Cybersecurity

Knowledge review

Jakub Piotr Hamerliński

The information provided on this presentation does not, and is not intended to, constitute legal advice; instead, all information, content, and materials available on this presentation are for general informational purposes only. Information on this presentation may not constitute the most up-to-date knowledge or other information. This presentation may contain links to other third-party resources. Such links are only for the convenience of the reader, user or browser; the author does not recommend or endorse the contents of the third-party sites.

USER ACCOUNT SECURITY

What's the difference between the following user groups: Users, Authenticated Users and Everyone?

What's the difference between the following user groups: Users, Authenticated Users and Everyone?

Users: This group includes all local and domain user accounts that have been created on the Windows operating system. These users have basic permissions to access and use the computer, such as logging in, creating files and folders, and running programs.

Authenticated Users: This group includes all users who have successfully logged in to the Windows operating system, including those in the Users group. These users have additional permissions to access network resources, such as shared folders and printers.

Everyone: This group includes all users, including those in the Users and Authenticated Users groups, as well as any other users that may have access to the computer through the network. This group has the most permissions and can access all resources on the computer, including system files and settings.

In summary, Users have basic permissions, Authenticated Users have additional network resource access and Everyone has the most permissions and can access all resources on the computer.

Is the failed-login attempt counter (set in the Account Lockout Policy) reset after a successful login?

Is the failed-login attempt counter (set in the Account Lockout Policy) reset after a successful login?

Yes, the failed-login attempt counter is reset after a successful login in the Windows Operating System. The Account Lockout Policy is a security feature that prevents unauthorised access to a user account by locking it after a specified number of failed login attempts. Once the account is locked, the user cannot log in until the lockout period expires or an administrator manually unlocks the account. Once the user successfully logs in, the failed-login attempt counter is reset to zero, and the lockout period begins again.

When the user’s password is required to unlock the computer locked by a screen saver or with the use of Windows + L key, are the incorrect password inputs treated as failed-login attempts?

When the user’s password is required to unlock the computer locked by a screen saver or with the use of Windows + L key, are the incorrect password inputs treated as failed-login attempts?

Yes, when a user's password is required to unlock a computer that is locked by a screen saver or with the use of the Windows key + L, incorrect password inputs are treated as failed-login attempts on Windows Operating System. This applies to both local and domain user accounts. If the number of failed login attempts exceeds the threshold specified in the Account Lockout Policy, the account will be locked and the user will not be able to log in until the lockout period expires or an administrator manually unlocks the account.

It's important to note that the number of failed login attempts is tracked and counted regardless of whether the computer is locked by a screen saver or with the Windows key + L. This is a security feature that helps prevent unauthorised access to the computer and user accounts.

How is auditing "logon events" different from auditing “account logon events" in Windows Audit Policy settings?

How is auditing "logon events" different from auditing "account logon events" in Windows Audit Policy settings?

In the Windows Audit Policy settings, auditing "logon events" and "account logon events" are two different types of events that are tracked and logged.

Auditing "logon events" refers to the tracking and logging of events related to user logons and logoffs on a local or remote computer. These events include logon and logoff events, such as when a user successfully logs on to a computer, when a user logs off, when a user disconnects from a computer, and when a user reconnects to a computer.

Auditing "account logon events" refers to the tracking and logging of events related to user authentication on a domain controller. These events include logon events such as when a user successfully logs on to a domain, when a user logs off from a domain, and when a user connects or disconnects from a domain.

In summary, Auditing "logon events" tracks and logs events related to user logons and logoffs on a local or remote computer, while Auditing "account logon events" tracks and logs events related to user authentication on a domain controller.

PROCESS SECURITY

What mandatory label is given to objects created by a process operating with a high level mandatory (e.g., by an administrator)?

What mandatory label is given to objects created by a process operating with a high level mandatory (e.g., by an administrator)?

Objects created by a process operating with a high level of mandatory access control (MAC) are typically labeled with a "high" or "system" level label. This indicates that the objects have a higher level of security and access to them is restricted to users with appropriate clearance or privileges.

What does "AppContainer" mean in the integrity level field?

What does "AppContainer" mean in the integrity level field?

The "AppContainer" integrity level refers to a security feature in the Windows operating system that allows for the isolation of applications from one another and from the rest of the system. When an application is run under an AppContainer, it is assigned a unique integrity level that is lower than the integrity level of the system, but higher than that of a standard user. This means that the application is restricted in the actions it can perform and the resources it can access, in order to prevent it from compromising the security of the system. An application that runs under an AppContainer will be isolated from other processes running on the same system and will be limited to a specific set of resources and permissions. This helps to protect the system from malicious or compromised applications by restricting their ability to perform actions that could harm the system.

Please find information how MS Windows can bypass the process restrictions (mandatory levels and UAC), e.g. using the Event Viewer tool.

Please find information how MS Windows can bypass the process restrictions (mandatory levels and UAC), e.g. using the Event Viewer tool.

One way that an attacker could potentially bypass process restrictions on a Windows system is by using the Event Viewer tool. The Event Viewer tool is a built-in Windows utility that allows an administrator to view and manage system logs, including security logs. An attacker with access to the Event Viewer tool could potentially use it to gain access to sensitive information or to elevate their privileges on the system. One way an attacker could use the Event Viewer to bypass restrictions would be to look for high-level privileges account credentials stored in the logs and use them to impersonate those account and gain access to restricted areas. Another way an attacker could use the Event Viewer is by using it to launch a process that runs with higher privileges than the attacker's current account. This can be done by locating and executing a scheduled task that runs with elevated privileges, or by creating a new task that runs with elevated privileges. It's important to note that these are just examples of potential ways that an attacker could use the Event Viewer tool to bypass process restrictions, and there are many other methods that could be used as well. Additionally, it's important to keep in mind that these types of attacks can be prevented through proper security measures, such as implementing strong access controls, monitoring system logs, and educating users on how to identify and prevent potential security threats.

What is CreateRemoteThread() system function used for?

What is CreateRemoteThread() system function used for?

CreateRemoteThread() is a system function in the Windows operating system that allows a process to create a new thread in the address space of another process. This function can be used to inject code into a remote process, allowing an attacker to execute code with the same privileges as the targeted process. It takes in the handle of the process in which to create the thread, and a pointer to a structure that contains information about the thread, such as the start address of the thread and any data that needs to be passed to the thread. This function is typically used by attackers in order to inject malicious code into a legitimate process, and then use that process to perform malicious actions on the system. This can include things like stealing sensitive information, creating a backdoor, or elevating the attacker's privileges on the system. Additionally, it can also be used by security researchers for reverse engineering and vulnerability testing purposes. It's important to note that the use of this function can have serious security implications, and it should be used with caution. Proper security measures, such as monitoring system logs and implementing strong access controls, can help to prevent and detect potential attacks that use this function.

END TO END ENCRYPTION

What is the structure of a X.509 certificate? What does a web browser and an e-mail client allow it to be used for?

What is the structure of a X.509 certificate? What does a web browser and an e-mail client allow it to be used for?

A X.509 certificate is a standard format for digital certificates that is used for secure communication on the internet. The structure of a X.509 certificate includes the following fields:

1.Version: The version number of the certificate.

2.Serial Number: A unique number that identifies the certificate.

3.Signature Algorithm: The algorithm used to sign the certificate.

4.Issuer: The entity that issued the certificate.

5.Validity Period: The date range during which the certificate is valid.

6.Subject: The entity that the certificate represents.

7.Subject Public Key Info: The public key of the entity that the certificate represents.

8.Extensions: Additional information included in the certificate, such as the intended use of the certificate and the identity of the certificate authority that issued it.

A web browser and an e-mail client allow the use of X.509 certificate for secure communication by using the certificate's public key to encrypt the data and the private key to decrypt the data. A web browser can use the X.509 certificate to authenticate the identity of a website, and to establish an encrypted connection with the website using the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols. This provides a secure way for users to share sensitive information with the website, such as login credentials and credit card information. An e-mail client can use the X.509 certificate to encrypt and sign email messages, ensuring that only the intended recipient can read the message and that it has not been tampered with during transit. It's important to note that X.509 certificates are issued and verified by a certificate authority (CA) and they can be used to verify the identity of the sender and the integrity of the data.

In an internet browser, display a certificate of any web server. What is the standard of the CA signature inside the certificate? What hash and cipher algorithms were used?

In an internet browser, display a certificate of any web server. What is the standard of the CA signature inside the certificate? What hash and cipher algorithms were used?

In an internet browser, you can view the certificate of a web server by visiting the website and clicking on the lock icon in the address bar. From there, you can click on the "Certificate" or "Security" tab to view the details of the certificate. The standard of the CA signature inside the certificate is typically X.509. It's a widely used standard for digital certificates that is defined by the International Telecommunications Union (ITU-T) and is used to verify the identity of a website and establish an encrypted connection with the website. The hash and cipher algorithms used in the certificate will vary depending on the website and the specific certificate. It can be SHA-256, SHA-512, RSA, ECDSA, AES-256, etc. These algorithms are used to ensure the integrity and confidentiality of the data exchanged between the browser and the web server. For example, the hash algorithm is used to create a digital signature that verifies the authenticity of the certificate, and the cipher algorithm is used to encrypt the data exchanged between the browser and the web server. It's important to note that these algorithms used in the certificate may change over time and will depend on the security standards and best practices adopted by the certificate authority and the website owner. Additionally, some certificate authorities and website owners may implement additional security measures, such as certificate pinning, to further secure the connection.

Get acquainted with the FKS #12 format of a certificate file.

Get acquainted with the FKS #12 format of a certificate file.

The FKS #12 format is a format for storing digital certificates on smart cards and other cryptographic devices. It is based on the PKCS #12 standard and is used to store the private key of a certificate along with its public key. A FKS #12 certificate file typically contains the following elements:

1.Version: The version number of the certificate file.

2.Authentication Safe: Contains the private key and the public key of the certificate.

3.Algorithm Identifier: The algorithm used for the encryption and decryption of the certificate.

4.Encrypted Data: The private key and public key of the certificate, which are encrypted using the algorithm specified in the Algorithm Identifier.

5.Integrity Check Value (ICV): A value that is used to verify the integrity of the certificate file.

The FKS #12 format is primarily used in Germany and Austria for the storage of digital certificates on smart cards and other cryptographic devices. It is used to store the private key and public key of a certificate, along with additional information such as the algorithm used to encrypt and decrypt the certificate. It's important to note that FKS#12 format is not widely used and it's specific to German and Austrian federal authorities and their partners. And it's a secure format, but it's not compatible with other certificate formats such as X.509.

How is the authenticity of the public key verified in the PG system?

How is the authenticity of the public key verified in the PG system?

In a Public Key Infrastructure (PKI) system, the authenticity of a public key is typically verified through the use of digital certificates. A digital certificate is an electronic document that is issued by a trusted third party, known as a certificate authority (CA), and that contains information about the identity of the certificate holder and the public key associated with that identity. When a user attempts to establish a secure connection with another user or a website, the user's web browser or email client will typically verify the authenticity of the public key by checking the digital certificate associated with the key.

The verification process typically involves the following steps:

1.The browser or client checks the certificate's validity period to ensure that it has not expired.

2.The browser or client checks the certificate's digital signature to ensure that it was issued by a trusted CA.

3.The browser or client checks the certificate's revocation status to ensure that it has not been revoked by the issuing CA.

Additionally, the browser or client will also check that the domain name specified in the certificate matches the domain name of the website or email address being accessed, to ensure that the certificate is being used by the intended party. If the certificate passes all of these checks, the browser or client will consider the public key to be authentic and will use it to establish a secure connection. It's important to note that the authenticity of a public key can also be verified through other means, such as by checking the key's fingerprint, or by verifying the key through a web of trust. However, the use of digital certificates is the most common and widely accepted method for verifying the authenticity of a public key in a PKI system.

How does GPG support revocation of a user's public key certificate?

How does GPG support revocation of a user's public key certificate?

GnuPG, also known as GPG, is a free and open-source implementation of the OpenPGP standard for encrypting and signing data. GPG supports the revocation of a user's public key certificate through the use of key revocation certificates. A key revocation certificate is a special type of certificate that is issued by a user to indicate that their public key certificate is no longer valid. This can happen if the user's private key is lost, stolen, or otherwise compromised. When a user wants to revoke their public key certificate, they can generate a key revocation certificate using their private key and distribute it to their contacts and certificate authorities. Once the key revocation certificate is received, GPG will check the certificate's digital signature to ensure that it was issued by the owner of the key being revoked. If the signature is valid, GPG will mark the key as revoked and will no longer use it for encryption or signing. GPG also allows the user to upload the key revocation certificate to public key servers. This enables other users who have the user's public key to check the key servers and see that the key has been revoked. It's important to note that once a key is revoked, it cannot be used again and the user will have to generate a new key pair. This means that any previously encrypted or signed data that was encrypted with the revoked key will be inaccessible. Additionally, the user will have to distribute the new public key to their contacts and update it on the key servers.

Consider what factors could affect the level of trust in someone's public key in PGP or a similar system (what could increase or decrease the level of trust over time.

Consider what factors could affect the level of trust in someone's public key in PGP or a similar system (what could increase or decrease the level of trust over time.

In a Public Key Infrastructure (PKI) system such as PGP, the level of trust in a public key can be affected by a variety of factors. Some factors that could increase the level of trust in a public key include:

1.Verification of the key's authenticity: If the key's authenticity can be verified through means such as a digital certificate issued by a trusted certificate authority (CA) or through a web of trust, the level of trust in the key will be increased.

2.Key Signatures: If the key has been signed by multiple trusted individuals, it increases the level of trust.

3.Key's usage history: If the key has been used for a long period of time and has been used to encrypt and sign many messages, the level of trust will increase.

4.Regular updates: If the key's metadata is updated regularly and the user is active, it increases the level of trust in the key.

Some factors that could decrease the level of trust in a public key include:

1.Key revocation: If the key is revoked by the owner or a trusted third party, the level of trust in the key will decrease.

2.Key's usage history: If the key has not been used for a long period of time, or has been used infrequently, the level of trust will decrease.

3.Lack of key signatures: If the key has not been signed by any trusted individuals, the level of trust will decrease.

4.Lack of key's metadata updates: If the key's metadata has not been updated for a long period of time, the level of trust will decrease.

It's important to note that the level of trust in a public key is a continuous process and it can change over time. As such, it's important to regularly check the key's trust level and to update the key's metadata and signatures as necessary.

Can some similar multi-level trust criteria be proposed for public key certificates in S/MIME?

Can some similar multi-level trust criteria be proposed for public key certificates in S/MIME?

Yes, similar multi-level trust criteria can be proposed for public key certificates in S/MIME (Secure/Multipurpose Internet Mail Extensions) which is a standard for secure email communication. Some possible trust criteria that can be used to evaluate the trustworthiness of a public key certificate in S/MIME include:

1.Verification of the certificate's authenticity: As with PGP, the certificate's authenticity can be verified through means such as a digital certificate issued by a trusted certificate authority (CA) or through a web of trust.

2.Certificate chain validation: The certificate should be verified by checking the validity of the chain of trust, starting from the root CA and ending with the end-entity certificate.

3.Validity Period: The certificate should be checked for its validity period and should not be expired.

4.Revocation status: The certificate should be checked for its revocation status, and it should not be revoked by the issuing CA.

5.Key usage: The certificate should be checked for the intended usage of the key, such as encryption or signing.

6.Algorithm strength: The certificate should be checked for the strength of the algorithm used, such as RSA or ECDSA, and it should be compliant with the current security standards.

7.Key's usage history: If the key has been used for a long period of time and has been used to encrypt and sign many messages, the level of trust will increase.

8.Regular updates: If the key's metadata is updated regularly and the user is active, it increases the level of trust in the key.

It's important to note that these are just examples of possible trust criteria, and different organizations may have different requirements for evaluating the trustworthiness of a public key certificate in S/MIME. Additionally, the trust criteria may change over time as security standards and best practices evolve.

Do you know about (and maybe use) any other messaging application that supports end-to-end encryption?

Do you know about (and maybe use) any other messaging application that supports end-to-end encryption?

Yes, there are several messaging applications that support end-to-end encryption, in addition to PGP and S/MIME. Some examples include:

1.Signal: This is a messaging application that is focused on privacy and security. It uses end-to-end encryption to protect the communication between users, and it is considered one of the most secure messaging applications available.

2.WhatsApp: This is a widely used messaging application that also supports end-to-end encryption. It uses the Signal Protocol to encrypt messages, calls, photos, and videos, and it also allows users to verify the identity of the person they are communicating with.

3.iMessage: This is the default messaging application for iOS devices and it uses end-to-end encryption for messages, photos, and videos.

4.Telegram: This is a messaging application that is known for its speed and security. It uses end-to-end encryption for "secret chats" feature, which allows users to encrypt their messages, photos, and videos.

5.Wickr: This is a messaging application that is designed for use by businesses and government organizations. It uses end-to-end encryption to protect messages, photos, and videos, and it also allows users to set expiration times for messages.

These are just a few examples of messaging applications that support end-to-end encryption. It's important to note that the level of security and privacy provided by these applications may vary, and users should research and compare the features of different applications before choosing one to use.

RESTRICTED EXECUTION ENVIRONMENT

How are hard and soft limits related?

How are hard and soft limits related?

In computing, hard limits and soft limits refer to the maximum and minimum values that can be set for various system parameters. A hard limit is a maximum or minimum value that cannot be exceeded or reduced, regardless of the actions of a user or a process. For example, a hard limit for the amount of memory that a process can use would prevent the process from using more memory than the specified limit, even if the system has more available. A soft limit, on the other hand, is a value that can be temporarily exceeded but will be enforced again after a certain period of time or when a specific event occurs. For example, a soft limit for the amount of memory that a process can use would allow the process to temporarily use more memory than the specified limit, but the system would eventually enforce the limit and the process would be required to release the excess memory. Hard limits are typically set by system administrators and are intended to prevent users or processes from consuming too many resources or causing other problems on the system. Soft limits, on the other hand, are intended to provide some flexibility for users or processes while still ensuring that resources are used responsibly. It's important to note that hard limits can be changed by the system administrator but soft limits can be changed by the user(if the system is configured to allow it) and it's a good practice to have a combination of both hard and soft limits to ensure that the system is functioning optimally and securely.

Why is it necessary to edit the sudo policy with the dedicated command, visudo?

Why is it necessary to edit the sudo policy with the dedicated command, visudo?

The "visudo" command is used to edit the sudo policy, which controls who is allowed to use the "sudo" command and what actions they are allowed to perform. It is necessary to use the "visudo" command to edit the sudo policy, rather than a regular text editor, because "visudo" provides several important safety features that help to prevent errors and accidental misconfigurations.

1.Syntax checking: When you edit the sudo policy with "visudo", the command checks the syntax of the policy file to ensure that it is correct and well-formed. This helps to prevent errors that could cause the sudo command to malfunction or become ineffective.

2.Exclusive file access: "visudo" opens the policy file in exclusive mode, which means that no other users or processes can access the file while it is being edited. This helps to prevent conflicts and data loss that could occur if multiple users were editing the file simultaneously.

3.Automatic backups: "visudo" automatically creates backups of the policy file before saving any changes. This allows you to easily revert to a previous version of the file if you make a mistake or if something goes wrong.

4.Terminal-based interface: "visudo" runs in the terminal and it allows you to edit the file in a text editor that is built into the command, it's more comfortable and user-friendly for administrators.

In summary, using "visudo" to edit the sudo policy is necessary because it provides important safety features that help to prevent errors and accidental misconfigurations, ensuring the security and stability of the system.

Whose password will be required, if the sudo policy requires a password to run a command: that of the user invoking the command, or that of the target account?

Whose password will be required, if the sudo policy requires a password to run a command: that of the user invoking the command, or that of the target account?

The sudo policy specifies which user's password is required in order to run a command. The two most common options are: The user invoking the command: When the sudo policy is configured to require the password of the user invoking the command, the user will be prompted to enter their own password in order to execute the command. This is the default configuration in most cases. The target account: When the sudo policy is configured to require the password of the target account, the user will be prompted to enter the password of the account that the command will be executed as. For example, if the user is running the command "sudo su - root", they will be prompted to enter the root account's password. This option is less common but it can be useful in certain scenarios. It's important to note that the sudo policy can be configured to require different passwords for different commands or users. For example, it could be configured to require the password of the user invoking the command for some commands, and the password of the target account for others. It's also important to mention that some systems and configurations are configured to not require any password when running a command with sudo, this is called "sudo without a password" and it's considered a security risk, and it's not recommended.

What role play SUID and SGID bits in the case of non-executable files (i.e. non binary-code programs)?

What role play SUID and SGID bits in the case of non-executable files (i.e. non binary-code programs)?

SUID (Set User ID) and SGID (Set Group ID) are two special permissions that can be set on files in Linux and Unix-like operating systems. These permissions are used to control the execution of files, but they also have an effect on non-executable files. The SUID bit, when set on a non-executable file, allows users to access the file with the permissions of the file's owner, rather than their own permissions. For example, if a file is owned by the root user and has the SUID bit set, any user who reads or writes to that file will do so with the permissions of the root user, regardless of their own permissions. This can be useful for allowing users to access files or resources that they would not normally have access to, but it can also be a security risk if not managed carefully. The SGID bit, when set on a non-executable file, allows users to access the file with the permissions of the file's group, rather than their own permissions. This can be useful for allowing users to access files or resources that are shared by a group of users, but it can also be a security risk if not managed carefully.

What are the most important advantages and disadvantages of the above mechanisms?

What are the most important advantages and disadvantages of the above mechanisms?

The SUID and SGID bits are special permissions that can provide certain advantages and disadvantages in controlling file access and execution in Linux and Unix-like operating systems.

Advantages of SUID and SGID bits:

1.Elevated access: The SUID and SGID bits can be used to allow users to access files or resources that they would not normally have access to, by granting them the permissions of the file's owner or group.

2.Shared access: The SGID bit can be used to allow a group of users to access shared resources, by granting them the permissions of the file's group.

3.Increased security: The SUID and SGID bits can be used to enhance security by allowing specific users or groups to access sensitive files or resources, while denying access to other users.

Disadvantages of SUID and SGID bits:

1.Security risks: The SUID and SGID bits can create security risks if not managed carefully, by allowing users to access files or resources that they should not have access to.

2.Complexity: The use of SUID and SGID bits can increase the complexity of file permissions and access control, making it more difficult to understand and manage.

3.Vulnerabilities: The SUID and SGID bits can be exploited by attackers to gain unauthorized access to files or resources, if vulnerabilities exist in the software that is being executed with elevated permissions.

4.Unintended consequences: The use of SUID and SGID bits can lead to unintended consequences, if a change in the system, user or group causes the files and resources to become accessible to an unintended user or group.

It's important to note that the use of SUID and SGID bits should be carefully considered, and they should be used only when necessary and with caution, as they can create potential security risks if not managed properly.

Would it be worth to expand any of these mechanisms? What functionality should be added?

Would it be worth to expand any of these mechanisms? What functionality should be added?

SUID and SGID bits are useful mechanisms for controlling file access and execution in Linux and Unix-like operating systems, but they do have some limitations. Some possible ways to expand these mechanisms could include:

1.Fine-grained access control: The ability to set specific permissions for different users or groups on a per-file basis, rather than relying on the file's owner or group. This would allow for more flexible and fine-grained access control.

2.Auditing and logging: The ability to track and log changes to files or resources that have the SUID or SGID bits set, in order to monitor and detect any potential security risks.

3.Dynamic permissions: The ability to change the permissions of a file or resource based on certain events, such as the user's location or the time of day. This would allow for more dynamic and adaptive access control.

4.Role-based access control: The ability to assign roles to users or groups, and then assign permissions to those roles, rather than to individual users or groups. This would allow for more efficient and flexible access control.

5.Conditional access: The ability to grant access to a file or resource based on certain conditions, such as multi-factor authentication or a successful vulnerability scan. This would add an extra layer of security to the access control.

It's important to note that expanding these mechanisms would increase the complexity of the system and could add more attack surface. Additionally, it's important to evaluate the trade-offs between adding more functionality and the increase of security risks.

POSIX ACL

What is the necessary condition that lets you change POSIX ACL permissions to a given object?

What is the necessary condition that lets you change POSIX ACL permissions to a given object?

In order to change POSIX Access Control Lists (ACLs) permissions to a given object, the user must have the appropriate permissions on the object itself and on the parent directory where the object is located.

1.The user must have the write permission on the parent directory where the object is located. This allows the user to make changes to the directory structure, such as adding, modifying, or deleting files.

2.The user must have the write permission on the object itself. This allows the user to make changes to the object's metadata, such as the permissions or ownership.

3.The user must have the execute permission on the parent directory where the object is located. This allows the user to access the directory and its contents.

It's important to note that the exact permissions required to change POSIX ACLs may vary depending on the specific implementation and configuration of the system. Additionally, some systems may have additional security mechanisms in place, such as mandatory access controls or role-based access controls, that may further restrict the ability to change POSIX ACLs. It's also important to mention that some systems may have a dedicated user or group that can change the permissions, such as the superuser or the root, this is determined by the system's configuration.

FILESYSTEM SECURITY: NTFS

What permissions are hidden behind "Special Permissions" seen among basic ACL rights?

What permissions are hidden behind "Special Permissions" seen among basic ACL rights?

In Windows, the "Special Permissions" option in the Access Control List (ACL) rights allows for granular control over specific actions that can be performed on a file or folder. These include the ability to take ownership of the file or folder, the ability to change permissions on the file or folder, the ability to change auditing settings, the ability to change the owner, and the ability to change the discretionary access control list (DACL). Additionally, it also include the ability to perform certain operations on the file or folder such as execute, write, read, and delete.

How are the active rights determined? Which authorization operations grant or denial- have priority in access control? Which permissions inherited or explicitly granted have priority?

How are the active rights determined? Which authorisation operations grant or denial- have priority in access control? Which permissions inherited or explicitly granted have priority?

In Windows, the active rights for a user or group on a file or folder are determined by evaluating the Access Control List (ACL) for that file or folder. The ACL is a list of permissions that are associated with the file or folder, and it contains a list of access control entries (ACEs) that specify the permissions that are granted or denied to users or groups. When a user or group attempts to access a file or folder, the operating system compares the user or group's security identifier (SID) to the SIDs in the ACEs in the ACL. If there is a match, the operating system applies the permissions specified in the ACE. If there is no match, the operating system applies the permissions inherited from the parent folder. The priority of authorization operations in access control is as follows:

1.Deny ACEs take precedence over allow ACEs.

2.Explicitly defined ACEs take precedence over inherited ACEs.

3.ACEs that are applied to the user take precedence over ACEs that are applied to a group that the user is a member of.

Inherited permissions take precedence over explicitly granted permissions when both permissions apply to the same user or group and the permissions are of the same type.

How can you change permissions for several files at once?

How can you change permissions for several files at once?

In Windows, you can change permissions for multiple files at once using the command line or a third-party tool.

Using Command line:

1.Open the Command Prompt.

2.Navigate to the folder that contains the files for which you want to change permissions.

3.Type the command "*icacls * /grant [username]:[permissions]*" and press Enter.

where * is to target all files and folders in the current directory, [username] is the name of the user or group you want to give permissions to, and [permissions] is the permissions you want to give (ex. "F" for full control, "RX" for read and execute, etc.).

4.You can also use the option /deny to deny permissions.

Using Third-Party tool: A third-party tool like "Advanced Security Settings" or "ACL Tool" can also be used to change permissions for multiple files at once. These tools provide a graphical user interface that makes it easy to manage permissions, and they also allow you to copy permissions from one file or folder to another. Note: Be cautious when changing permissions on multiple files at once, as it can cause unintended consequences if not done correctly.

Is there any way to copy the ACL from one file to another?

Is there any way to copy the ACL from one file to another?

Yes, there is a way to copy the Access Control List (ACL) from one file to another in Windows. You can use the command line tool "icacls" to accomplish this.

Here are the steps to copy the ACL from one file to another:

1.Open the Command Prompt.

2.Navigate to the folder that contains the file from which you want to copy the ACL.

3.Type the command "icacls [source file] /save aclfile" and press Enter.

where [source file] is the file you want to copy the ACL from and "aclfile" is the file where you want to save the ACL.

4.Navigate to the folder that contains the file to which you want to apply the ACL.

5.Type the command "icacls [destination file] /restore aclfile" and press Enter.

where [destination file] is the file you want to apply the ACL to and "aclfile" is the file where you saved the ACL.

By using this method, all the ACEs from the source file's ACL will be copied to the destination file's ACL, including all explicit and inherited permissions, and also the permissions from the parent folder. Another alternative is using the command xcopy /o to copy the files, it copies the security attributes along with the files. It's worth noting that copying the permissions using this method will only work if the destination folder is on the same filesystem as the source folder, otherwise the inherited permissions won't be copied.

What does WDAC permission (available with the icacls command) mean?

What does WDAC permission (available with the icacls command) mean?

WDAC (Windows Defender Application Control) permission is a feature of the Windows operating system that allows administrators to control which applications can run on a computer. This is done by creating a policy file that specifies which applications are allowed to run. The policy can be enforced using icacls, a command-line utility that sets access control lists (ACLs) on files and directories. By using this feature, administrators can limit which applications can be used on the computer, ensuring that only trusted programs are running. This provides an additional layer of security and helps protect the computer from malicious software or malware.

How does NTFS Bypass Traverse Checking work?

How does NTFS Bypass Traverse Checking work?

NTFS Bypass Traverse Checking is an advanced security feature of the Windows NTFS file system. It restricts certain operations on a file or directory, such as opening, writing, or deleting. This is done by checking the access control lists (ACLs) associated with each file or directory. If a user has the correct permissions, they can bypass the traverse checking, allowing them to perform operations on files or directories they would otherwise not be able to access. This feature is important for ensuring that users are only able to access files or directories they are allowed to, thus providing an additional layer of security.

How does EFS encrypt the contents of a file: symmetrically or asymmetrically? Using which key? Where is this key stored and how is it secured?

How does EFS encrypt the contents of a file: symmetrically or asymmetrically? Using which key? Where is this key stored and how is it secured?

The Encrypting File System (EFS) uses symmetric encryption to protect the contents of a file. It uses a data encryption key (DEK) to encrypt the file and this DEK is stored and encrypted with a File Encryption Key (FEK). The FEK is generated for each file and is stored on the user's local computer. The FEK is secured by the user's logon credentials, which ensures only the user with the correct credentials can access the file.

Can the EFS mechanism encrypt a file on the FAT filesystem?

Can the EFS mechanism encrypt a file on the FAT filesystem?

Yes, the Encrypting File System can be used to encrypt files stored on the FAT filesystem. This is done by converting the file to be encrypted into the NTFS format, which is the only format the EFS can use. After the file has been encrypted, it will then be stored on the FAT filesystem in its encrypted form. This ensures that the file can only be accessed by the user who has the correct credentials to decrypt the file.

What do EFS certificates contain?

What do EFS certificates contain?

Encrypting File System (EFS) certificates in Windows operating systems contain a public key and a private key. The public key is used to encrypt the data, while the private key is used to decrypt it. The public key is stored in the EFS certificate, while the private key is stored in the user profile. EFS certificates are typically created by the user when they enable encryption. The certificate is then used to encrypt and decrypt files, folders, and even entire drives.

How to share an encrypted file in Windows?

How to share an encrypted file in Windows?

To share an encrypted file in Windows, you must first encrypt the file using the Encrypting File System (EFS). To do so, right-click on the file or folder you want to encrypt, select "Properties," and then click on the "Advanced" button. Once the Advanced Attributes window opens, select the "Encrypt contents to secure data" checkbox and then click "OK." After that, you will be able to share the file with anyone. However, the recipient will need the encryption key to decrypt the file. To share the encryption key, you can either print it, save it to a USB drive, or send it via email.

What is the rekeywiz system tool used for? Can the same results be achieved with cipher command as well?

What is the rekeywiz system tool used for? Can the same results be achieved with cipher command as well?

Rekeywiz is a system tool used to quickly change the Encrypting File System (EFS) encryption keys for multiple files and folders. The tool is designed to reduce the time and effort needed to re-encrypt a large number of files. The same results can be achieved using the cipher command, but it can take longer and is more difficult to use.

SECURE SHELL

What encryption algorithm is the user's private key file encrypted with?

What encryption algorithm is the user's private key file encrypted with?

In Linux OS, the user's private key file is usually encrypted with a symmetric encryption algorithm, such as AES. This symmetric encryption algorithm offers strong security and is the most commonly used algorithm for encrypting private key files. The encryption key used for this encryption process is determined by the user and is typically a long, random passphrase that is difficult to guess or brute-force.

Is it necessary to enter the password (passphrase) protecting this file with every access attempt?

Is it necessary to enter the password (passphrase) protecting this file with every access attempt?

Yes, it is necessary to enter the password (passphrase) protecting the file with each access attempt. This helps ensure that only the user who knows the correct passphrase can decrypt and access the contents of the file. Without the correct passphrase, even if someone were to gain access to the encrypted file, they would not be able to open it.

WINDOWS NETWORK SECURITY

What does IPC$ share mean, and what is it used for?

What does IPC$ share mean, and what is it used for?

IPC$ stands for Inter-Process Communication, and it is a share created by the Windows operating system. This share is used to facilitate communication between two or more programs and processes. It can also be used to connect to a remote computer and exchange files, printer information, and other data. It is important to note that IPC$ does not provide access to the file system of the remote computer.

How does remote access to a network share work for a user whose local account (on the system providing the share) does not have a password?

How does remote access to a network share work for a user whose local account (on the system providing the share) does not have a password?

In Windows OS, remote access to a network share for a user whose local account does not have a password is possible through the Guest account. The Guest account is a special account that is enabled by default on Windows systems, allowing users to connect to a network share without authentication. Once the user has connected to the network share using the Guest account, they can access the resources on the network share without needing to provide any login credentials.

What do the netshare, netview, netuse, netfile(netfiles), and netsession (net sessions) commands allow you to do?

What do the netshare, netview, netuse, netfile(netfiles), and netsession (net sessions) commands allow you to do?

The `netshare`, `netview`, `netuse`, `netfile`, and `netsession` commands in Windows OS allow you to administer and manage network shares and sessions. The netshare command is used to create, delete, and configure network shares, while the netview command can be used to view the list of network computers and resources. The netuse command allows you to map network drives, view current connections, and disconnect network drives. The netfile command is used to view and manage open files on

In Windows Firewall, why do all the predefined rules have an "Allow" action set up?

In Windows Firewall, why do all the predefined rules have an "Allow" action set up?

By default, Windows Firewall enables all incoming network traffic from trusted networks and only blocks traffic from unknown and potentially malicious sources. This is done by setting the predefined rules to "Allow" by default. This is beneficial for users as it allows the user to connect to the Internet, other computers on the network, and other services without any additional configuration. The user can then customize the Windows Firewall settings to block certain traffic or even add additional rules.

FIREWALLS

Does the iptables tool allow any form of communication encryption/obfuscation?

Does the iptables tool allow any form of communication encryption/ obfuscation?

The iptables tool does not provide any form of communication encryption or obfuscation. However, it does allow the user to control the flow of data packets by creating rules for packet filtering, port forwarding, source address translation, and other packet manipulation. It also supports protocols such as ICMP, IPv4, and IPv6, so it can be used to control incoming and outgoing data traffic on a network. Additionally, iptables can be used to block specific types of traffic, making it an effective tool for preventing unwanted malicious traffic from entering the network.

WINDOWS IPSEC

The IPsec protocol inherits from the IP protocol connectionless semantics of communication, but it uses some mechanisms known typical for connection-oriented protocols, such as sequence numbers, for instance. Does the IPsec association use any kind of sliding window mechanism (as in TCP)?

The IPsec protocol inherits from the IP protocol connectionless semantics of communication, but it uses some mechanisms known typical for connection-oriented protocols, such as sequence numbers, for instance. Does the IPsec association use any kind of sliding window mechanism (as in TCP)?

No, the IPsec association does not use any kind of sliding window mechanism like TCP does. Instead, it uses a cryptographic key exchange protocol that is based on the Diffie-Hellman algorithm. This protocol requires two peers (the initiator and the responder) to securely exchange keying material and negotiate session parameters. The negotiated parameters are then used to establish an authenticated and encrypted tunnel between the two peers. This tunnel can be used to send and receive data securely.

SESSION-LAYER VPN

Can a VPN client be authenticated by username and password?

Can a VPN client be authenticated by username and password?

Yes, a VPN client can be authenticated by username and password. This is a common authentication method used by VPNs, where the user must enter their credentials into the VPN client in order to connect to the VPN server. The VPN server will then authenticate the user's credentials and establish a secure tunnel for the client to access the VPN network. This is a secure and convenient way to authenticate users, as it ensures only authorized users have access to the VPN network.


thank you
feel free to contact me at
jakub.hamerlinski@cs.put.poznan.pl