**Week 2: Windows**

**Windows Security Architecture**

Local vs Domain Accounts

UNIX has users and groups. UNIX subjects are applied to one machine.

* Workgroup-collection of computers connected using one network. No domain controller in a workgroup, authentication performed at each computer.
* A networked Windows system can have two configurations – either domain joined or workgroup. In a domain joined computer, users can access accounts using centrally managed Active Directory.
* Users can also login using local account, but local accounts will not have access to domain resources - networked printers, Web servers, e-mail servers.
* In a workgroup – local accounts managed by SAM are used.
* Security policies can be centrally managed using AD.

Security Reference Monitor (SRM)

* access checks,
* generates audit log entries, and
* manipulates user rights, also called privileges.

Local Security Authority (LSA) – lsass.exe

* Issues security tokens to accounts
* Password policy, such as complexity rules and expiration times
* Auditing policy, or which operations on what objects to audit
* Privilege settings, or which accounts on a computer can perform privileged operations.

Security Account Manager (SAM)

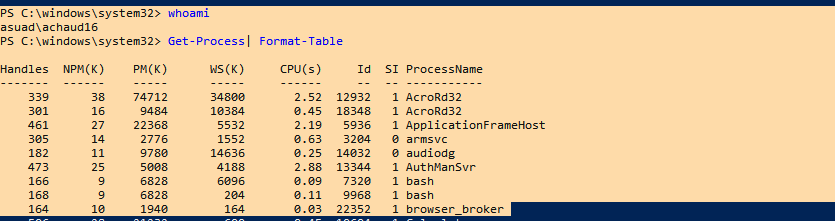
* Account data and relevant security information about local principals and local groups
* User Login 🡪 SAM process (SamSrv) takes logon information and performs lookup in SAM DB in Windows System 32\config
* Passwords stored as MD4, PBKCS

Active Directory (AD)

* MS LDAP directory for security operations and account logon.
* All currently supported client versions of Windows, including Windows XP and Windows 7, can communicate with AD to perform security operations including account logon
* A windows client will authenticate using AD when user logs on to the computer using domain account.

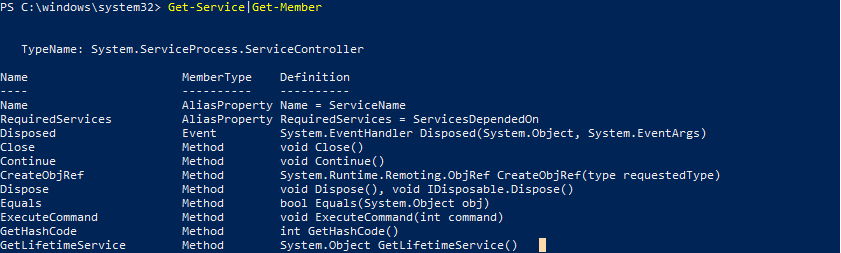
Powershell (PS)

* Flexible scripting language based on .NET framework.
* Rich access to Windows computers and security settings.
* Commands in PS are called cmdlets – consistent (verb-noun) syntax.
* PS supports command piping. PS pipes objects not text. Allows rich data processing, filtering, and analysis.
* whoami
* Get-Process | Format-Table



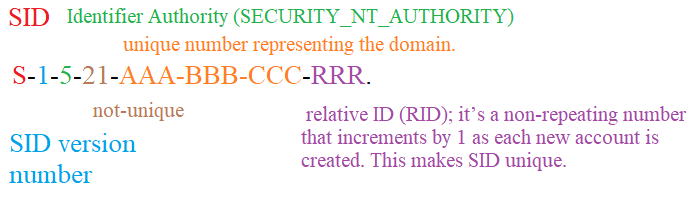
* Get-Process -name chrome | Stop-Process
* Get-Service | Get-Member

List of object methods and properties.



What happens when a user logs on to a Windows system – End to end domain Example

* Domain admin should add user’s account information to the system before he can log on (username, account name – domain specific, and password).
* Windows creates and account in domain controller running AD. Each account has unique Security ID (SID) – unique to domain. E.g. - S-1-5-21-AAA-BBB-CCC-RRR, S-1-5-21-123625317-425641126-188346712-2895.
* If you create account “Mike”, delete and re-create “Mike”, they are two totally different accounts because they will have different SID.
* Once user log’s in token SID is generated by OS and assigned to user.
* Token contains user’s SID, group membership information, and privileges.



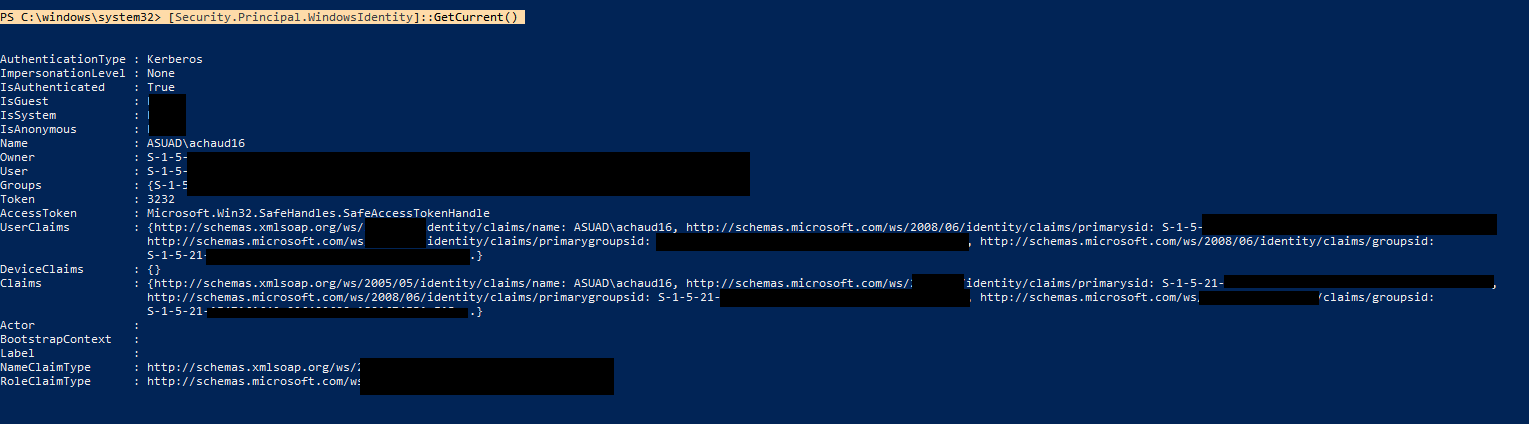
* On a domain-joined computer (we’ll use the ‘Marketing’ domain), it is possible for a user to logon to a local account by using the “.” domain.
* So rather than using “Marketing\Paige” or just “Paige” Paige can use “.\Paige” assuming there is a local Paige account on the computer. The “.” will substitute the machine name as the workgroup name.

Username Format

* SAM format - Domain\Username.
* User Principal Name (UPN) – [username@domain.company.com](mailto:username@domain.company.com).
* “Mike’s” PC in ASUAD domain, he can login using “ASUAD\Mike” or [Mike@asu.edu](mailto:Mike@asu.edu).

Admin Accounts and Blank Password

* Setting blank password applies to only local accounts, not to domain accounts,
* Remote access from one Windows computer to another using an account that is a member of the local Administrators group can only be performed if the account has a password.
* Access is denied when using a non-password admin account remotely.
* you can dump information about the current logged on user with this line:
* :[Security.Principal.WindowsIdentity]::GetCurrent()



Privileges in Windows

Systemwide permissions assigned to user accounts, e.g.,

* ability to backup computer (bypasses all access to perform complete backup),
* change time (can cause Kerberos authentication fail and lead to erroneous data being written to logging system).
* More than 45 privileges – some deemed dangerous.
* Act as part of OS privilege – account with this privilege can run code as part of trusted OS code. Granted only to Local system account – most dangerous privilege in Windows.
* Debug program privilege - allows debugging of processes running in Windows, user can run any code he wants in any process using this privilege.
* Backup files and directories privilege – process with this privilege can bypass Access Control List (ACL) checks, and read all files for complete backup.
* Restore files and directory privilege is also able to bypass ACL.
* Bypass traverse checking – assigned to all user accounts by default – used as NTFS file system optimization. Deemed benign.