



# **GUIDE**

# **Endpoint Security with the Windows Registry**

*v1.0.0*

Author:

**Eldon Gabriel**

August 30, 2025



Cybersecurity Professional | IT Security Consultant

## TABLE OF CONTENTS


<b>REVISION HISTORY</b>	<b>2</b>
<b>1.0 UNDERSTANDING THE WINDOWS REGISTRY</b>	<b>3</b>
1.1 Project Description	3
1.2 The Five Root Keys	3
1.3 Data Types and Storage	4
1.4 Who Uses the Registry	4
1.5 Tools, Protection, and Recovery	5
1.6 Using the Registry for Restrictions	5
1.7 Why This Matters	6
<b>2.0 CONCLUSION</b>	<b>7</b>
2.1 Key Takeaways & Practical Implications	7





Cybersecurity Professional | IT Security Consultant

## REVISION HISTORY

Version	Date	 Author	Description of Changes
v1.0.0	08/30/2025	Eldon G.	Initial draft.





## 1.0 UNDERSTANDING THE WINDOWS REGISTRY

### 1.1 Project Description

The Windows Registry is a central database used by Windows to manage system and user settings. It acts as the operating system's "brain" optimized for fast read and write operations, and is loaded into the RAM at startup.

The registry comprises two main components:

- **Registry Keys:** This works like folders that organize settings.
  - **Registry Values:** This contains the actual configuration data stored within these keys.
- 

### 1.2 The Five Root Keys

Windows organizes its registry under five primary root keys:

- **HKEY\_CLASSES\_ROOT (HKCR):** Defines file associations and tracks the registered applications.
- **HKEY\_CURRENT\_USER (HKCU):** Stores the active user profile settings.
- **HKEY\_LOCAL\_MACHINE (HKLM):** Contains system-wide settings, including
  - **SAM:** Security Accounts Manager (a common target for attackers).
  - **Security:** Local system security policies are also important.
  - **System:** Boot and kernel configurations.
  - **Software:** Installed applications and their settings.
  - **Hardware:** Hardware details captured during booting.
  - **Components / BCD.dat:** Low-level configuration data.
- **HKEY\_USERS (HKU):** Holds HKCU equivalents for all user accounts.
- **HKEY\_CURRENT\_CONFIG (HKCC):** Reflects the current hardware setup and runtime-only settings.



## 1.3 Data Types and Storage

Registry values can take several forms.

- String Values (Unicode): Text data
- Binary Values: Raw data blocks
- DWORD (32-bit) and QWORD (64-bit): Integer data
- Multi-string: Lists of text entries
- Resource descriptors: Hardware resource mappings

Registry data are stored in **hives**, such as:

- System hive → `C:\Windows\System32\Config`
- User hive → `C:\Users\<username>\NTUSER.DAT`

---

## 1.4 Who Uses the Registry

The registry is accessed by nearly all parts of Windows, including

- Kernel
- Drivers
- Services
- SAM (Security Accounts Manager)
- User interface components
- Third-party applications

### Exceptions:

- Some .NET applications use XML instead.
- Portable apps can store configurations within their own folders



## 1.5 Tools, Protection, and Recovery

- **Regedit.exe:** The built-in tool for editing and exporting registry keys.
  - **Backups:** Always back up your data before making any changes.
  - **System Restore Points:** Provides recovery for corrupted keys or failed edits. Stored under `C:\System Volume Information`, these restore points require at least 200MB of free space. They are created automatically during software installations, unsigned driver installations, or manually by users.
- 

## 1.6 Using the Registry for Restrictions

Registry restrictions prevent untrusted users from executing programs. To set up:

1. Navigate to:  
`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer`
  - If Explorer does not exist, right-click Policies → New → Key → Explorer.
2. **The `DisallowRun` Value:** This is a **DWORD (32-bit) value** created under the Explorer folder. Its function is to act as a master switch: setting its value to 1 enables the program restriction policy for the user.
  - Create a **DWORD (32-bit) Value**.
  - Name: `DisallowRun`
  - Value: 1
3. **The `DisallowRun` Key:** This is a **subkey** (like a folder) that you create *under* the `Explorer` key. Its purpose is to hold the list of programs you want to block.
  - Under the Explorer key, create a **key named `DisallowRun`**.
4. **The String Values:** Inside **The `DisallowRun` Key**, you create **string values**. Each of these string values corresponds to a specific executable file you want to prevent from running. The name of the string can be anything you choose, but the value must be the exact executable name (e.g., `cmd.exe`).



5. Inside DisallowRun, create **string values** for each program to block:

String Name	Value (Executable)
[Program Name]	<ProgramExecutable>

Replace *[Program Name]*, *<ProgramExecutable>* with the actual program and executable names when applying the guide.

5. Log in as **Administrator** to verify that all programs run.
6. Log in as an **Untrusted User** to confirm that blocked programs cannot execute.
7. Run `whoami /user` for each account. This shows the SID.

---

## 1.7 Why This Matters

- Reduces the risk of **living-off-the-land (LOTL)** attacks by limiting built-in utilities.
  - Enforces the **principle of least privilege**.
  - It works in combination with group policies, application whitelisting, and endpoint monitoring.
  - Supports layered defense strategies and can be reused in laboratories or real-world cases.
-



## 2.0 CONCLUSION

### 2.1 Key Takeaways & Practical Implications

This guide shows how registry restrictions can block the execution of programs by untrusted users. This method is reusable across laboratories and real-world deployments. When combined with group policies, application whitelisting, and endpoint monitoring, it strengthens control over endpoints and reduces opportunities for attackers.

#### **Risk:**

Untrusted users may run powerful system tools that attackers exploit to mask their malicious activities.

#### **Fixes:**

- Add the **DisallowRun** key and list executables to be blocked.
- Restrict Control Panel access for untrusted users.
- GPOs are used to apply restrictions across multiple accounts.
- Confirm that blocked programs cannot run.

#### **Procedures:**

- Optionally, only allow approved programs while blocking all others.
- Maintain a record of restricted executables and justifications for each.

#### **Security Alignment:**

- **Least Privilege:** Users only access what is required.
- **CIS Control 4:** Reinforces secure system configurations.
- **NIST CSF Protect:** Safeguards critical services and assets.

#### **Enterprise Application: Registry vs. Group Policy**

While the **DisallowRun** registry method is effective for a single system, it's not a scalable solution for an enterprise environment. In a business setting, administrators use **Group Policy Objects (GPOs)** to centrally manage and enforce security settings. GPOs are powerful because they can apply policies





Cybersecurity Professional | IT Security Consultant

to thousands of computers at once from a central location, and they automatically reinforce those settings at set intervals. This prevents users from circumventing security controls and ensures consistent security across the entire network.

