Name: - Sooraj B

User I'D: - 34753

E-mail I'D: - rajsooraj318@gmail.com

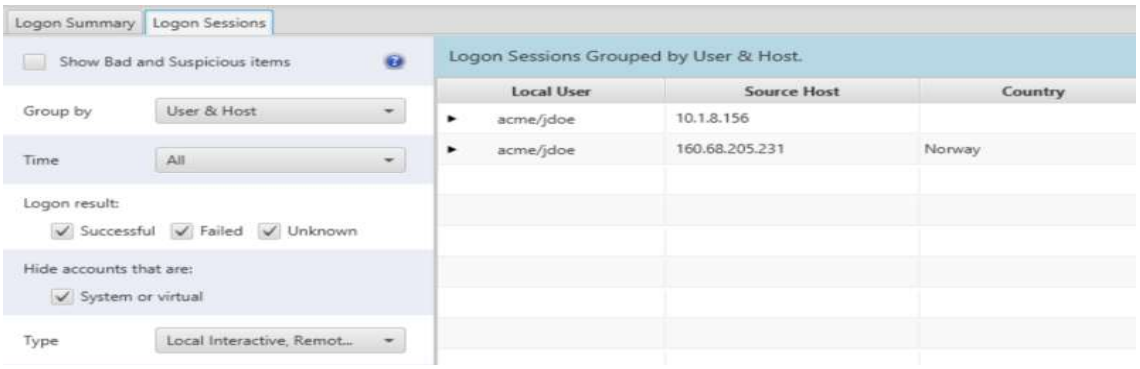
Assignment Topic: - Mention all windows tools for debugging with Screenshots and steps to create for Microsoft Intune Portal

Understanding Windows Debugging Tools (Sys internals Suite)

Today's session gave me a solid understanding of how powerful Sys internals tools are for troubleshooting Windows systems. Here's how I grasped each tool:

1.Logon Sessions:

- **Purpose:** Lists active user logon sessions on a Windows system.
- **Tool:** LogonSessions.exe (Sys internals Suite).
- **Use Case:**
  - Audit active sessions for security compliance.
  - Detect unauthorized logins (e.g., after-hours access).



This tool lists all active user logins on a system. It's useful for IT admins to track who's logged in, especially for security checks or auditing. Running LogonSessions.exe in the command line gives a quick snapshot of sessions, which helps in identifying unauthorized access.

2.Auto logon:

Purpose: Automates Windows login without manual password entry.

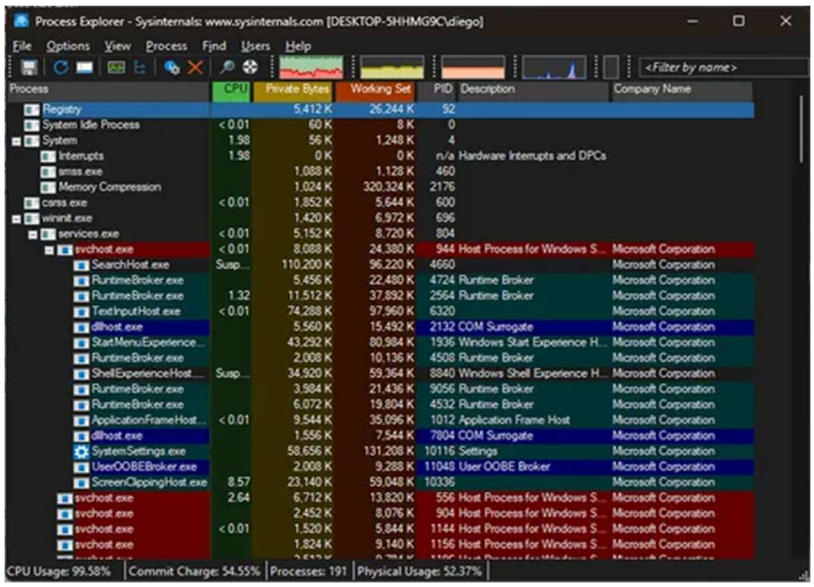
- **Tool:** GUI-based (Sysinternals).
- **How It Works:**
- Stores encrypted credentials in HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.
- **Use Case:**
- Kiosks, automated testing environments.
- **Security Note:** Passwords are encrypted but should be used cautiously.

A handy tool for automating Windows logins without manual password entry. It securely stores credentials in the registry under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Win logon. While great for test labs or kiosks, I learned it should be used cautiously in production due to security risks, even though passwords are encrypted.

3.Process Explorer:

- **Purpose:** Advanced process monitoring (replaces Task Manager).
- **Features:**
- Tree-view of parent/child processes.

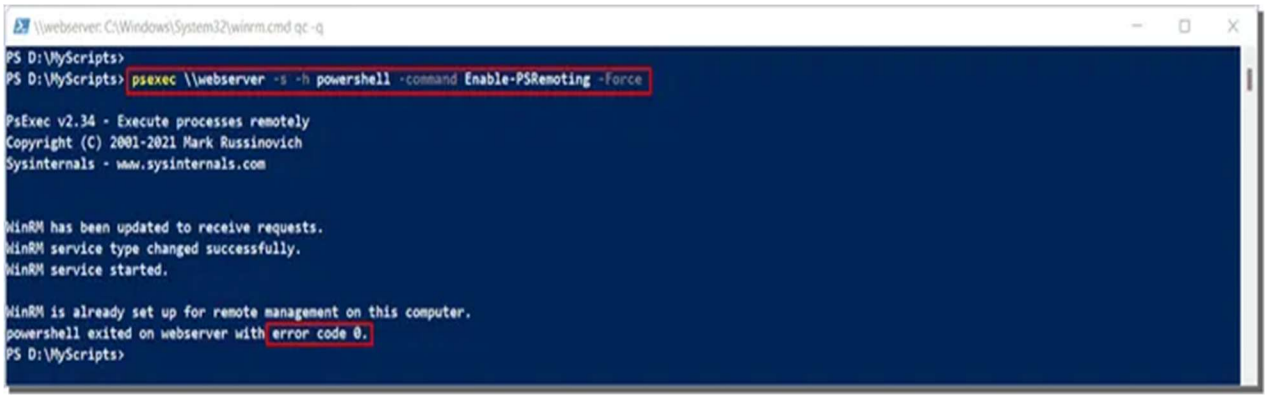
- DLL/file/registry access per process.
- **Use Case:**
- Identify malware (e.g., injected DLLs).
- Debug application crashes.



It shows detailed process hierarchies, loaded DLLs, and even file/registry activity for each process. I can see myself using this to diagnose slow systems or detect malware hiding in processes.

4.PS Exec :

- **Purpose:** Execute commands remotely.
- **Syntax:** PsExec \\RemotePC -u User -p Password command.
- **Use Case:**
- Deploy scripts across multiple machines without RDP.

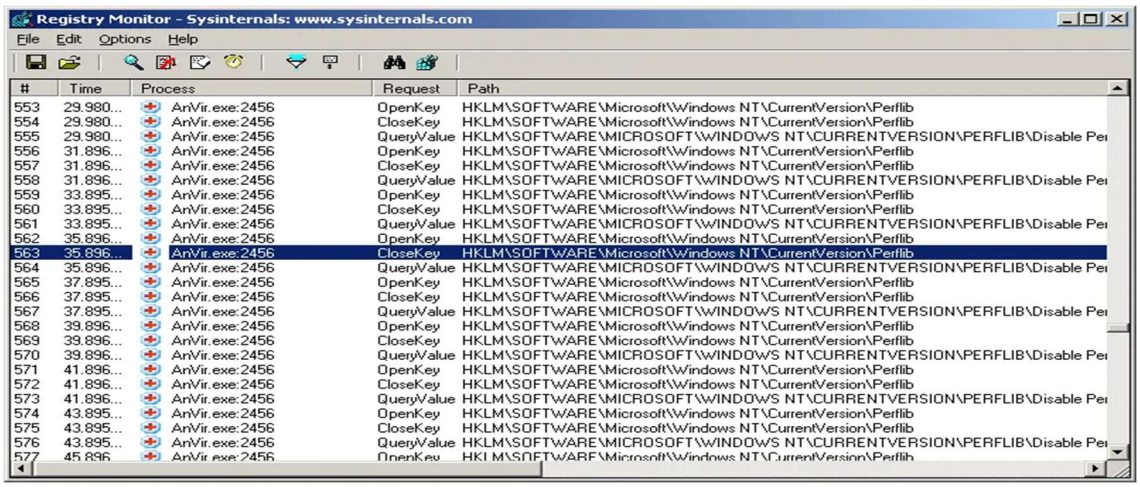


- These are game-changers for remote administration. With PS Exec, I can run commands on another PC remotely, like restarting a service or deploying a script. The PS Tools suite (PS Kill, PS Shutdown, etc.) makes managing multiple machines easier—no need for manual logins.

6. RegMon (Legacy)

- **Replaced by:** Process Monitor (ProcMon).
- **Use Case:**

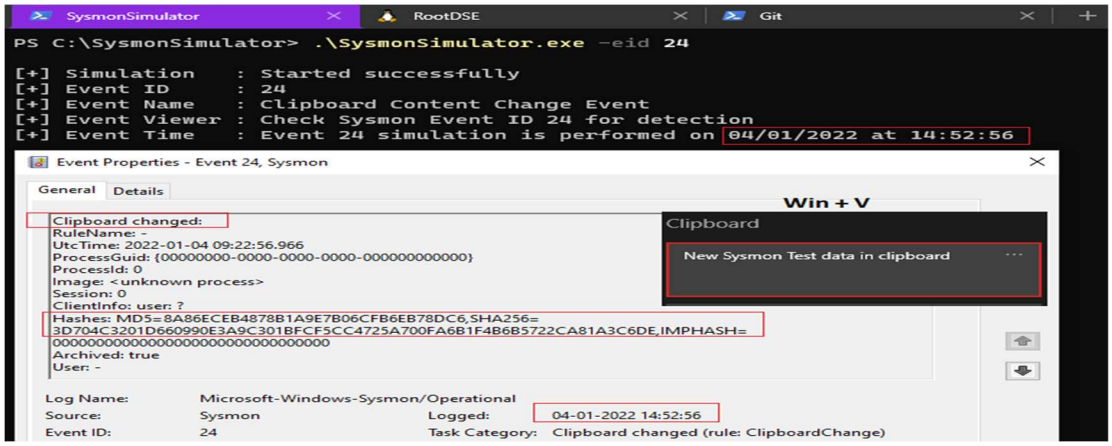
- Track registry changes during software installations.



7.Sysmon

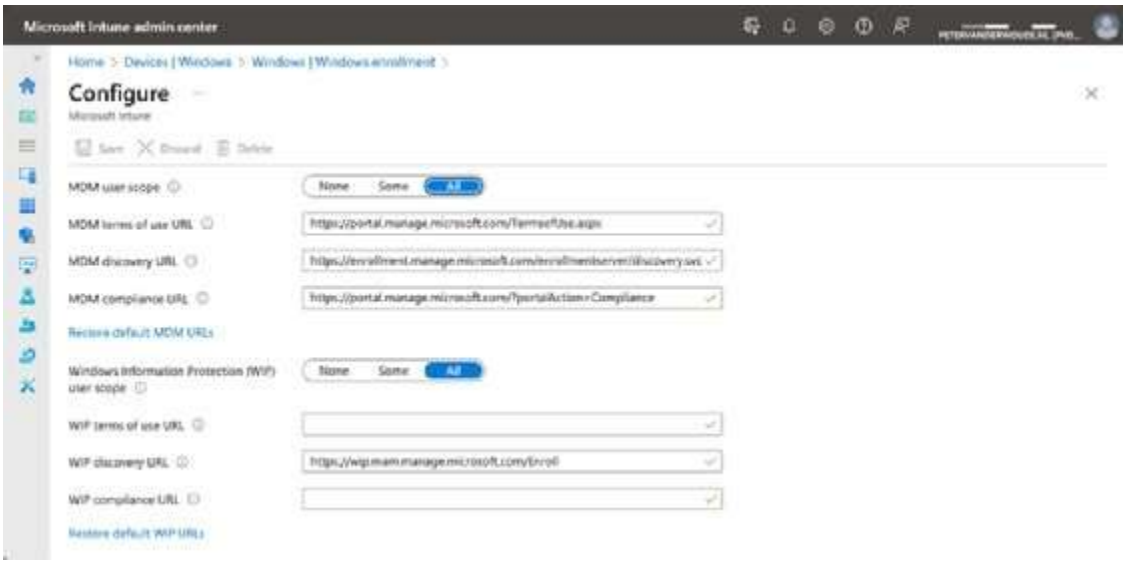
- **Purpose:** Log system events (e.g., process creation).

- **Configuration:** XML-based rules for event tracking.



- **Sysmon**: This tool logs system events (like new processes or network connections) to the Windows Event Log. It's a must for security teams to detect suspicious activity, like unexpected processes running in the background.

Intune Enrollment & Device Management



I also learned how to set up and use Microsoft Intune for managing devices:

- **Enrolling in the Free Trial**: Signing up is straightforward—just visit the Intune trial page, use a work/school email, and follow the prompts. Existing Azure AD users might need extra license assignments.
- **Syncing Devices**: After assigning apps or policies, devices need to sync to apply changes. This can be done through the Company Portal app (Settings > Sync) or directly from the Intune Admin Center under Devices > Select Device > Sync. For Windows, it can even be done via the taskbar's Company Portal icon.

Application Packaging & Registry Insights

We covered key concepts in MSI packaging and registry management:

- **Active Setup Versioning**: This ensures user-specific setups run during login. By tweaking the version number in HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{GUID}, I can force setups to rerun for new users or after updates.
- **Intune win Conversion**: To deploy apps via Intune, they must be packaged as. Intune win files. The IntuneWinAppUtil.exe tool helps convert installers, but silent install commands (like msi exec / i app.msi /qn) are a must.
- **Tracking Installations**: The registry keys under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall (for all users) and HKCU\... (per-user) store app details. Each app has a GUID, which can be used to silently uninstall it (msiexec /x {GUID} /QN).

Interactive vs. Non-Interactive Apps

I now understand the difference:

- **Interactive apps** (like Word or Chrome) need user input and have a GUI. They're what users directly interact with.
- **Non-interactive apps** (like Windows Update or background services) run silently. They're crucial for automation but don't need user attention.

Troubleshooting with Logs & Proc Mon

- **Event Viewer**: The go-to for checking installation errors (Application logs) or system crashes (System logs).
- **Process Monitor (Proc Mon)**: Filters (e.g., Reg Set Value) help track registry/file changes during installs, making it easier to debug failed setups.

Final Thoughts

This session tied together tools and techniques for system debugging, Intune management, and software packaging. The hands-on tools like Sys internals and Intune will be incredibly useful in real-world scenarios—whether it's automating logins, remotely managing PCs, or deploying apps silently. Next, I'd like to practice converting an MSI to .intunewin and testing it in Intune's trial environment.