ASR Multi-Vector Dropper - User Guide

Overview

The ASR Multi-Vector Dropper is a PowerShell-based tool designed for security professionals to test Windows Defender Attack Surface Reduction (ASR) rules. This tool helps organizations evaluate their security posture by safely simulating various attack techniques and verifying whether ASR rules are properly configured to detect and block them.

IMPORTANT: This tool is intended for legitimate security testing purposes only. Use only in environments where you have explicit permission to conduct security testing.

Features

- Multiple delivery vectors for testing different ASR rule triggers
- Interactive menu-based configuration
- Detailed HTML report generation
- ASR rules testing and evaluation
- Optional persistence mechanisms
- Sandbox evasion capabilities
- Signature string inclusion for YARA rule testing

Installation

1. Download the `asrdropper.ps1` script
2. Open PowerShell with administrative privileges
3. Navigate to the directory containing the script
4. Run the script: `.\asrdropper.ps1`

Main Menu Options

When you run the script, you'll be presented with the following menu:

```
=== ASR Multi-Vector Dropper ===
Configure your test suite:
[1] Toggle: Execute Payloads (False)
[2] Toggle: Generate HTML Report (True)
[3] Toggle: Enable Persistence (False)
[4] Toggle: Create .lnk Shortcut (False)
[5] Toggle: Cleanup After Execution (False)
[6] Toggle: Include Signature Strings (False)
[7] Toggle: Sandbox Evasion (sleep, env checks) (False)
[8] Select Stage-2 Payload Source
[9] Choose Process for HTA/COM Launch (calc.exe)
[A] Configure ASR Rules Testing
[0] Run Dropper with Selected Options
```

Configuration Options Explained

Toggle: Execute Payloads

- **Default:** False
- **Purpose:** When enabled, the script will actually execute the generated payloads
- **Use Case:** Enable to fully test ASR rule blocking capabilities
- **Note:** Keep disabled during initial setup to avoid premature execution

Toggle: Generate HTML Report

- **Default:** True
- **Purpose:** Creates a detailed HTML report of the test results
- **Use Case:** Useful for documentation and analysis
- **Output:** Report is saved to C:\Users\Public by default

Toggle: Enable Persistence

- **Default:** False
- **Purpose:** Creates persistence mechanisms to test startup/registry based ASR rules
- **Mechanisms:** Creates startup folder entry and registry run key
- **Note:** Enable only if testing persistence-related rules

Toggle: Create .lnk Shortcut

- **Default:** False

- **Purpose:** Creates a shortcut file that points to one of the payloads
- **Use Case:** Tests shortcut-based delivery vectors and related ASR rules

Toggle: Cleanup After Execution

- **Default:** False
- **Purpose:** Removes all files created by the script after execution
- **Use Case:** Enable for automated testing to avoid manual cleanup
- **Note:** Keeps the HTML report even when enabled

Toggle: Include Signature Strings

- **Default:** False
- **Purpose:** Adds known malicious strings to payloads to trigger signature-based detections
- **Included Strings:** References to tools like Mimikatz, CobaltStrike, etc.
- **Use Case:** Tests signature-based detection capabilities

Toggle: Sandbox Evasion

- **Default:** False
- **Purpose:** Adds sandbox detection techniques to payloads
- **Techniques:** Sleep commands, environment checks
- **Use Case:** Tests anti-evasion capabilities of security tools

Select Stage-2 Payload Source

- **Purpose:** Configures the actual code that will be executed by the dropper
- **Options:**
  - Use default payload (harmless echo command)
  - Paste inline payload (custom PowerShell code)
  - Load from .ps1 file (external script file)
  - Use template (preconfigured payloads for specific tests)
- **Templates:**
  - AMSI Bypass (tests AMSI evasion detection)
  - Simulated LSASS Access (tests credential theft protection)
  - Web Callback Beacon (tests command and control detection)

Choose Process for HTA/COM Launch

- **Default:** calc.exe
- **Purpose:** Specifies which process to launch when using HTA or COM-based dropper
- **Use Case:** Customize to test application-specific ASR rules

Configure ASR Rules Testing

- **Purpose:** Select which ASR rules to test and evaluate
- **Features:** Filter by category, select standard protection rules, or custom selection
- **Provides:** Detailed rule information including GUIDs and descriptions

ASR Rules Testing

The tool can test and evaluate 16 different ASR rules across various categories:

- Mail protection rules
- Office application rules
- Script execution rules
- Executable blocking rules
- Credential theft prevention
- Lateral movement blocking
- USB protection
- Driver protection
- Ransomware protection

ASR Rules Menu Options

When you select the "Configure ASR Rules Testing" option, you'll see:

```
=== ASR Rules Testing Configuration ===
Select which ASR rules to test:

Standard Protection Rules:
[1] [ ] Block credential stealing from the Windows local security authority subsystem
    GUID: 9E6C4E1F-7D60-472F-BA1A-A39EF669E4B2
    Blocks credential theft from LSASS
...

Other Protection Rules:
[4] [ ] Block all Office applications from creating child processes
    GUID: D4F940AB-401B-4EFC-AADC-AD5F3C50688A
    Prevents Office apps from creating child processes, a common malware technique
...

[A] Select All Rules
[C] Clear All Selections
[S] Select Standard Protection Rules Only
[F] Filter by Category
[T] Toggle Test ASR Rules (False)
[B] Back to Main Menu
```

Rules Selection Options

- **Select Individual Rules:** Enter the number next to a rule to toggle its selection
- **Select All Rules:** Choose option [A] to select all available ASR rules
- **Clear All Selections:** Choose option [C] to deselect all rules
- **Select Standard Rules:** Choose option [S] to select only the standard protection rules
- **Filter by Category:** Choose option [F] to view and select rules by category
- **Toggle Test ASR Rules:** Enable or disable ASR rule testing

Payload Types

The tool creates multiple payload types to test different attack vectors:

1. **Encoded PS:** Base64-encoded PowerShell script
2. **COM Shell:** Uses COM object to launch process
3. **Multi-stage:** Chunked and reconstructed payload
4. **HTA:** HTML Application file with embedded script
5. **Signature Bait:** Contains strings that match common YARA rules (only when signatures enabled)

HTML Report

When enabled, the tool generates a comprehensive HTML report with:

- Summary of test results
- Detailed ASR rule testing outcomes
- Payload execution results
- Configuration settings
- Visualizations and charts
- Detailed rule information including GUIDs and descriptions

To enable HTML report generation:

1. Toggle option [2] to "True"
2. Run the dropper with option [0]
3. Enter a custom filename or accept the default name
4. The report will be saved to C:\Users\Public by default

Usage Examples

Basic Testing

```powershell
# Launch the script
.\asrdropper.ps1

# Toggle HTML report generation on (option 2)
# Select ASR Rules to test (option A)
# Select standard protection rules only (option S)
# Run with default harmless payload (option 0)
```

Advanced Testing with Custom Payload

```powershell
# Launch the script
.\asrdropper.ps1

# Toggle Execute Payloads on (option 1)
# Toggle HTML Report on (option 2)
# Select Stage-2 Payload Source (option 8)
# Choose "Paste inline payload" option
# Enter custom PowerShell code
# Configure ASR Rules Testing (option A)
# Select specific rules to test
# Run Dropper (option 0)
```

Cleanup After Testing

```powershell
# Launch the script
.\asrdropper.ps1

# Toggle Cleanup After Execution on (option 5)
# Configure other options as needed
# Run Dropper (option 0)
```

Best Practices

1. **Start with Execute Payloads disabled** to preview what the tool will create
2. **Begin with the default harmless payload** before using custom code
3. **Always enable HTML Report** for documentation purposes
4. **Use Cleanup option** to automatically remove test files
5. **Test in isolated environments** first before testing in production
6. **Run with administrative privileges** to ensure all features work correctly
7. **Start with Standard Protection Rules** before testing specialized rules

Troubleshooting

Script Won't Run

- Ensure you're running PowerShell as Administrator
- Check execution policy: `Get-ExecutionPolicy` should be `Unrestricted` or `Bypass`

- If restricted: `Set-ExecutionPolicy Bypass -Scope Process`

ASR Rules Not Testing Properly

- Verify Windows Defender is running: `Get-Service WinDefend`
- Check current ASR rule settings: `Get-MpPreference | Select-Object -ExpandProperty AttackSurfaceReductionRules_Ids`
- Ensure Execute Payloads is toggled ON

HTML Report Not Generating

- Check write permissions to C:\Users\Public
- Ensure script is running with administrative privileges
- Try specifying a different output path

Security Considerations

This tool is designed for legitimate security testing only. To use responsibly:

1. Obtain proper authorization before testing
2. Use only in environments where you have permission to conduct security testing
3. Do not use actual malware or dangerous payloads
4. Clean up after testing is complete
5. Document all testing activities
6. Do not distribute any generated payloads

About ASR Rules

Attack Surface Reduction (ASR) rules are part of Microsoft Defender for Endpoint's advanced features designed to reduce the attack surface of applications. They target specific software behaviors that are commonly abused by attackers.

Each rule can be configured in one of three modes:

- **Not Configured:** Rule is disabled
- **Block:** Rule will block the behavior and log an event
- **Audit:** Rule will only log an event but not block the behavior
- **Warn:** Rule will warn the user but allow them to proceed

For more information on ASR rules, visit the Microsoft documentation.