

# IY5512: Coursework 7b: Worked solutions

---

1. *It is often said that Unix is inherently less subject to attack than Windows as a result of flaws in software. Is this true, and if so why? Does this mean that Unix is completely secure, regardless of how well written it is?*

There are almost as many answers to this question as there are security experts in the world. Indeed, it is often difficult to distinguish between facts and opinions on this issue.

Some interesting discussions of this point can be found here:

<http://www.kernelthread.com/publications/security/uw.html>

here:

<http://www.esecurityplanet.com/trends/article.php/3933491/Is-Linux-Really-More-Secure-than-Windows.htm>

and here:

<http://www.windowsecurity.com/uplarticle/18/nt-vs-unix.pdf>

2. *In Windows, why do SIDs need to be unique within a domain forest?*

Within a domain forest, there is complete bidirectional trust between domains. An account within any domain in the forest can be authenticated by any other domain in the forest. Consequently, following a single successful logon process, a user with the proper permissions can access resources in any domain in the forest. As a result SIDs need to be globally unique within a forest.

3. *Describe a vulnerability in the Kerberos authentication protocol.*

In the exchange between a client and the Authentication Server (AS), a message is sent from the AS to the client encrypted using the user's long term secret (shared by the user and the AS). This secret key is often derived from a human-memorised password. This allows any party which intercepts this message to conduct a brute force password search, as follows.

Each possible password is used in turn to derive a key (using the standard process). This key is then used to decrypt the intercepted message. If the decrypted result has the correct syntax, then the password used to derive the key is the user's password (almost certainly).

4. *Give an example of a Windows security principal, and describe the relationship between a Security Identifier (SID) and a security principal.*

A user is one example of a security principal. Groups are also examples of security principals.

A security identifier (SID) is issued when an account is created for a security principal. Windows uses SIDs to uniquely identify security principals. SIDs are used in access tokens, security descriptors and access control entries. A SID identifies an entity within a hierarchical namespace using a sequence of one or more identifying authorities.

5. *In Windows, a securable object possesses a security descriptor. Give two examples of securable objects. Briefly describe one of the functional components of a security descriptor.*

Securable objects have a security descriptor that contains security attributes. Examples of securable objects include files, user objects and registry keys.

The security descriptor contains:

- the group to which the object belongs – a SID,
- the owner of the secured object – also a SID,
- a discretionary access control list (DACL) that controls the users and groups that can access the object, and
- a system access control list (SACL), used for auditing purposes.

6. *Describe the main steps that occur in a Windows interactive login procedure when a user is logging in to a domain account. In your description cover the roles of Kerberos, the Winlogon process, the LSA, Active Directory, the SAS, a GINA, and the creation of an access token.*

A user logged on to a domain account benefits from 'single sign-on' within the domain. Network authentication is performed automatically (based on the domain's prior authentication of the user) whenever a user requests a network service. Windows uses Kerberos v5.0 as the default authentication mechanism for interactive logons and network authentication. A Kerberos key distribution centre (KDC) is installed on every domain controller. Every Windows machine includes a Kerberos 5.0 client.

The user enters credentials based on a shared secret, usually username and password. A user can also use biometric identification, such as fingerprints or retinal scans or a hardware token. Interactive logon uses the Winlogon process, one or more authentication packages, the local security authority (LSA), and a repository of user information (SAM (in the case of a local computer account) or Active Directory (in the case of a domain account)).

Winlogon is the only process that intercepts logon requests from the keyboard. Login requests are initiated by the secure attention sequence (SAS). The default SAS is Ctrl+Alt+Del, and is used to provide a trusted path to the operating system to ensure that other (malicious) applications cannot capture user passwords. The Winlogon process passes control to a GINA. A GINA (graphical identification and authentication) DLL (dynamic link library) is a suite of pre-compiled functions and

procedures that collect authentication information from the user. The GINA determines the logon sequence in Windows. The standard Windows logon dialog box is part of the default GINA (`Msgina.dll`).

If the user is attempting to log on to a domain account, the LSA interacts with the Kerberos authentication package and interrogates Active Directory to confirm the logon credentials.

If a Kerberos client cannot locate a domain controller, the LSA checks for locally cached credentials from a previous logon (which are used to authenticate the user with the `MSV1_0` authentication package). If authentication is successful the LSA creates an access token for the user.

An access token includes the SID of the authenticated user, the SIDs of the groups to which the user belongs, the SID of the default owner of any new objects the user creates (typically the SID of the user), and a list of privileges assigned to the user.

7. *Outline at a high level the pieces of information used by Windows in making an access control decision when a subject wishes to access an object.*

Almost any access control mechanism will involve comparing three main pieces of information:

- the security information of the subject (e.g. a process), typically inherited from the user;
- the security information of the object (e.g. a file); and
- the type of access being requested, typically a set of operations specified in the form of some kind of access mask.

In Windows:

- the security information of the subject (including the user's SID and the SIDs of the groups to which the user belongs) is held in the *access token* generated at logon;
- the security information of the object is held in the *security descriptor* generated when the object is created, where the security descriptor contains an access control list (the *DACL*) and ACEs in the DACL refer to SIDs; and
- the type of access being requested is specified in the *Requested Access Mask*.