principle of least privilege. Least privilege in access control means that only the minimum amount of privileges necessary to perform a job or function should be allocated. This helps reduce the attack surface by eliminating unnecessary privileges that could provide an avenue for an attacker. Least privilege should apply both to users and to processes running on the system. For processes, it is important that they be designed so that they run at the minimum security level needed in order to correctly function. Users also should be given only those privileges for which they need to perform their required tasks. Different options for securely providing privileges exist. For example, in Apple Mac OS X and Linux/UNIX systems, the system administrator can give specific users or groups access to higher-level commands without revealing the main root password to those users or groups. A user must simply enter the sudo (superuser do) command, which prompts the user for his personal password and confirms the request to execute a command (previously approved by the system administrator). The sudo command also logs all actions as an audit trail.Although least privilege is recognized as an important element in security, the temptation to assign higher levels of privileges is great due to the challenges of assigning users lower security levels.
- **Implicit Deny** Implicit deny in access control means that if a condition is not explicitly met, the request for access is rejected. (Implicit means that something is implied or indicated but not actually expressed.) For example, a network router may have a rule-based access control restriction. If no conditions match the restrictions, the router rejects access because of an implicit deny all clause: any action that is not explicitly permitted is denied. When creating access control restrictions, it is recommended that unless the condition is specifically met, access should be denied.
- **Mandatory Vacations** In many fraud schemes, the perpetrator must be present every day in order to continue the fraud or keep it from being exposed. Many organizations require mandatory vacations for all employees to counteract this. For sensitive positions within an organization, an audit of the employees' activities is usually scheduled while they are away on vacation.

Implementing Access Control

Several technologies can be used to implement access control. These include access control lists, Group Policy, and account restrictions.

- **Access Control Lists (ACLs)** An access control list (ACL) is a set of permissions that is attached to an object. This list specifies which subjects are allowed to access the object and what operations they can perform on it. When a subject requests to perform an operation on an object, the system checks the ACL for an approved entry in order to decide if the operation is allowed. Although ACLs can be associated with any type of object, these lists are most often viewed in relation to files maintained by the operating system. For example, a user setting permissions in a UNIX DAC operating system would use the commands setfacl and getacl l (to set and display ACL settings, respectively) The structure behind ACL tables can be complex. In the Microsoft Windows, Linux, and Mac OS X operating systems, each entry in the ACL table is known as an access control entry (ACE). In Windows, the ACE includes four items of information:
  - An SID for the user account, group account, or logon session. An SID is a unique number issued to the user, group, or session that is used to identify the user in all subsequent interactions with Windows security.
  - An access mask that specifies the access rights controlled by the ACE. An access mask is a value that specifies the rights that are allowed or denied, and is also used to request access rights when an object is opened.
  - A flag that indicates the type of ACE. This flag corresponds to a particular set of operations that can be performed on an object.
  - A set of flags that determines whether objects can inherit permissions.

Although widely used, ACLs have limitations. First, using ACLs is not efficient. The ACL for each file, process, or resource must be checked every time the resource is accessed. ACLs control not only user access to system resources but also application and system access as well. This means that in a typical computing session ACLs are checked whenever a user accesses files when applications are opened (along with the files and applications those applications open and modify), when the operating system applications perform functions, and so on. A second limitation to ACLs is that they can be difficult to manage in an enterprise setting where many users need to have different levels of access to many different resources. Selectively adding, deleting, and changing ACLs on individual files, or even groups of files can be time-consuming and open to errors, particularly if changes must be made frequently.

- **Group Policies** This is a Microsoft Windows feature that provides centralized management and configuration of computers and remote users using the Microsoft directory services A**ctive Directory (AD)**.

  Group Policy is usually used in enterprise environments to enforce access control by restricting user actions that may pose a security risk, such as changing access to certain folders or downloading executable files. Group Policy can control an object's script for logging on and off the system, folder redirection, Internet Explorer settings, and Windows Registry settings (the registry is a database that stores settings and options for the operating system).

  Group Policy settings are stored in **Group Policy Objects (GPOs).** These objects may, in turn, be linked to multiple domains or websites, which allows for multiple systems and users to be updated by a change to a single GPO. Group Policies are analyzed and applied for computers when they start up and for users when they log in. Every 1 to 2 hours, the system looks for changes in the GPO and reapplies them as necessary.

  A **Local Group Policy (LGP)** has fewer options than a Group Policy. Generally, an LGP is used to configure settings for systems that are not part of Active Directory. Although older versions of Windows using LGP could not be used to apply policies to individual users or groups of users, recent Windows versions support multiple Local Group Policy objects, which allows setting Local Group Policy for individual users.

Reviewer in Data and Application Security

Account Restrictions

Another means of enforcing access control is to place restrictions on user accounts. Two common account restrictions are time-of-day restrictions and account expiration.

- **Time-of-Day Restrictions** Time-of-day restrictions can be used to limit when a user can log in to a system or access resources. In addition to time-of-day restrictions, some programs can also even restrict what websites are viewed and which programs are used by specific users. When setting these restrictions, a custodian would typically indicate the times a user is restricted from accessing the system or resources.
- **Account Expiration** Orphaned accounts are user accounts that remain active after an employee has left an organization, while a dormant account is one that has not been accessed for a lengthy period of time. These types of accounts can be a security risk. For example, an employee who left under unfavorable circumstances may be tempted to "get even" with the organization by stealing or erasing sensitive information through her account. Dormant accounts that are left unchecked can provide an avenue for an attacker to exploit without the fear of the actual user or a system administrator noticing. Several recommendations for dealing with orphaned or dormant accounts include:
  - ***Establish a formal process.*** It is important that a formal procedure be in place for disabling accounts for employees who are dismissed or resign from the organization.
  - ***Terminate access immediately.*** It is critical that access be ended as soon as the employee is no longer part of the organization.
  - ***Monitor logs.*** Current employees are sometimes tempted to use an older dormant account instead of their own account. Monitoring logs can help prevent the use of other
    accounts.

Locating and terminating orphaned and dormant accounts, however, still remains a problem for many organizations. To assist with controlling orphaned and dormant accounts, account expiration can be used. Account expiration is the process of setting a user's account to expire. Account expiration is not the same as password expiration. Account expiration indicates when an account is no longer active; password expiration sets the time when a user must create a new password in order to access his account. Account expiration can be explicit, in that the account expires on a set date, or it can be based on a specific number of days of inactivity. For example, in a Linux or UNIX system, when an account is created, an option allows for a set number of days after a password has expired before the account itself will be disabled.

Authentication Services

A user accessing a computer system must present credentials or identification when logging in to the system. Verifying the person's credentials to be sure that they are genuine and the user actually is who she claims to be is the process of authentication. Authentication services can be provided on a network by a dedicated authentication, authorization, and accounting (AAA) server or by an authentication server, which is a server that performs the only authentication. The most common type of authentication and AAA servers are RADIUS, Kerberos, Terminal Access Control Systems (TACACS), generic servers built on the Lightweight Directory Access Protocol (LDAP), and Security Assertion Markup Language (SAML).

- **RADIUS, or Remote Authentication Dial-In User Service**, was developed in 1992 and quickly became the industry standard with widespread support across nearly all vendors of networking equipment. RADIUS was originally designed for remote dial-in access to a corporate network. However, the word Remote in the name RADIUS is now almost a misnomer because RADIUS authentication is used for more than connecting to remote networks. With the development of IEEE 802.1x port security for both wired and wireless LANs, RADIUS has seen even greater usage. A RADIUS client is not the device requesting authentication, such as a desktop system or wireless notebook computer. Instead, a RADIUS client is typically a device such as a wireless access point (AP) or the dial-up server that is responsible for sending user credentials and connection parameters in the form of a RADIUS message to a RADIUS server. The RADIUS server authenticates and authorizes the RADIUS client request, and sends back a RADIUS message response. RADIUS clients also send RADIUS accounting messages to RADIUS servers. The strength of RADIUS is that messages are never sent directly between the wireless device and
  the RADIUS server. This prevents an attacker from penetrating the RADIUS server and compromising security. The detailed steps for RADIUS authentication with a wireless device in an IEEE 802.1xnetwork, are:
  - A wireless device called the supplicant (it makes an "appeal" for access), sends a request to an AP requesting permission to join the WLAN. The AP prompts the user for the user ID and password.
  - The AP, serving as the authenticator that will accept or reject the wireless device, creates a data packet from this information called the authentication request. This packet includes information such as identification of the specific AP that is sending the authentication request and the user name and password. For protection from eavesdropping, the AP (acting as a RADIUS client) encrypts the password before it is sent to the RADIUS server. The authentication request is sent over the network from the AP to the RADIUS server. This communication can be done over either a local area network or a wide area network. This allows the RADIUS clients to be remotely located from the RADIUS server. If the RADIUS server cannot be reached, the AP can usually route the request to an alternate server.
  - When an authentication request is received, the RADIUS server validates that the request is from an approved AP and then decrypts the data packet to access the user name and password information. This information is passed on to the appropriate security user database. This could be a text file, UNIX password file, a commercially available security system, or a custom database.
  - If the user name and password are correct, the RADIUS server sends an authentication acknowledgment that includes information on the user's network system and service requirements. For example, the RADIUS server may tell the AP that the user needs TCP/IP. The acknowledgment can even contain filtering information to limit a user's access to specific resources on the network. If the user name and password are not correct, the RADIUS server sends an authentication reject message to the AP and the user is denied access to the network. To ensure that requests are not responded to by unauthorized persons or devices on the network, the RADIUS server sends an authentication key, or signature, identifying itself to the RADIUS client.
  - If accounting is also supported by the RADIUS server, an entry is started in the accounting database.
  - Once the server information is received and verified by the AP, it enables the necessary configuration to deliver the wireless services to the user.

RADIUS allows an organization to maintain user profiles in a central database that all remote servers can share. Doing so increases security, allowing a company to set up a that can be applied at a single administered network point. Having a central service also means that it is easier to track usage for billing and for keeping network statistics. policy

- **Kerberos** is an authentication system developed by the Massachusetts Institute of Technology (MIT) in the 1980s and used to verify the identity of networked users. Named after a three-headed dog in Greek mythology that guarded the gates of Hades, Kerberos uses encryption and authentication for security. Kerberos will function under Windows, Apple Mac OS X, and Linux. Kerberos has often been compared to using a driver's license to cash a check. A state agency, such as the Department of Motor Vehicles (DMV), issues a driver's license that has these characteristics:

- It is difficult to copy.
- It contains specific information (name, address, weight, height, etc.).
- It lists restrictions (must wear corrective lenses, etc.).
- It will expire at some future date.

Kerberos, which works in a similar fashion, is typically used when a user attempts to access a network service and that service requires authentication. The user is provided a ticket that is issued by the Kerberos authentication server, much as a driver's license is issued by the DMV. This ticket contains information linking it to the user. The user presents this ticket to the network for a service. The service then examines the ticket to verify the identity of the user. If the user is verified, he is then accepted. Kerberos tickets share some of the same characteristics as a driver's license: tickets are difficult to copy (because they are encrypted), they contain specific user information, they restrict what a user can do, and they expire after a few hours or a day. Issuing and submitting tickets in a Kerberos system is handled internally and is transparent to the user.

- **Terminal Access Control Access Control System (TACACS)** Similar to RADIUS, Terminal Access Control Access Control System (TACACS) is an authentication service commonly used on UNIX devices that communicates by forwarding user authentication information to a centralized server. The centralized server can be either a TACACS database or a database such as a Linux or UNIX password file with TACACS protocol support. The first version was simply called TACACS, while a later version introduced in 1990 was known as Extended TACACS (XTACACS). The current version is TACACS+.
- **Lightweight Directory Access Protocol (LDAP)** A directory service is a database stored on the network itself that contains information about users and network devices. It contains information such as the user's name, telephone extension, email address, login name, and other facts. The directory service also keeps track of all the resources on the network and a user's privileges to those resources, and grants or denies access based on the directory service information. Directory services make it much easier to grant privileges or permissions to network users. The International Organization for Standardization (ISO) created a standard for directory services known as X.500. The purpose of the X.500 standard was to standardize how the data was stored so that any computer system could access these directories. It provides the capability to look up information by name (a white-pages service) and to browse and search for information by category (a yellow-pages service). The information is held in a directory information base (DIB). Entries in the DIB are arranged in a tree structure called the directory information tree (DIT). Each entry is a named object and consists of a set of attributes. Each attribute has a defined attribute type and one or more values. The directory defines the mandatory and optional attributes for each class of object. Each named object may have one or more object classes associated with it. The X.500 standard defines a protocol for a client application to access an X.500 directory called the Directory Access Protocol (DAP). However, the DAP is too large to run on a personal computer. The Lightweight Directory Access Protocol (LDAP), sometimes called X.500 Lite, is a simpler subset of DAP. The primary differences between DAP and LDAP are:

    - Unlike X.500 DAP, LDAP was designed to run over TCP/IP, making it ideal for Internet and intranet applications. X.500 DAP requires special software to access the network.
    - LDAP has simpler functions, making it easier and less expensive to implement.
    - LDAP encodes its protocol elements in a less complex way than X.500 that enables it to streamline requests.

If the information requested is not contained in the directory, DAP only returns an error to the client requesting the information, which must then issue a new search request. By contrast, LDAP servers return only results, making the distributed X.500 servers appear as a single logical directory. By default LDAP traffic is transmitted in cleartext. LDAP traffic can be made secure by using Secure Sockets Layer (SSL) or Transport Layer Security (TLS). This is known as Secure LDAP or LDAP over SSL (LDAPS). LDAP makes it possible for almost any application running on virtually any computer platform to obtain directory information. Because LDAP is an open protocol, applications need not worry about the type of server hosting the directory. Today many LDAP servers are implemented using standard relational database management systems as the engine, and communicate via the Extensible Markup Language (XML) documents served over the hypertext transport protocol (HTTP). However, a weakness of LDAP is that it can be subject to LDAP injection attacks. These attacks, similar to SQL injection attacks, can occur when user input is not properly filtered. This may allow an attacker to construct LDAP statements based on user input statements. The attacker could then retrieve information from the LDAP database or modify its content. The defense against LDAP injection attacks is to examine all user input before processing.

- **Security Assertion Markup Language (SAML)** is an Extensible Markup Language (XML) standard that allows secure web domains to exchange user authentication and authorization data. This allows a user's login credentials to be stored with a single identity provider instead of being stored on each web service provider's server. SAML is used extensively for online e-commerce business-to-business (B2B) and business-to-consumer (B2C) transactions.

The steps of a SAML transaction, are:

1. The user attempts to reach a website of a service provider that requires a username and password.

2. The service provider generates a SAML authentication request that is then encoded and embedded into a URL.

3. The service provider sends a redirect URL to the user's browser that includes the encoded SAML authentication request, which is then sent to the identity provider.

4. The identity provider decodes the SAML request and extracts the embedded URL. The identity provider then attempts to authenticate the user either by asking for login
credentials or by checking for valid session cookies.

5. The identity provider generates a SAML response that contains the authenticated user's username, which is then digitally signed using asymmetric cryptography.

6. The identity partner encodes the SAML response and returns that information to the user's browser.

7. Within the SAML response, there is a mechanism so that the user's browser can forward that information back to the service provider, either by displaying a form that requires the user to click on a Submit button or by automatically sending to the service provider.

8. The service provider verifies the SAML response by using the identity provider's public key. If the response is successfully verified, the user is logged in.

Lesson Summary

Best practices for implementing access control include separation of duties (dividing a process between two or more individuals), job rotation (periodically moving workers from one job responsibility to another), using the principle of least privilege (giving users only the minimal amount of privileges necessary in order to perform their job functions), using implicit deny (rejecting access unless it is specifically granted), and mandatory vacations (requiring that employees take periodic vacations). Implementing access control methods include using access control lists (ACLs), which are provisions attached to an object. ACLs define which subjects are allowed to access which objects and specify which operations they can perform. Group Policy is a Microsoft Windows feature that

provides centralized management and the configuration of computers that use Active Directory. Time of day restrictions limit when a user can log into a system or access resources. Account expiration specifies when a user's account expires.

Authentication services can be provided on a network by a dedicated AAA or authentication server. RADIUS, or Remote Authentication Dial In User Service, has become the industry standard with widespread support across nearly all vendors of networking equipment. The strength of RADIUS is that messages are never directly sent between the wireless device and the RADIUS server. This prevents an attacker from penetrating the RADIUS server and compromising security. Kerberos is an authentication system used to verify the identity of networked users. Similar to RADIUS, Terminal Access Control Access Control System (TACACS), XTACACS, and TACACS+ are protocol specifications that forward user name and password information to a centralized server.

Lesson 2 Authentication and Account Management 1

Introduction

Restricting a user to just his office is similar to the concept of access control, the actions of the night security guard to verify users' identity parallel the act of authentication in information security. Authentication is the process of ensuring that the person desiring access to resources is authentic, and not an imposter.

In this chapter we will study authentication and the secure management of user accounts that enforces authentication. First you will look at the different types of authentication credentials that can be used to verify a user's identity. Next you will see how a single sign-on might be used. Finally, you will look into the techniques and technology used to manage user accounts in a secure fashion.

Lesson Proper

Authentication Credentials

Consider this scenario: Ermanno works on a local military base. Each afternoon he stops at the gym on the base to exercise. After Ermanno locks his car, he walks into the club and is recognized by Li, the clerk at the desk. Li congratulates Ermanno for winning the recent competition for doing the most pushups in one minute. She then allows him to pass on to the locker room. Once inside the locker room, Ermanno opens his locker's combination lock with a series of numbers that he has memorized. While he is exercising, Kristen walks over to Ermanno and says, "I knew it was you doing those pushups even though I could not see your face. Nobody comes close to doing as many as you can. Congratulations on winning the trophy." In this scenario, Ermanno has been demonstrated to be genuine or authentic, and not an imposter, by five separate elements.

1. Somewhere he is. Because the military base is surrounded by fencing and guards, an imposter Ermanno would not be approved to enter the base. This means that the location of Ermanno can help prove his authenticity.

2. Something he has. By locking the doors of his car with his car's wireless key fob, an item that only the real Ermanno would possess, what he has helped to prove his genuineness.
3. Something he is. Access to the locker room is protected by what Ermanno is. Li has to recognize his unique characteristics (his hair color, face, body type, voice, etc.) before he will be allowed to enter the locker room, so these characteristics serve to confirm his authenticity.
4. Something he knows. The contents of Ermanno's locker are protected by what only the real Ermanno knows, namely, the lock combination. The lock will not open for an imposter, but only for the real Ermanno who knows the combination.
5. Something he does. Because only Ermanno is able to do the record number of pushups, what he does helps to uniquely prove his authenticity

Because only the real or "authentic" Ermanno possesses these elements—where he is, what he has, what he is, what he knows, and what he does—they can be considered as types of authentication or proof of his genuineness. This authentication confirms his identity and can be used to protect his belongings by preventing access by an imposter.

In information technology (IT), these five elements are known as authentication factors (sometimes called authentication credentials). Although there are many different authentication credentials that can be presented to an IT system in order to verify the genuineness of the user, all credentials can be classified into one of these five categories.

What You Know: Passwords

In most systems, a user logging in would be asked to identify himself. This is done by entering an identifier known as the username, such as F_McGee. Yet because anyone could enter this username, the next step is for the user to authenticate himself by proving that he actually is F_McGee. This is often done by providing information that only he would know, namely, a password. A password is a secret combination of letters, numbers, and/or characters that only the user should have knowledge of. Passwords are the most common type of authentication today.

Despite their widespread use, passwords provide only weak protection. Although there are several different attacks that can be launched against passwords, actions can be taken to strengthen passwords.

Password Weaknesses

The weakness of passwords centers on **human memory**. Human beings can memorize only a limited number of items. Passwords place heavy loads on human memory in multiple ways:

1. The most effective passwords are long and complex. However, these are difficult for users to memorize and then accurately recall when needed.
2. Users must remember passwords for many different accounts. Most users have accounts for different computers and mobile devices at work, school, and home; multiple email accounts; online banking; Internet site accounts; and so on. In one study, 28 percent of a group of users had more than 13 passwords each, while in another study a group of 144 users had an average of 16 passwords per user.

3. For the highest level of security, each account password should be unique, which further strains human memory.

4. Many security policies mandate that passwords expire after a set period of time, such as every 45–60 days, when a new one must be created. Some security policies even prevent a previously used password from being recycled and used again, forcing users to repeatedly memorize new passwords.

Because of the burdens that passwords place on human memory, users take shortcuts to help them memorize and recall their passwords. The first shortcut is to use a weak password. Weak passwords use a common word as a password (princess), a short password (desk), a predictable sequence of characters (abc123), or personal information (Hannah) in a password. Even when users attempt to create stronger passwords, they generally follow predictable patterns of appending and replacing:

- ***Appending.*** When users combine letters, numbers, and punctuation (character sets), they do it in a pattern. Users typically append one character set with another set or sets. Most often they only add a number after letters (caitlin1 or cheer99). If

they add all three character sets, it is in the sequence letters+punctuation+number (amanda.7 or chris#6).

- ***Replacing.*** Users also use replacements in predictable patterns. Generally, a zero is used instead of the letter o (passw0rd), the digit 1 for the letter i (denn1s), or a dollar sign for an s (be$tfriend).

Attackers are aware of these patterns in passwords and can search for them, dramatically weakening passwords and make it easier for attackers to crack them. Another common shortcut is to reuse the same password for multiple accounts. Although this makes it easier for the user, it also makes it easier for an attacker who compromises one account to access other accounts. The alarming use of weak passwords can be easily illustrated. Several recent attacks have stolen tens of millions of passwords, which later were posted on the Internet. An analysis of one theft of 32 million user passwords showed that 30 percent of users had created passwords of only five or six characters, while just 12 percent of the user passwords were a stronger nine characters in length. Almost one in every five users created a password that was one of the 5000 most common passwords, including names, slang words, dictionary words, or trivial passwords (consecutive digits, adjacent keyboard keys, etc.). The 10 most common passwords found and their number of occurrences are listed in Table 12-1.

Rank	Password	Number of users with password
1	123456	290,731
2	12345	79,078
3	123456789	76,790
4	Password	61,958
5	iloveyou	51,622
6	princess	35,231
7	rockyou	22,588
8	1234567	21,726
9	12345678	20,553
10	abc123	17,542

Attacks on Passwords

Most average users think that passwords are compromised by an attacker guessing a password by typing different variations. Although it may be possible for an attacker to enter different passwords at the login prompt to attempt to guess a password, in reality this is not practical. Even at two or three tries per second, it could take thousands of years to guess the right password. In addition, most accounts can be set to disable all logins after a limited number of incorrect attempts (such as five), thus locking out the attacker. Instead of randomly guessing a password, attackers use far more sophisticated methods. Attacks that can be used to discover a password include:

- ***Social engineering.*** Passwords can be revealed through social engineering attacks, including phishing, shoulder surfing, and dumpster diving.
- ***Capturing.*** There are several methods that can be used to capture passwords. A keylogger on a computer can capture the passwords that are entered on the keyboard. While passwords are in transit, man-in-the-middle and replay attacks can be used. A protocol analyzer also can capture transmissions that contain passwords.
- **Resetting**. If an attacker can gain physical access to a user's computer, she can erase the existing password and reset it to a new password. Password reset programs require that the computer be rebooted from an optical drive or USB flash drive that usually contains a version of a different operating system along with the password reset program. For example, to reset a password on a Microsoft Windows computer, a USB flash drive with Linux and the password reset program would be used.

These attacks, however, have their limitations, such as the need to physically access a user's computer or watch the user enter a password. Most password attacks today instead use offline cracking. When a password is created, a one-way hash algorithm creates a unique digital fingerprint digest (sometimes called a message digest or hash) of the password. This digest is then stored instead of the original cleartext password. When a user attempts to log in, she enters her password and a digest is then created from it. The two digests are compared, and if they match, the user is authenticated.

With offline cracking, attackers steal the file of password digests and load that file onto their own computers. They can then attempt to discover the passwords by comparing the stolen digests with their own digests that they have created, called candidates. Several offline cracking techniques attempt to match a known password digest with stolen digests. These are brute force, dictionary, hybrid, rainbow tables, and password collections.

- **Brute Force** In an automated brute force attack, every possible combination of letters, numbers, and characters are used to create candidate digests that are then matched against those in the stolen digest file. This is the slowest yet most thorough method. Using an automated brute force attack program, an attacker enters into the attack program the following types of parameters:
  - **Password length.** The minimum and maximum lengths of the passwords to be generated (such as a range from 1–15) can be entered.
  - **Character set.** This is the set of letters, symbols, and characters that make up the password. Because not all systems accept the same character set for passwords, if characters can be eliminated from the character set, this will dramatically increase the speed of the attack.
  - **Language.** Many programs allow different languages to be chosen, such as Arabic, Dutch, English, French, German, Italian, Portuguese, Russian, or Spanish.
  - **Pattern.** If any part of the password is known, a pattern can be entered to reduce the number of passwords generated. A question mark (?) can replace one symbol and an asterisk (*) can replace multiple symbols. For example, if the first two letters of a sixcharacter password were known to be sk, the pattern could be sk????.
  - **Skips.** Because most passwords are wordlike combinations of letters, some brute force attack programs can be set to skip nonsensical combinations of characters (wqrghea) so that only passwords such as elmosworld and carkeys are created.

- **Dictionary Attack** Another common password attack is a dictionary attack. A dictionary attack begins with the attacker creating digests of common dictionary words as candidates and then comparing them against those in a stolen digest file. Dictionary attacks can be successful because users often create passwords that are simple dictionary words. A dictionary attack that uses a set of dictionary words and compares it with the stolen digests is known as a pre-image attack, in that one known digest (dictionary word) is compared to an unknown digest (stolen digest). A birthday attack is slightly different, in that the search is for any two digests that are the same.

- **Hybrid Attack** A variation of the dictionary attack is the hybrid attack. This attack combines a dictionary attack with a brute force attack and will slightly alter dictionary words by adding numbers to the end of the password, spelling words backward, slightly misspelling words, or including special characters such as @, $, !, or %. Rainbow Tables Although brute force and dictionary attacks were once the primary tools used by attackers to crack stolen digest passwords, more recently attackers have used rainbow tables. Rainbow tables make password attacks easier by creating

a large pregenerated data set of candidate digests. There are two steps in using a rainbow table. First is creating the table itself. Next, that table is used to crack a password. A rainbow table is a compressed representation of cleartext passwords that are related and organized in a sequence (called a chain). To create a rainbow table, each chain begins with an initial password that is hashed and then fed into a function that produces a different cleartext password. This process is repeated for a set number of rounds. The initial password and the last digest value of the chain comprise a rainbow table entry. Using a rainbow table to crack a password also requires two steps. First, the password to be broken is hashed and run through the same procedure used to create the initial table. This results in the initial password of the chain. Then the process is repeated, starting with this initial password until the original digest is found. The password used at the last iteration is the cracked password.
Although generating a rainbow table requires a significant amount of time, once it is created it has three significant advantages over other password attack methods:

- A rainbow table can be used repeatedly for attacks on other passwords.
- Rainbow tables are much faster than dictionary attacks.
- The amount of memory needed on the attacking machine is greatly reduced.

- **Password Collections** A watershed moment in password attacks occurred in late 2009. An attacker using an SQL injection attack broke into a server belonging to a developer of several popular social media applications. This server contained more than 32 million user passwords, all in cleartext. These passwords were later posted on the Internet Attackers seized this opportunity to examine actual user passwords. These passwords provided two key elements for password attacks. First, this "treasure-trove" collection of passwords gave attackers, for the first time, a large corpus of real-world passwords. Because users repeat their passwords on multiple accounts, attackers could now use these passwords as candidate passwords in their attacks. It is estimated that in excess of 100 million passwords were stolen and published online in one year alone. Websites now host lists of these leaked passwords along with statistical analysis that attackers can utilize. In addition, these password collections have provided attackers insight into the strategic thinking of how users create passwords. For example, on those occasions when users mix uppercase and lowercase in passwords, users tend to capitalize at the beginning of the password, much like writing a sentence. Likewise, punctuation and numbers are more likely to appear at the end of the password, again mimicking standard sentence writing. And a high percentage of passwords were comprised of a name and date, such as Braden2008. Such insights can be valuable to attackers in designing a "mask" (such as ?dabcdef -2 ?l?u ?1?1? 2?2?2?2?2) to crack passwords. Password mask attacks can significantly reduce the amount of time needed to break a password when compared to a raw brute force attack.

Password Defenses

There are four primary defenses against password attacks. These include password complexity, credential management, password hashing algorithms, and salts.

- **Password Complexity** One insight into creating complex and strong passwords is to examine how a password attack program attempts to break a password.6 Most passwords consist of a root (not necessarily a dictionary word but generally "pronounceable") along with an attachment, either an ending suffix (about 90 percent of the time) or a prefix (10 percent). An attack program will first test the password against 1000 common passwords (such as 123456, password1, and letmein). If it is not successful, it then combines these common passwords with 100 common suffixes (such as 1, 4u, and abc). This results in almost 100,000 different combinations that can crack 25 percent of all passwords. Next the program (in order) uses 5000 common dictionary words, 10,000 names, 100,000 comprehensive dictionary words, and combinations from a phonetic pattern dictionary, varying the dictionary words between lowercase (the most common), initial uppercase (the second most common), all uppercase, and then final character as uppercase. The program also makes common substitutions with letters in the dictionary words, such as $ for s, @ for a, 3 for E, etc. Finally, it uses a variation of attachments, such as:

  - Two-digit combinations
  - Dates from 1900 to the present
  - Three-digit combinations
  - Single symbols (#, $, %)
  - Single digit plus single symbol
  - Two-symbol combinations

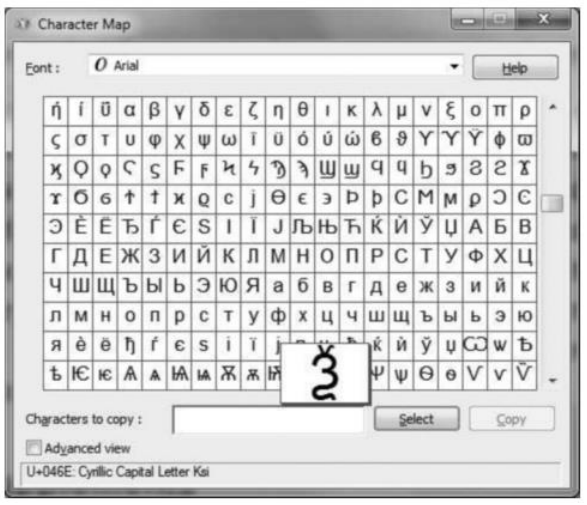
Understanding how a password attack program attempts to break a password can lead to the following general observations regarding creating passwords:

- Do not use passwords that consist of dictionary words or phonetic words.
- Do not repeat characters (xxx) or use sequences (abc, 123, qwerty).
- Do not use birthdays, family member names, pet names, addresses, or any personal information.
- Do not use short passwords. A strong password should be a minimum of 15 characters in length.

A longer password is always more secure than a shorter password because the longer a password is, the more attempts an attacker must make in order to determine it. The formula for determining the number of possible passwords given the number of characters that can be used in the password and the password length is Number-of-Keyboard-Keys ^ Password-Length = Total-Number-of-Possible-Passwords. Longer passwords force attackers to spend significantly more time attempting to break them.

Keyboard keys	Password length	Number of possible passwords
95	2	9025
95	3	857,375
95	4	81,450,625
95	5	7,737,809,375
95	6	735,091,890,625

One way to make passwords stronger is to use nonkeyboard characters, or special characters that do not appear on the keyboard, thus extending the number of possible keys beyond 95. These characters are created by holding down the ALT key while simultaneously typing a number on the numeric keypad (but not the numbers across the top of the keyboard). For example, ALT + 0163 produces the £ symbol. A list of all the available nonkeyboard characters can be seen by clicking Start and entering charmap.exe, and then clicking on a character. The code ALT + 0xxx will appear in the lower-left corner of the screen (if that character can be reproduced in Windows). Figure blow shows a Windows character map.



- **Credential Management** Equally important to creating good passwords is to properly manage password credentials. For an organization, one important defense against password cracking is to prevent attackers from capturing the password digest files. There are several defenses against the theft of these files:

Reviewer in Data and Application Security

- Do not leave a computer running unattended, even if it is in a locked office. All screensavers should be set to resume only when a password is entered.
- Do not set a computer to boot from an optical drive or USB flash drive.
- Password-protect the ROM BIOS.
- Physically lock the computer case so that it cannot be opened.
- Good credential management also includes the following:
    - ❖ Change passwords frequently.
    - ❖ Do not reuse old passwords.
    - ❖ Never write a password down.
    - ❖ Have a unique password for each account.
    - ❖ If it is necessary for a user to access another user's account, a temporary password should be set up and then immediately changed.
    - ❖ Do not allow a computer to automatically sign into an account or record a password so that a login is not necessary.
    - ❖ Do not enter passwords on public access computers or other individuals' computers that could be infected.
    - ❖ Do not enter a password while using an unencrypted wireless network.

A secure solution to credential management is to rely on technology rather than human memory to store and manage passwords. Modern web browsers contain a function that allows a user to save a password that has been entered while using the browser or through a separate dialog box that "pops up" over the browser. Browser-based solutions, however, have disadvantages. Users are restricted to using the computer that has that password information previously stored, they must avoid clearing the passwords from the computer, and the passwords may be vulnerable if another user is allowed access to the computer.

A better solution is password management applications. These programs let a user create and store multiple strong passwords in a single user "vault" file that is protected by one strong master password. Users can retrieve individual passwords as needed by opening the user file, thus freeing the user from the need to memorize multiple passwords. Yet most password management applications are more than a password-protected list of passwords. Many of these applications also include drag-and-drop capabilities, enhanced encryption, in-memory protection that prevents the operating system cache from being exposed to reveal retrieved passwords, and timed Clipboard clearing. Some password management applications can even require that a specific key file be present (such as on a USB flash drive) in addition to entering the master password to open the vault. This means that if the vault file was stolen, it could not be opened.

- **Password Hashing** Algorithms Although passwords are hashed before being stored, not all hash algorithms for passwords are considered equal. Microsoft Windows operating systems hash passwords in two ways. The first is known as the LM (LAN Manager) hash. The LM hash is not actually a hash, because a hash is a mathematical function used to fingerprint the data. The LM hash instead uses a cryptographic one-way function (OWF): instead of encrypting the password with another key, the password itself is the key. The LM hash is considered to be a very weak function for storing passwords. First, the LM hash is not case sensitive, meaning that there is no difference between uppercase (A) and lowercase (a). This significantly reduces the character set that an attacker must use. Second, the LM hash splits all passwords into two 7-character parts. If the original password is fewer than 14 characters, it simply pads the parts; if it is longer, the extra characters are dropped. This means that an attacker attempting to break an LM hash must break only two 7-character passwords from a limited character set. To address the security issues in the LM hash, Microsoft later introduced the NTLM (New Technology LAN Manager) hash. Unlike the LM hash, the NTLM hash does not limit stored passwords to two 7-character parts. In addition, it is case sensitive and has a larger character set of 65,535 characters. The original version of NTLM uses a weak cryptographic function and does not support more recent cryptographic methods; Microsoft recommends that it should not be used. The current version is NTLMv2 and uses the Hashed Message Authentication Code (HMAC) with MD5.

Using general-purpose hash algorithms like MD5 and SHA, however, is not considered secure for creating digests because these hashing algorithms are designed to create a digest as quickly as possible. The fast speed of general-purpose hash algorithms works in an attacker's favor. When an attacker is creating candidate digests, a general-purpose hashing algorithm can rapidly create a very large number of passwords for matching purposes. A more secure approach for creating password digests is to use a specialized password hash algorithm that is intentionally designed to be slower. This would then limit the ability of an attacker to crack passwords because it requires significantly more time to create each candidate digest, thus slowing down the entire cracking process. This is called key stretching.

Two popular key stretching password hash algorithms are bcrypt and PBKDF2. These can be configured to require more time to create a digest. A network administrator can specify the number of iterations (rounds), which sets how "expensive" (in terms of computer time
and/or resources) the password hash function will be. Whereas the increased time is a minor inconvenience when one user logs in and waits for the password digest to be generated, it can significantly reduce attackers' speed of generating candidates.

- **Salts** In order to increase the strength of hashed passwords, a salt also can be used. A salt consists of a random string that is used in hash algorithms. Passwords can be protected by adding a random string to the user's cleartext password before it is hashed.

Username	Password	Unsalted password hash	Random salt	Salted password	Salted password hash
Alice	apple	4r9g8	&hgu$	&hgu$apple	r$wdc
Bob	banana	3ca53	#x!@3	#x!@3banana	ei832
Carol	carrot	8dusi	5!%vX	5!%vXcarrot	5t9ri
Devin	banana	3ca53	9*^cs	9*^csbanana	xde4z
Elisa	eggplant	4v37d	={4*f	={4*feggplant	i8s74

Salts make dictionary attacks and brute force attacks for cracking large number of passwords much slower (although they do not benefit cracking just one password), and also limit the impact of rainbow tables. Another benefit of a salt is that if two users choose the same password, this will not help the attacker. In Table 12-3, both Bob and Devin selected the same password (banana) that resulted in the same unsalted hashed password (3ca53). Without salts, an attacker who is able to crack Bob's password would also immediately know Devin's password without performing any computations. By adding salts, however, each password digest will be different.

What You Have: Tokens, Cards, and Cell Phones

Another type of authentication credential is based on the approved user having a specific item in his possession. Such items are often used in conjunction with passwords. Because the user is using more than one type of authentication credential—both what a user knows (the password) and what the user has—this type of authentication credential is called multifactor authentication. (Using just one type of authentication is called single-factor authentication.) The most common items that are used for authentication are tokens, cards, and cell phones.

- **Tokens** A token is typically a small device (usually one that can be affixed to a keychain) with a window display, as shown in Figure 12-4. Instead of the user presenting a password (what she knows), a token introduces a different form of authentication based on what the person has (a token). Tokens can be used to create a one-time password (OTP), an authentication code that can be used only once or for a limited period of time. There are two types of OTPs. A time-based one-time password (TOTP)

changes after a set time period. the token and a corresponding authentication server share an algorithm (each user's token has a different algorithm), and the token generates a code from the algorithm once every 30 to 60 seconds. This code is valid for only the brief period of time that it is displayed on the token. When the user logs in, she enters her username along with the code currently being displayed on the token. When the authentication server receives it, the server looks up the algorithm associated with that specific user, generates its own code, and then compares it with what the user entered. If they are identical, the user is authenticated. An attacker who steals the code would have to use it within the token's time limit. Instead of changing after a set number of seconds, an HMAC-based one-time password (HOTP) is "event-driven" and changes when a specific event occurs, such as when a user enters a personal identification number (PIN) on the token's keypad, which triggers the token to create a random code. For example, after entering the PIN 1694, the code 190411 is displayed. Tokens have several advantages over passwords. First, standard passwords are static: they do not change unless the user is forced to create a new password. Because passwords do not change frequently, this can give an attacker a lengthy period of time in which to crack and then use the password. In contrast, tokens produce dynamic passwords that change frequently. Second, a user might not know if an attacker has stolen her password, and confidential information could be accessed without the user knowing it was taking place. If a token is stolen, it would become obvious and steps could be taken immediately to disable that account.

- **Cards** Several types of cards can be used as authentication credentials. A smart card, contains an integrated circuit chip that can hold information, which then can be used as part of the authentication process. Smart cards can be either contact cards, which contain a tell-tale "pad" allowing electronic access to the contents of the chip, or contactless cards that do not require physical contact with the card itself. One type of smart card is currently being distributed by the U.S. government. A common access card (CAC) is a U.S. Department of Defense (DoD) smart card that is used for the identification of active-duty and reserve military personnel along with civilian employees and special contractors. A CAC resembles a credit card. In addition to an integrated circuit chip, it has a bar code and magnetic stripe along with the bearer's picture and printed information. This card can be used to authenticate the owner as well as for encryption. The smart card standard covering all U.S. government employees is the Personal Identity Verification (PIV) standard.
- **Cell Phones** Tokens and cards are increasingly being replaced today with cell phones. A code can be sent to a user's cell phone through an app on the device or as a text message when using TOTP. Cell phones also allow a user to send a request via the phone to receive an HOTP authorization code.

Lesson Summary

Different authentication credentials can be presented to an information technology system to verify the genuineness of the user. These can be classified into five categories: what you know, what you have, what you are, what you do, and where you are. The most common "what you know" type of authentication is a password.

A password is a secret combination of letters, numbers, and/or characters that only the user should have knowledge of and is the most common type of authentication in use today. Passwords provide a weak degree of protection because they rely on human memory. Human beings have a finite limit to the number of items that they can memorize. Because of the burdens that passwords place on human memory, users often take shortcuts to help them recall their passwords.

Although there are several different types of password attacks, the most common password attacks today use offline cracking. Attackers steal the file of password digests and then load that file onto their own computers so they can attempt to discover the passwords by comparing the stolen digest passwords with candidate digests that they have created. An automated brute force attack uses every possible combination of letters, numbers, and characters to create candidates that are matched with those in the stolen file. A dictionary attack begins with the attacker creating digests of common dictionary words, which are then compared with those in a stolen password file. The hybrid attack slightly alters dictionary words. Attackers often use rainbow tables, which make password attacks easier by creating a large pre-generated data set of encrypted passwords. Large collections of stolen password files have allowed attackers to create a larger number of accurate candidates and to understand how users create passwords.

There are several defenses against password attacks. The most basic is password complexity or creating long and complex passwords. Credential management involves properly managing passwords, often by using technology instead of human memory. Another defense is to use a password hashing algorithm instead of a general-purpose hash algorithm. Salts, or random strings added to passwords, also can make passwords more difficult for attackers to break.

Lessson 3 Access Control and Identity Management 2

Introduction

Restricting a user to just his office is similar to the concept of access control, the actions of the night security guard to verify users's identity parallel the act of authentication in information security. Authentication is the process of ensuring that the person desiring access to resources is authentic, and not an imposter.

In this chapter we will study authentication and the secure management of user accounts that enforces authentication. First you will look at the different types of authentication credentials that can be used to verify a user's identity. Next you will see how a single sign-on might be used. Finally, you will look into the techniques and technology used to manage user accounts in a secure fashion.

Lesson Proper

What You Are: Biometrics

In addition to authentication based on what a person knows or has, another category rests on the features and characteristics of the individual. This type of "what you are" authentication involves standard biometrics and cognitive biometrics. Standard Biometrics Standard biometrics uses a person's unique physical characteristics for authentication (what he is).

1. Standard Biometrics. Standard biometrics can use fingerprints or other unique characteristics of a person's face, hands, or eyes (irises and retinas) to authenticate a user. Fingerprint scanners have become the most common type of standard biometric device. Every user's fingerprint consists of a number of ridges and valleys, with ridges being the upper skin layer segments of the finger and valleys the lower segments. In one method of fingerprint scanning, the scanner locates the point where these ridges end and split, converts them into a unique series of numbers, and then stores the information as a template. A second method creates a template from selected locations on the finger. There are two basic types of fingerprint scanners:

1. Static fingerprint Scanner. A static fingerprint scanner requires the user to place the entire thumb or finger on a small oval window on the scanner. The scanner takes an optical "picture" of the fingerprint and compares it with the fingerprint image onfile.

2. Dynamic fingerprint Scanner. The other type of scanner is known as a dynamic fingerprint scanner. A dynamic fingerprint scanner has a small slit or opening where the user needs to rest its finger for identification.

Standard biometrics has two disadvantages. The first is the cost. Biometric readers (hardware scanning devices) must be installed at each location where authentication is required. The second disadvantage is that biometric readers are not always foolproof and can reject authorized users while accepting unauthorized users. These errors are mainly due to the many facial or hand characteristics that must be scanned and then compared.

2. Cognitive Biometrics. Whereas standard biometrics considers a person's physical characteristics, the field of cognitive biometrics is related to the perception, thought process, and understanding of the user. Cognitive biometrics is considered to be much easier for the user to remember because it is based on the user's life experiences. This also makes it more difficult for an attacker to imitate. One type of cognitive biometrics is ***picture gesture authentication (PGA)*** for touch-enabled devices. Users select a picture to use for which there should be at least 10 "points of interest" on the photograph that could serve as "landmarks" or places to touch, connect with a line, or draw a circle around. Specific gestures—tap, line, or circle—are then used to highlight any parts of the picture and these gestures are recorded. When logging in, a user reproduces those same gestures on the photograph. In order for an attacker to replicate these actions, she would need to know the parts of the image that were highlighted, the order of the gestures, as well as the direction, and the starting and ending points, of the circles and lines.

A similar example of cognitive biometrics requires the user to identify specific faces. Users are provided a random set of photographs of different faces, typically three to seven, to serve as their password. They are taken through a "familiarization process" that is intended to imprint the faces in the user's mind. When the user logs in, he must select his assigned faces from three to five different groups, with each group containing nine faces. These groups are presented one at a time until all the faces have been correctly identified. Cognitive biometrics is considered much easier for the end-user and may provide a higher degree of protection. It is predicted that cognitive biometrics could become a key element in authentication in the future.

What You Do: Behavioral Biometrics

Another type of authentication is based on actions that the user is uniquely qualified to perform. This is sometimes called behavioral biometrics. Two examples are keystroke dynamics and voice recognition.

1. Keystroke Dynamics. Keystroke Dynamics One type of behavioral biometrics is **keystroke dynamics**, which attempts to recognize a user's unique typing rhythm. All users type at a different pace. During World War II, the U.S. military could distinguish enemy coders who tapped out Morse code from Allied coders by their unique rhythms. A study funded by the U.S. National Bureau of Standards concluded that the keystroke dynamics of entering a username and password could provide up to 98 percent accuracy. 8 Keystroke dynamics **uses two unique typing variables**. The **first is known as dwell time**, which is the time it takes for a key to be pressed and then released. **The second characteristic is flight time,** or the time between keystrokes (both "down" when the key is pressed and "up" when the key is released are measured). Multiple samples are collected to form a user typing template. When the user enters his username and password, they are sent, along with the user's individual typing sample obtained by entering the username and password, to the authentication server. If both the password and the typing sample match, those stored on the authentication server, and the user is approved; if the typing template does not match even though the password does, the user is not authenticated.

Keystroke dynamics holds a great deal of potential. Because it requires no specialized hardware and because the user does not have to take any additional steps beyond entering a username and password, some security experts predict that keystroke dynamics will become widespread in the near future.

2 Voice Recognition, Voice Recognition Because all users' voices are different, voice recognition can be used to authenticate users based on the unique characteristics of a person's voice. Several characteristics make each person's voice unique, from the size of the head to age. These differences can be quantified and a user voice template can be created, much like the template used in keystroke dynamics. One of the concerns regarding voice recognition is that an attacker could record the user's voice and then create a recording to use for authentication. However, this would be extremely difficult to do. Humans speak in phrases and sentences instead of isolated words. The phonetic cadence, or speaking two words together in a way that one word "bleeds" into the next word, becomes part of each user's speech pattern. It would be extremely difficult to capture several hours of someone's voice, parse it into separate words, and then combine the words in real time to defeat voice recognition security.

Where You Are: Geolocation

A final type of authentication can be based where the user is located. Known as geolocation, it is the identification of the location of a person or object using technology. Although geolocation may not uniquely identify the user, it can indicate if an attacker is trying to perform a malicious action from a location different from the normal location of the user. For example, where does Alice normally access her bank's website? If it is typically from her home computer on nights and weekends, then this information can be used to establish a geolocation pattern based on the Internet Protocol (IP) address of Alice's computer. If a computer located in China attempts to access her bank's website, this may be an indication that an attacker instead of Alice is at work. Geolocation is done to some degree by most banks, so that generally a bank will turn down requests for wire transfers from overseas locations unless the user has specifically approved such a transfer in advance with the bank.

Geolocation is not restricted to banking. Many websites will not allow a user to access an account if the computer is located in North Korea when normally the access is from Philippines. The website may require a second type of authentication, such as a code sent as a text message to a cell phone number on file before the user can be authenticated.

Single Sign-On

One of the problems facing users today is the fact that they have multiple accounts across multiple platforms that all ideally use a unique username and password. The difficulty in managing all of these different authentication credentials frequently causes users to compromise and select the least burdensome password and then use it for all accounts. A solution to this problem is to have one username and password to gain access to all accounts so that the user has only one username and password to remember. This is the idea behind identity management, which is using a single authentication credential that is shared across multiple networks. When those networks are owned by different organizations, it is called federated identity management (FIM), or just federation. One application of FIM is called single sign-on (SSO), or using one authentication credential to access multiple accounts or applications. SSO holds the promise of reducing the number of usernames and passwords that users must memorize (potentially, to just one).

There are **several implementations of web-based FID systems**. Examples of some popular SSOs include Microsoft Account, OpenID, and OAuth.

1. Microsoft Account. Microsoft has promoted SSO technology for several years. In 1999 Microsoft introduced .NET Passport before changing the name to Microsoft Passport Network. The name was changed again to Windows Live ID in 2006 as an SSO for web commerce. Today the technology is simply known as Microsoft Account. Although Windows Live ID was originally designed as a federated identity management system that would be used by a wide variety of web servers, because of security issues and

privacy concerns, Windows Live ID received limited support. Microsoft Account is similar to Windows Live ID and serves as the authentication system for different Microsoft products.

Microsoft Account, like Windows Live ID, requires a user to create a standard username and password. When the user wants to log in to a website that supports Microsoft Account, the user is redirected to the nearest authentication server, which asks for the username and password over a secure connection. Once authenticated, the user is given an encrypted timelimited "global" cookie that is stored on her computer along with an encrypted ID tag. This ID tag is then sent to the website that the user wants to log into. The website uses this ID tag for authentication and stores its own encrypted and time-limited "local" cookie on the user's computer. The use of "global" and "local" cookies is the basis of Microsoft Account. When the user logs out of her Microsoft account, these cookies are erased.

2. OpenID. Unlike Microsoft Account, which is proprietary and has centralized authentication, OpenID is a decentralized open-source FIM that does not require specific software to be installed on the desktop. OpenID is a Uniform Resource Locator (URL)–based identity system. An OpenID identity is only a URL backed up by a username and password. OpenID provides a means to prove that the user owns that specific URL.

The **steps for creating and using OpenID** are as follows:

1. The user goes to a free website that provides OpenID accounts, such as MyOpenID.com, and creates an account with a username (Me) and password. The user is then given the OpenID account of Me.myopenid.com.

2. When the user visits a website like BuyThis.com that requires him to sign in, he can instead choose to use OpenID. He simply enters his OpenID URL, Me.myopenid.com.

3. BuyThis.com redirects him to MyOpenID.com where he is required to enter his password to authenticate himself and indicate he trusts BuyThis.com with his identity.

4. MyOpenID.com sends him back to BuyThis.com, where he is now authenticated.

OpenID does have some security weaknesses. One weakness is that OpenID depends on the URL identifier routing to the correct server, which depends on a domain name server (DNS) that may have its own security weaknesses. In its current format, OpenID is generally not considered strong enough for most banking and e-commerce websites. However, OpenID is considered suitable for other less secure sites.

3. Open Authorization (OAuth). Consider Abby who wants to post photos online of her latest vacation for her friends. Abby starts by first logging into her account on an online storage site (Box.net) to upload her photos from her cell phone. Then she accesses her favorite photo sharing site (Flickr.com) to post her photos along with her comments. Abby must log in to this site with another username and password. After the photos are posted, she then accesses her online contact list (Gmail.google.com) to create a list of her friends to whom she wants to show her photos; again, Abby uses another username and password for her Gmail account. She then goes to her social media site (Facebook.com) to spread the word, and once again must enter a username and password.

A technology to avoid using multiple passwords is an open-source service similar to OpenID called **Open Authorization (OAuth). OAuth** permits users to share resources stored on one site with a second site without forwarding their authentication credentials to the other site. It also allows for different applications to seamlessly share data across sites. This would enable Abby to send her photos to Box, which would then automatically communicate with Flickr, Gmail, and Facebook. OAuth relies upon token credentials. A user sends her authentication credentials to a server (such as a web application server) and also authorizes the server to issue token credentials to a third-party server. These token credentials are used in place of transferring the user's username and password. The tokens are not generic, but are for specific resources on a site for a limited period of time.

Account Management

Managing credentials such as passwords in user accounts can be accomplished by setting restrictions regarding the creation and use of passwords. Although these restrictions can be performed on a user-by-user basis, this quickly becomes cumbersome and is a security risk: it is too easy to overlook one setting in one user account and create a security vulnerability. A preferred approach is to assign privileges by group. In a Microsoft Windows environment, there are two categories of group password settings. The **first category is called Password Policy Settings** and is configured by using Group Policy at the domain level. There are six common domain password policy settings called password setting objects.

Attribute	Description	Recommended setting
Enforce password history	Determines the number of unique new passwords a user must use before an old password can be reused (from 0 to 24).	24 new passwords
Maximum password age	Determines how many days a password can be used before the user is required to change it. The value of this setting can be between 0 and 999.	90 days
Minimum password age	Determines how many days a new password must be kept before the user can change it (from 0 to 999). This setting is designed to work with the Enforce password history setting so that users cannot quickly reset their passwords the required number of times, and then change back to their old passwords.	1 day
Minimum password length	Determines the minimum number of characters a password can have (0–28).	12 characters
Passwords must meet complexity requirements	Determines whether the following are used in creating a password: Passwords cannot contain the user's account name or parts of the user's full name that exceed two consecutive characters; must contain characters from three of the following four categories—English uppercase characters (A through Z), English lowercase characters (a through z), digits (0 through 9), and nonalphabetic characters (!, $, #, %).	Enabled
Store passwords using reversible encryption	Provides support for applications that use protocols which require knowledge of the user's password for authentication purposes. An attacker who can circumvent the encryption will be able to log on to the network with these passwords.	Disabled

The **second category is the Account Lockout Policy**, which is an Active Directory Domain Services (AD DS) security feature. The lockout prevents a login after a set number of failed login attempts within a specified period and also can specify the length of time that the lockout is in force. This helps prevent attackers from online guessing of user passwords.

In addition to account policy enforcement, other steps should be taken as well. For example, generic accounts should be prohibited, and user access should be subject to continuous monitoring and review. Also, care must be taken with transitive trust. Transitive trust is a two-way relationship that is automatically created between parent and child domains in a Microsoft Active Directory Forest. When a new domain is created, it shares resources with its parent domain by default, which can enable an authenticated user to access resources in both the child and the parent.

Lesson Summary

Different authentication credentials can be presented to an information technology system to verify the genuineness of the user. These can be classified into five categories: what you know, what you have, what you are, what you do, and where you are. The most common "what you know" type of authentication is a password.

A password is a secret combination of letters, numbers, and/or characters that only the user should have knowledge of and is the most common type of authentication in use today. Passwords provide a weak degree of protection because they rely on human memory. Human beings have a finite limit to the number of items that they can memorize. Because of the burdens that passwords place on human memory, users often take shortcuts to help them recall their passwords.

Behavioral biometrics, or "what you do," authenticates by normal actions that the user performs. Behavioral biometric technologies include keystroke dynamics and voice recognition. A final type of

authentication, geolocation, is the identification of the location ("where you are") of a person or object using technology. Although geolocation may not uniquely identify the user, it can indicate if an attacker is trying to perform a malicious action from a location different from the normal location of the user.

One of the problems facing users today is that they have multiple accounts across multiple platforms that all ideally use a unique username and password. The difficulty in managing all these different authentication credentials frequently causes users to compromise and select the least burdensome password and then use it for all accounts. A solution to this problem is to have one username and password to gain access to all accounts so that the user has only one username and password to remember. This is called single sign-on (SSO). Examples of some of the popular SSOs include Microsoft Account, OpenID, and OAuth.

Lesson 4. Data and Application Security

Introduction

According to Veracode's State of Software Security Vol. 10 reports, 83% of the 85,000 applications it tested had at least one security flaw. Many had much more, as their research found a total of 10 million flaws, and 20% of all apps had at least one high severity flaw. Not all of those flaws present a significant security risk, but the sheer number is troubling.

The faster and sooner in the software development process you can find and fix security issues, the safer your enterprise will be. Because everyone makes mistakes, the challenge is to find those mistakes in a timely fashion. For example, a common coding error could allow unverified inputs. This mistake can turn into SQL injection attacks and then data leaks if a hacker finds them.

Application security tools that integrate into your application development environment can make this process and workflow simpler and more effective. These tools are also useful if you are doing compliance audits since they can save time and expense by catching problems before the auditors seen them.

In this lesson, we are going to study data and application security in more details together with tools for security and the current state of the web application security

Lesson Proper

Application Security

Application security is the process of making apps more secure by finding, fixing, and enhancing the security of apps. Much of this happens during the development phase, but it includes tools and methods to protect apps once they are deployed. This is becoming more important as hackers increasingly target applications with their attacks. Hundreds of tools are available to secure various elements of your applications portfolio, from locking down coding changes to assessing inadvertent coding threats, evaluating encryption options and auditing permissions and access rights. There are specialized tools for mobile apps, network-based apps, and for firewalls designed especially for web applications.

The rapid growth in the application security segment has been helped by the changing nature of how enterprise apps are being constructed in the last several years. Gone are the days where an IT shop would take months to refine requirements, build and test prototypes, and deliver a finished product to an end-user department. The idea almost seems quaint nowadays.

Most common software weaknesses

One way to keep aware of the software vulnerabilities that attackers are likely to exploit is MITRE's annual CWE Most Dangerous Software Weaknesses list. MITRE tracks CWEs (Common Weakness Enumeration), assigning them a number much as they do with its database of Common Vulnerabilities and Exposures (CVEs). Each weakness is rated depending on the frequency that it is the root cause of a vulnerability and the severity of its exploitation.

Below are the top 10 CWEs in MITRE's 2020 CWE top 25 with scores:

1. Cross-site scripting (46.82)
2. Out-of-bounds write (46.17)
3. Improper input validation (33.47)
4. Out-of-bounds read (26.5)
5. Improper restriction of operations within the bounds of a memory buffer (23.73)
6. SQL injection (20.69)
7. Exposure of sensitive information to an unauthorized actor (19.16)
8. Use after free (18.87)
9. Cross-site reques forgery (CSRF) (17.29)
10. OS command injection (16.44)

Application security tools

While there are numerous application security software product categories, the meat of the matter has to do with two: **security testing tools** and **application shielding products**. The former is a more mature market with dozens of well-known vendors, some of them are lions of the software industry such as IBM, CA and MicroFocus. These tools are well enough along that Gartner has created its Magic Quadrant and classified their importance and success. Review sites such as IT Central Station have been able to survey and rank these vendors, too. Gartner categorizes the security testing tools into several broad buckets, and they are somewhat useful for how you decide what you need to protect your app portfolio:

- **Static testing**, which analyzes code at fixed points during its development. This is useful for developers to check their code as they are writing it to ensure that security issues are being introduced during development.
- **Dynamic testing**, which analyzes running code. This is more useful, as it can simulate attacks on production systems and reveal more complex attack patterns that use a combination of systems.
- **Interactive testing,** which combines elements of both static and dynamic testing.
- **Mobile testing** is designed specifically for mobile environments and can examine how an attacker can leverage the mobile OS and the apps running on them in its entirety.

Application Shielding Tools. Another way to look at the testing tools is how they are delivered, either via an on-premises tool or via a SaaS-based subscription service where you submit your code for online analysis. Some even do both.

One caveat is the programming languages supported by each testing vendor. Some limit their tools to just one or two languages. (Java is usually a safe bet.) Others are more involved in the Microsoft .Net universe. The same goes for integrated development environments (IDEs): some tools operate as plug-ins or extensions to these IDEs, so testing your code is as simple as clicking on a button.

Another issue is whether any tool is isolated from other testing results or can incorporate them into its own analysis. IBM is one of the few that can import findings from manual code reviews, penetration testing, vulnerability assessments, and competitors' tests. This can be helpful, particularly if you have multiple tools that you need to keep track of.

The **main objective of these tools** is **to harden the application so that attacks are more difficult to carry out**. This is less charted

territory. Here you'll find a vast collection of smaller, point products that in many cases have limited history and customer bases. The goal of these products is to do more than just test for vulnerabilities and actively prevent your apps from corruption or compromise. They encompass a few different broad categories:

- **Runtime application self-protection (RASP):** These tools could be considered a combination of testing and shielding. They provide a measure of protection against possible reverse-engineering attacks. RASP tools are continuously monitoring the behavior of the app, which is useful particularly in mobile environments when apps can be rewritten, run on a rooted phone or have privilege abuse to turn them into doing nefarious things. RASP tools can send alerts, terminate errant processes, or terminate the app itself if found compromised.
  RASP will likely become the default on many mobile development environments and built-in as part of other mobile app protection tools. Expect to see more alliances among software vendors that have solid RASP solutions.
- **Code obfuscation**: Hackers often use obfuscation methods to hide their malware (Links to an external site.), and now tools allow developer to do this to help protect their code from being attacked.
- **Encryption and anti-tampering tools**: These are other methods that can be used to keep the bad guys from gaining insights into your code.
- **Threat detection tools**: These tools examine the environment or network where your apps are running and make an assessment about potential threats and misused trust relationships. Some tools can provide device "fingerprints" to determine whether a mobile phone has been rooted or otherwise compromised.

Application security challenges

Part of the problem is that IT has to satisfy several different masters to secure their apps.

1. They first have to keep up with the evolving security and application development tools market.

2. IT also has to anticipate the business needs as more enterprises dive deeper into digital products and their application portfolio needs evolve to more complex infrastructure.

3. They also have to understand how SaaS services are constructed and secured. This has been an issue, as a recent survey of 500 IT managers has found the average level of software design knowledge has been lacking. The report states, "CIOs may find themselves in the hot seat with senior leadership as they are held accountable for reducing complexity, staying on budget and how quickly they are modernizing to keep up with business demands."

4. The responsibility for application security could be spread across several different teams within your IT operations: The network folks could be responsible for running the web app firewalls and other network-centric tools, the desktop folks could be responsible for running endpoint-oriented tests, and various development groups could have other concerns. This makes it hard to suggest one tool that will fit everyone's needs, which is why the market has become so fragmented.

Application security trends

In January 2019, Imperva published its State of Web Application Vulnerabilities in 2018. The overall findings were positive. While the number of web application vulnerabilities continues to grow, that growth is slowing. That's due primarily to a decline in IoT vulnerabilities--only 38 new ones reported in 2018 versus 112 in 2017. API vulnerabilities, on the other hand, increased by 24% in 2018, but at less than half the 56% growth rate of 2017. Another area seeing more vulnerabilities emerge according to the Imperva report is in content management systems, WordPress in particular. That platform saw a 30% increase in the number of reported vulnerabilities.

The report noted that the Drupal content management system, despite being far less popular than WordPress, is becoming a target for attackers because of two vulnerabilities: Drupalgeddon2 (CVE-2018-7600) and Drupalgeddon3 (CVE-2018-7602). Both allow attacks to connect to back-end databases, scan and infect networks and clients with malware, or mine cryptocurrencies. Imperva claims to have blocked more than a half-million of attacks that use these vulnerabilities in 2018.

The Veracode report shows that the most common types of flaws are:

- Information leakage (64%)
- Cryptographic issues (62%)
- CRLF injection (61%)
- Code quality (56%)
- Insufficient input validation (48%)
- Cross-site scripting (47%)
- Directory traversal (46%)
- Credentials management (45%)

One positive trend that the Veracode study found was that application scanning makes a big difference when it comes to fixing rate and time to fix for application flaws. Overall fix rates, especially for high-severity flaws, are improving. The overall fix rate is 56%, up from 52% in 2018, and the highest severity flaws are fixed at a rate of 75.7%. A DevSecOps approach with frequent scanning and testing of software will drive down the time to fix flaws. The median time to repair for applications scanned 12 times or fewer per year was 68 days, while an average scan rate of daily or more lowered that rate to 19 days.

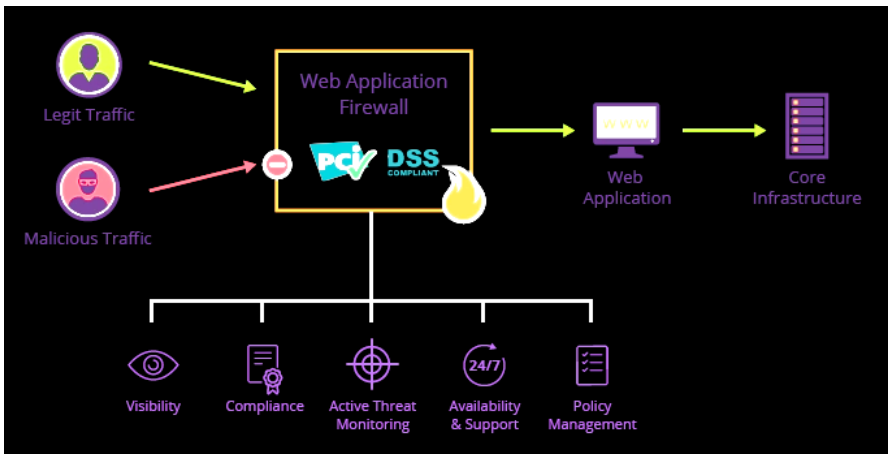
The State of Web Application Vulnerabilities in 2018

As a web application firewall provider, part of our job at Imperva is to continually monitor for new security vulnerabilities. To do this, we use internal software that collects information from various data sources such as vulnerability databases, newsletters, forums, social media and more, integrates it into a single repository, and assesses each vulnerability's priority. Having this kind of data puts us in a unique position to provide an analysis of all web application vulnerabilities throughout the year, view trends, and notice significant changes in the security landscape. As we did last year, we took a look back at 2018 to understand the changes and trends in web application security over the past year.

The bad news is that in 2018, like 2017, we continued to see a trend of an increasing number of web application vulnerabilities, particularly vulnerabilities related to injection such as SQL injection, command injection, object injection, etc. On the content management system (CMS) front, WordPress vulnerabilities continue to grow, and they continue to dominate in terms of the number of vulnerabilities published in the CMS category. Although WordPress leads the pack in sheer vulnerabilities numbers, Drupal vulnerabilities had a larger effect and were used in mass attacks that targeted hundreds of thousands of sites during 2018. However, there is some good news for the security industry — the number of Internet of Things (IoT) vulnerabilities declined, as well as the number of vulnerabilities related to weak authentication. In the server-side technologies category, the number of PHP vulnerabilities continued to decline. In addition, the growth in API vulnerabilities also slightly declined.

Reviewer in Data and Application Security

Web application firewall (WAF)



Web application firewalls (WAFs) are hardware and software solutions used for protection from application security threats. These solutions are designed to examine incoming traffic to block attack attempts, thereby compensating for any code sanitization deficiencies. By securing data from theft and manipulation, WAF deployment meets a key criteria for PCI DSS certification. Requirement 6.6 states that all credit and debit cardholder data held in a database must be protected. Generally, deploying a WAF doesn't require making any changes to an application, as it is placed ahead of its DMZ at the edge of a network. From there, it acts as a gateway for all incoming traffic, blocking malicious requests before they have a chance to interact with an application. WAFs use several different heuristics to determine which traffic is given access to an application and which needs to be weeded out. A constantly-updated signature pool enables them to instantly identify bad actors and known attack vectors.

Almost all WAFs can be custom-configured for specific use cases and security policies, and to combat emerging (a.k.a., zero-day) threats. Finally, most modern solutions leverage reputational and behavior data to gain additional insights into incoming traffic. WAFs are typically integrated with other security solutions to form a security perimeter. These may include distributed denial of service (DDoS) protection services that provide additional scalability required to block high-volume attacks.

Web application security checklist

In addition to WAFs, there are a number of methods for securing web applications. The following processes should be part of any web application security checklist:

- **Information gathering** – Manually review the application, identifying entry points and client-side codes. Classify third-party hosted content.
- **Authorization** – Test the application for path traversals; vertical and horizontal access control issues; missing authorization and insecure, direct object references.
- **Cryptography** – Secure all data transmissions. Has specific data been encrypted? Have weak algorithms been used? Do randomness errors exist?
- **Denial of service** – Improve an application's resilience against denial of service threats by testing for anti-automation, account lockout, HTTP protocol DoS, and SQL wildcard DoS. This doesn't cover protection from high-volume DoS and DDoS attacks, which are best countered by a combination of filtering solutions and scalable resources.

Data Security: Definition, Explanation, and Guide

Data Security is a process of protecting files, databases, and accounts on a network by adopting a set of controls, applications, and techniques that identify the relative importance of different datasets, their sensitivity, regulatory compliance requirements, and then applying appropriate protections to secure those resources.

Similar to other approaches like perimeter security, file security, or user behavioral security, data security is not the be-all, end-all for a security practice. It's one method of evaluating and reducing the risk that comes with storing any kind of data.

What are the Main Elements of Data Security?

The core elements of data security are confidentiality, integrity, and availability. Also known as the CIA triad, this is a security model and guide for organizations to keep their sensitive data protected from unauthorized access and data exfiltration.

- **Confidentiality** ensures that data is accessed only by authorized individuals;
- **Integrity** ensures that information is reliable as well as accurate; and
- **Availability** ensures that data is both available and accessible to satisfy business needs.

What is Data Security Considerations?

There are a few data security considerations you should have on your radar:

- **Where is your sensitive data located?** You won't know how to protect your data if you don't know where your sensitive data is stored.
- **Who has access to your data?** When users have unchecked access or infrequent permission reviews, it leaves organizations at risk of data abuse, theft or misuse. Knowing who has access to your company's data at all times is one of the most vital data security considerations to have.
- **Have you implemented continuous monitoring and real-time alerting on your data?** Continuous monitoring and real-time alerting are important not just to meet compliance regulations, but can detect unusual file activity, suspicious accounts, and computer behavior before it's too late.

What is Data Security Technologies?

The following are data security technologies used to prevent breaches, reduce risk and sustain protections.

- **Data Auditing.** The question isn't if a security breach occurs, but *when* a security breach will occur. When forensics gets involved in investigating the root cause of a breach, having a data auditing solution in place to capture and report on access control changes to data, who had access to sensitive data, when it was accessed, file path, etc. are vital to the investigation process. Alternatively, with proper data auditing solutions, IT administrators can gain the visibility necessary to *prevent* unauthorized changes and potential breaches.
- **Data Real-Time Alerts.** Typically it takes companies several months to discover a breach. Companies often find out about breaches through their customers or third parties instead of their own IT departments. By monitoring data activity and suspicious behavior in real-time, you can discover more quickly security breaches that lead to accidental destruction, loss, alteration, unauthorized disclosure of, or access to personal data.
- **Data Risk Assessment.** Data risk assessments help companies identify their most overexposed sensitive data and offer reliable and repeatable steps to prioritize and fix serious security risks. The process starts with identifying sensitive data accessed via global groups, stale data, and/or inconsistent permissions. Risk assessments summarize important findings, expose data vulnerabilities, provide a detailed explanation of each vulnerability, and include prioritized remediation recommendations.
- **Data Minimization.** The last decade of IT management has seen a shift in the perception of data. Previously, having more data was almost always better than less. You could never be sure ahead of time what you might want to do with it. data is a liability. The threat of a reputation-destroying data breach, loss in the millions,

or stiff regulatory fines all reinforce the thought that collecting anything beyond the minimum amount of sensitive data is extremely dangerous.

- **Purge Stale Data**. Data that is not on your network is data that can't be compromised. Put in systems that can track file access and automatically archive unused files. In the modern age of yearly acquisitions, reorganizations, and "synergistic relocations," it's quite likely that networks of any significant size have multiple forgotten servers that are kept around for no good reason.

How Do You Ensure Data Security?

While data security isn't a panacea, you can take several steps to ensure data security. Here are a few that we recommend.

- **Quarantine Sensitive Files**. A rookie data management error is placing a sensitive file on a share open to the entire company. Quickly get control of your data with data security software that continually classifies sensitive data and moves data to a secure location.
- **Track User Behavior against Data Groups**. The general term plaguing rights management within an organization is "overpermissioning'. That temporary project or rights granted on the network rapidly becomes a convoluted web of interdependencies that result in users collectively having access to far more data on the network than they need for their role. Limit a user's damage with data security software that profiles user behavior and automatically puts in place permissions to match that behavior.
- **Respect Data Privacy.** Data Privacy is a distinct aspect of cybersecurity dealing with the rights of individuals and the proper handling of data under your control.

Data Security Regulations

Regulations such as **HIPAA (healthcare)**, **SOX (public companies)** and **GDPR (anyone who knows that the EU exists)** are best considered from a data security perspective. From a data security perspective, regulations such as HIPAA, SOX, and GDPR require that organizations:

- Track what kinds of sensitive data they possess
- Be able to produce that data on demand
- Prove to auditors that they are taking appropriate steps to safeguard the data

These regulations are all in different domains but require a strong data security mindset. Let's take a closer look to see how data security applies under these compliance requirements:

Health Insurance Portability and Accountability Act (HIPAA)

The **Health Insurance Portability and Accountability Act** was legislation passed to regulate health insurance. Section 1173d—calls (Links to an external site.) for the Department of Health and Human Services "to adopt security standards that take into account the technical capabilities of record systems used to maintain health information, the costs of security measures, and the value of audit trails in a computerized record system."

From a data security point of view, **here are a few areas you can focus on to meet HIPAA compliance:**

- **Continually Monitor File and Perimeter Activity – Continually monitor activity and access to sensitive data – not only to achieve HIPAA compliance but as a general best practice.
- **Access Control** – Re-compute and revoke permissions to file share data by automatically permission access to individuals who only have a need-to-know business right.
- **Maintain a Written Record** – Ensure you keep detailed activity records for all user objects including administrators within an active directory and all data objects within file systems. Generate changes automatically and send to relevant parties who need to receive the reports.

Sarbanes-Oxley (SOX)

The Sarbanes-Oxley Act of 2002, commonly called "SOX" or "Sarbox," is a United States federal law requiring publicly traded companies to submit an annual assessment of the effectiveness of their internal financial auditing controls.

From a data security point of view, here are your focus points to meet SOX compliance:

- **Auditing and Continuous Monitoring** – SOX's Section 404 is the starting point for connecting auditing controls with data protection: it asks public companies to include in their annual reports an assessment of their internal controls for reliable financial reporting, and an auditor's attestation.
- **Access Control –** Controlling access, especially administrative access, to critical computer systems is one of the most vital aspects of SOX compliance. You'll need to know which administrators changed security settings and access permissions to file servers and their contents. The same level of detail is prudent for users of data, displaying access history and any changes made to access controls of files and folders.
- **Reporting –** To provide evidence of compliance, you'll need detailed reports including:
  - data use, and every user's every file-touch
  - user activity on sensitive data
  - changes including permissions changes which affect the access privileges to a given file or folder
  - revoked permissions for data sets, including the names of users

General Data Protection Regulation (GDPR)

The EU's General Data Protection Regulation covers the protection of EU citizen personal data, such as social security numbers, date of birth, emails, IP addresses, phone numbers, and account numbers. From a data security point of view, here's what you should focus on to meet GDPR compliance:

- **Data Classification** – Know where sensitive personal data is stored. It's critical to both protecting the data and also fulfilling requests to correct and erase personal data, a requirement known as the right to be forgotten.
- **Continuous Monitoring** – The breach notification requirement enlists data controllers to report the discovery of a breach within 72 hours. You'll need to spot unusual access patterns against files containing personal data. Expect hefty fines if you fail to do so.
- **Metadata** – With the GDPR requirement to set a limit on data retention, you'll need to know the purpose of your data collection. Personal data residing on company systems should be regularly reviewed to see whether it needs to be archived and moved to cheaper storage or saved for the future.
- **Data Governance** – Organizations need a plan for data governance. With data security by design as the law, organizations need to understand who is accessing personal data in the corporate file system, who should be authorized to access it, and limit file permission based on employees' actual roles and business needs.