==========

==========

CATEGORY: Processes

==========

==========

This category includes the following scripts:

1. **List Running Processes**
2. **Kill Process by Name or ID**
3. **Get Process Details**
4. **Start New Process**
5. **Find Process by Port**

Below is the complete SOP library for this category.

SOP 1 – List Running Processes

Script Name: List Running Processes **Category:** Processes **Version:** 1.0 **Approved By:** IT Operations / Security

1. Purpose

This script enumerates all running processes on the system, supporting troubleshooting, performance analysis, and security investigations.

2. Scope

- Windows servers and workstations
- All user and system processes
- Used by operations, engineering, and security teams

3. Definitions

- **Process:** An executing program instance.
- **PID:** Process Identifier.

4. Preconditions

- Operator must have permission to query process information.
- System must be reachable.

5. Required Inputs

- Optional: Process name filter
- Optional: User filter

6. Procedure Steps

1. Input Collection
    - Wizard prompts for optional filters.
2. Process Enumeration
    - Retrieve all running processes.
    - Apply filters if provided.
3. Attribute Retrieval
    - Extract:
        - PID
        - Process name
        - CPU usage
        - Memory usage
        - Start time
        - User context
4. Output Formatting
    - Present structured process list.
5. Logging
    - Log filters, operator, timestamp.

7. Expected Output

- List of running processes with key attributes.

8. Post-Execution Validation

- Operator may verify using Task Manager or `Get-Process`.

9. Error Handling

- Access denied
- Invalid filter
- Process list unavailable

10. Security Considerations

- Process data may reveal sensitive activity; restrict access.

11. Audit Logging Requirements

- Operator ID
- Filters used
- Timestamp

12. Organizational Benefit Statement

This script provides a consistent, auditable method for enumerating processes, supporting troubleshooting and security monitoring.

SOP 2 – Kill Process by Name or ID

Script Name: Kill Process by Name or ID **Category:** Processes

1. Purpose

This script terminates a running process by name or PID. It supports troubleshooting, remediation, and incident response.

2. Scope

- Windows servers and workstations
- User and system processes (with appropriate permissions)

3. Definitions

- **Terminate Process:** Forcefully stop a running process.

4. Preconditions

- Operator must have administrative rights for system processes.
- Termination must be authorized.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for name or PID.
2. Process Resolution
    - Identify matching process(es).
    - If multiple matches by name, return list for selection (if supported).
3. Safety Check
    - Prevent termination of critical system processes unless explicitly authorized.
4. Termination Operation
    - Terminate process using appropriate API.
5. Post-Termination Verification
    - Confirm process no longer exists.
6. Logging
    - Log PID/name, operator, timestamp.

7. Expected Output

- Confirmation of successful termination.

8. Post-Execution Validation

- Operator may verify via Task Manager.

9. Error Handling

- Process not found
- Access denied
- Termination blocked by system

10. Security Considerations

- Terminating processes may disrupt services; ensure approvals.
- Avoid terminating security or monitoring tools.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script ensures process termination is performed safely and with full accountability, supporting troubleshooting and incident response.

SOP 3 – Get Process Details

Script Name: Get Process Details **Category:** Processes

1. Purpose

This script retrieves detailed information about a specific process, supporting troubleshooting, performance analysis, and security investigations.

2. Scope

- Windows servers and workstations
- User and system processes

3. Definitions

- **Process Details:** Includes CPU, memory, handles, threads, path, and command line.

4. Preconditions

- Operator must have permission to query process details.
- Process must exist.

5. Required Inputs

- Process name or PID

6. Procedure Steps

1. Input Collection
    - Wizard prompts for name or PID.
2. Process Resolution
    - Identify matching process.

3. Attribute Retrieval
    - Retrieve:
        - PID
        - Process name
        - Executable path
        - Command line
        - CPU usage
        - Memory usage
        - Handle count
        - Thread count
        - Start time
        - User context
4. Output Formatting
    - Present structured process details.
5. Logging
    - Log PID/name, operator, timestamp.

7. Expected Output

- Detailed process information.

8. Post-Execution Validation

- Operator may verify using Task Manager or Process Explorer.

9. Error Handling

- Process not found
- Access denied
- Process terminated during query

10. Security Considerations

- Command line arguments may contain sensitive data.
- Restrict access to authorized personnel.

11. Audit Logging Requirements

- Operator ID
- Process name/PID
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for retrieving process details, supporting troubleshooting and forensic analysis.

SOP 4 – Start New Process

Script Name: Start New Process **Category:** Processes

1. Purpose

This script launches a new process on the system, supporting automation, troubleshooting, and administrative workflows.

2. Scope

- Windows servers and workstations
- User and system processes

3. Definitions

- **Process Launch:** Starting an executable with optional arguments.

4. Preconditions

- Operator must have permission to launch processes.
- Executable path must exist.
- Action must be authorized.

5. Required Inputs

- Executable path
- Optional: Arguments
- Optional: Run as administrator

6. Procedure Steps

1. Input Collection

    - Wizard prompts for executable path and options.
2. Validation
    - Confirm file exists.
    - Validate arguments.
3. Launch Operation
    - Start process with provided parameters.
    - Capture PID.
4. Post-Launch Verification
    - Confirm process is running.
5. Logging
    - Log executable, arguments, PID, operator, timestamp.

7. Expected Output

- Confirmation of process launch and PID.

8. Post-Execution Validation

- Operator may verify via Task Manager.

9. Error Handling

- File not found
- Access denied
- Invalid arguments

10. Security Considerations

- Launching processes may introduce risk; ensure approvals.
- Avoid launching untrusted executables.

11. Audit Logging Requirements

- Operator ID
- Executable path
- Arguments
- PID
- Timestamp

12. Organizational Benefit Statement

This script ensures process launches are performed safely and with full accountability, supporting automation and troubleshooting.

SOP 5 – Find Process by Port

Script Name: Find Process by Port **Category:** Processes

1. Purpose

This script identifies which process is bound to a specific TCP or UDP port. It supports troubleshooting, security investigations, and application diagnostics.

2. Scope

- Windows servers and workstations
- TCP and UDP ports

3. Definitions

- **Port Binding:** Association between a process and a network port.

4. Preconditions

- Operator must have permission to query network and process information.
- Port must be valid.

5. Required Inputs

- Port number
- Optional: Protocol (TCP/UDP)

6. Procedure Steps

1. Input Collection
    - Wizard prompts for port and optional protocol.
2. Validation
    - Validate port is numeric and within 1–65535.
3. Port Query
    - Enumerate active connections and listeners.
    - Identify process owning the port.

4. Process Resolution
    - Retrieve process details (PID, name, path).
5. Output Formatting
    - Present structured results.
6. Logging
    - Log port, protocol, operator, timestamp.

7. Expected Output

- Process associated with the specified port.

8. Post-Execution Validation

- Operator may verify using `netstat -ano`.

9. Error Handling

- Port not in use
- Access denied
- Invalid port

10. Security Considerations

- Port usage may reveal sensitive service information.
- Restrict access to authorized personnel.

11. Audit Logging Requirements

- Operator ID
- Port
- Protocol
- Timestamp

12. Organizational Benefit Statement

This script provides a controlled, auditable method for identifying processes bound to network ports, supporting troubleshooting and security investigations.