Sysmon

Prepared By:
Kazim Ali Obad

Supervisor:

Anmar Mohammed

MOHAMMED .B. HASSAN

Table of Contents

What Is Endpoint Detection?

Almost every cyber attack eventually involves executing malicious code on a target machine. Think about it an attacker sends you a file in an email, you open it, and that file contains code. Where does that code run? On the employee's computer the endpoint.

what if the file gets through the network? We cannot rely only on the network layer. We must also secure the endpoint itself the Windows (or Linux) machine the employee is using.

This is the concept of Defense in Depth is like the layers of a castle. The outer wall is your network perimeter. If that falls, the next wall is the endpoint.

Antivirus vs. EDR What Is the Difference?

The most well-known endpoint protection tool is the Antivirus (AV). It works by checking a file against a database of known malware signatures before that file is written to disk or executed. Think of airport security if you carry a known prohibited item (like a listed weapon), the scanner catches it.

But what about unknown threats? What if someone swallows contraband or hides it in a way the scanner cannot see? You have to understand behavior, not just look for known items.

That is where EDR (Endpoint Detection & Response) comes in. EDR does not only check signatures — it monitors behavior. It watches what a process does after it runs. If it suddenly creates unusual files, touches the registry, spawns hidden processes, or makes suspicious network connections the EDR flags it.

Key Difference	AV = known signature matching. EDR = behavior monitoring + analysis. EDR is more powerful but also more complex to configure and operate.

What Data Does System-Level Detection Collect?

To detect threats at the endpoint, we collect and analyze activitZy logs. Every action on a Windows or Linux machine generates a log entry. These logs are called Events. Here is what we monitor:

- **Process Activity** every time a program starts (execution), a log is created. Malware must run as a process, so this is critical for detection.
- **Disk Activity** when files are created, modified, or deleted on the hard drive.
- **Registry Activity** the Windows Registry is the central database that controls Windows settings. Many attacks leave traces here, or use the registry to persist (survive reboots).
- **Account Management** adding/removing users, changing passwords, login events.
- **Log Management** if an attacker deletes logs, that deletion itself is logged. Suspicious.
- **PowerShell / CMD** Commands attackers often use PowerShell or CMD to move laterally, enumerate the network, or execute payloads. We can monitor executed commands.
- **Network Connections** from the Endpoint when a process opens a network socket.

Why This Matters	If an attacker compromises a machine and tries to cover their tracks by wiping logs, the act of stopping the logging service itself creates an alert. You cannot fully hide.

Windows Event Viewer The Native Log System

Windows stores all its logs in a built-in tool called Event Viewer (eventvwr). Logs are organized under categories like Application, Security, Setup, and System. Each event has a unique Event ID number like a plate number for a car. No two event types share the same ID.

To enable the right logging across all company machines, we use Group Policy Objects (GPO), which are pushed through Active Directory (AD). Active Directory is a Windows Server service that acts as a central authentication database for all company users. GPO lets you enforce policies across all endpoints for example, disable CMD access for regular employees, or enable specific audit logging.

Note	We do not log everything. Storage is expensive. Instead, we configure GPO to log only the events relevant to security monitoring a balance between security value and storage cost.

Native Windows Event logs have limitations:

- They cannot log DNS queries from the endpoint. (e.g., you cannot see which domains a process resolved.)
- Limited customization you cannot easily say alert me if CMD is opened AND an ipconfig command is run in the same session.
- No built-in correlation of commands across PowerShell and CMD together.

These gaps are exactly where attackers operate. Many attack techniques rely on tools already present in Windows (Living off the Land), and the default log coverage misses them.

Sysmon

Sysmon acts as an enhanced logging driver at the kernel level. It captures everything Windows Event Viewer misses DNS queries, parent-child process relationships, full command lines, file creation times, registry changes, and more all with unique Event IDs.

Best Practice	For Sysmon configuration, use the SwiftOnSecurity template. It is a community-maintained XML config file that covers most known attack patterns out of the box. If you are starting a new SOC, load this template and then customize from there.

Sysmon Event IDs — The Language of the Endpoint

Sysmon generates its own Event IDs. These are different from Windows native IDs. You must know them:

Event ID	Name	What It Means	Attack Relevance
0	Sysmon Config Change	Sysmon service stopped, deleted, or config changed	CRITICAL: Attacker is tampering with detection. Escalate immediately.
1	Process Creation	A new process started, with full command line and hashes	Most important event. Malware MUST create a process.
2	File Creation Time Changed	A file's creation timestamp was retroactively altered	Attackers do this to hide recently dropped files (timestomping).
3	Network Connection	A process made an outbound network connection	C2 beaconing detection. Look for unusual ports and IPs.
4	Sysmon State Changed	The Sysmon driver state changed (started/stopped)	Monitor for unexpected stops.
5	Process Terminated	A process ended (PID released)	Pair with Event 1 to track process lifespan.
6	Driver Loaded	A kernel driver was loaded	Rootkits and bootkits use unsigned drivers.
7	Image Loaded	A DLL was loaded into a process	DLL hijacking and injection techniques.
8	Create Remote Thread	Process A created a thread inside Process B	Classic process injection (Cobalt Strike, Meterpreter).
9	Raw Disk Access	Direct disk read bypassing the filesystem	Credential dumping (e.g., ntds.dit, LSASS memory).

10	Process Access	A process opened the memory of another process	LSASS memory dumping (Mimikatz target).
11	File Create	A new file was written to disk	Malware dropper stage. Check hash and path.
12	Registry Object Added/Deleted	Registry key created or deleted	Persistence via Run keys, uninstall entries.
13	Registry Value Set	A registry value was modified	Malware often modifies HKCU\\Run or HKLM\\Run for persistence.
14	Registry Key/Value Renamed	Registry key or value renamed	Evasion technique to confuse detection.
15	File Stream Created	Alternate Data Stream (ADS) written	Attackers hide payloads in NTFS ADS.
16	Sysmon Config Change	Configuration file loaded or updated	Verify this is an expected admin change.
17/18	Pipe Created/Connected	Named pipe created or connected to	Cobalt Strike SMB beacon uses named pipes.
22	DNS Query	The endpoint queried a DNS server	DNS-based C2, domain generation algorithms (DGA).
23	File Delete	A file was permanently deleted	Anti-forensics, ransomware file replacement.
25	Process Tampering	Process image was hollowed or replaced	Process hollowing / process doppelganging.
27	File Block Executable	An executable file download was blocked	Prevention action — attacker tried to download a payload.
255	Error	Sysmon internal error occurred	Check why — could indicate a configuration fault.

Critical Alert

Event ID 0 means someone tampered with Sysmon itself stopped it, deleted it, or changed its configuration. This is one of the most serious alerts you can receive. Escalate immediately.

Sysmon Blocking Executable File Downloads

The most common Phishing Attack Chain initial access technique follows this sequence:

In enterprise security, we place policies at the endpoint level to stop executable files from even being downloaded — not just to detect them after the fact. Sysmon Event ID 27 (FileBlockExecutable) is the prevention mechanism.

Creating the File Block Rule Step by Step

1	**Choose the correct rule type** Use a FileBlockExecutable rule . This targets Event ID 27, which fires BEFORE the file is fully written to disk.
2	**Define the target path condition** Set condition to 'contains' with value C:\Users\[vm]\Downloads this tells Sysmon to block any executable targeting that path.
3	**Name the rule group descriptively** Name it 'BlockExecutable' or 'Block executable in Downloads folder'. Naming matters when multiple rules are in use and you need to identify which rule fired.
4	**Save as .rule (not .xml)** Save the file as FileBlock.rule. Malicious loaders (like ThreadLoader) use PowerShell to download .xml files. Saving as .rule avoids false associations with suspicious .xml download behavior in the environment.
5	**Apply via PowerShell (run as Administrator)** Run: sysmon.exe -c FileBlock.rule The output should confirm 'Configuration file validated. Configuration updated.'
6	**Verify in Event Viewer** Open Event Viewer → Applications and Services Logs → Microsoft → Windows → Sysmon → Operational → Filter by Event ID 27. Each blocked download appears as 'File Block Executable (rule: FileBlockExecutable)'.

Block Executables in Downloads Folder

```
<!-- FileBlock.rule -->
<Sysmon schemaversion="4.82">
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <RuleGroup groupRelation="or">
      <FileBlockExecutable onmatch="include">
        <Rule name="Block executable in Downloads folder" groupRelation="or">
          <!-- Block any executable that tries to write to the Downloads folder -->
          <TargetFilename condition="contains">\Downloads\</TargetFilename>
        </Rule>
      </FileBlockExecutable>
    </RuleGroup>
  </EventFiltering>
</Sysmon>
```

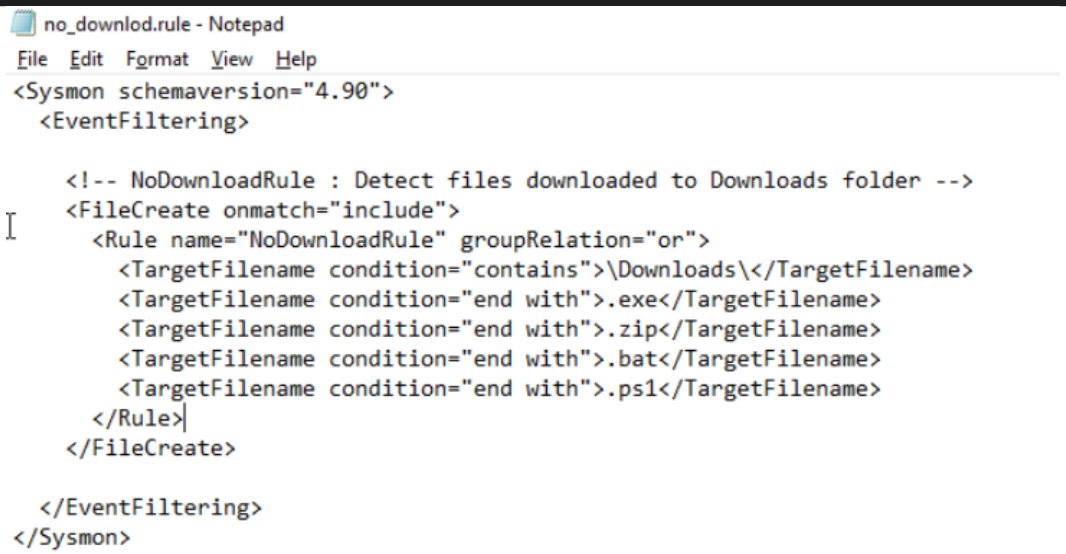

Figure 1: Sysmon rule file open in Notepad. This is a FileCreate (Event ID 11) rule named NoDownloadRule

```
PS C:\> dir


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          3/8/2026   8:13 PM                inetpub
d-----          3/6/2026   6:47 AM                netcat-win32-1.11 (1)
d-----         12/7/2019   1:14 AM                PerfLogs
d-r---        12/20/2025   5:08 AM                Program Files
d-r---         12/3/2023   6:53 PM                Program Files (x86)
d-----          3/8/2026   8:44 PM                Sysmon
d-----        12/20/2025   4:50 AM                test
d-r---         2/28/2026   8:06 AM                Users
d-----          3/8/2026   8:24 PM                Windows
-a----          3/6/2026   6:47 AM         109604 netcat-win32-1.11 (1).zip


PS C:\> cd sysmon
PS C:\sysmon> .\sysmon64.exe -c .\no_downlod.rule


System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
Configuration file validated.
Configuration updated.

PS C:\sysmon>
```

Figure 2: PowerShell session showing rule deployment.

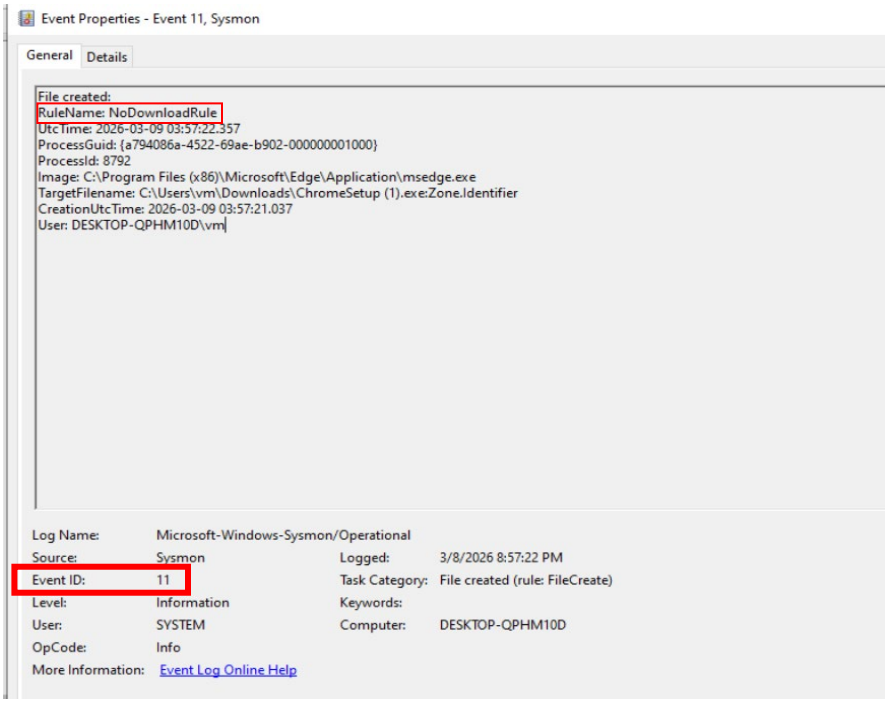

Figure 3: Windows Event Viewer showing Event ID 11 (File Created) triggered by the NoDownloadRule

blocked - Notepad

File Edit Format View Help

```
<Sysmon schemaversion="4.82">
  <HashAlgorithms>*</HashAlgorithms>

  <EventFiltering>
    <RuleGroup groupRelation="or">

      <FileBlockExecutable onmatch="include">
        <Rule groupRelation="and" name="Block executable anywhere on C drive">

          <TargetFile condition="begin with">C:\</TargetFile>

        </Rule>
      </FileBlockExecutable>

    </RuleGroup>
  </EventFiltering>

</Sysmon>
```

Figure 4: Sysmon rule file named 'blocked' using the FileBlockExecutable event type. Instead of targeting only the Downloads folder, this rule uses condition='begin with' C:\ — blocking executables from being written anywhere on the C: drive.

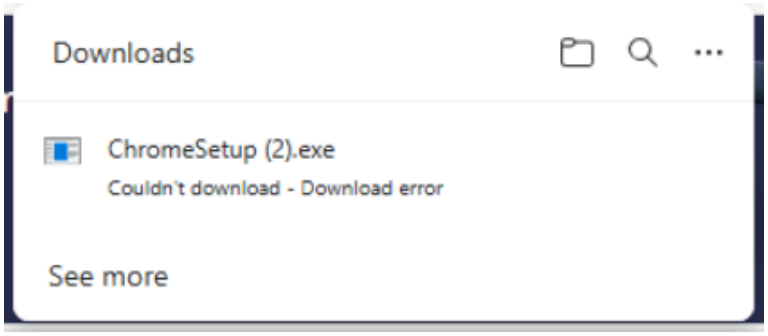

Figure 5: Microsoft Edge's download panel showing 'ChromeSetup (2) This confirms that Sysmon's FileBlockExecutable rule successfully prevented the executable from being written to disk.

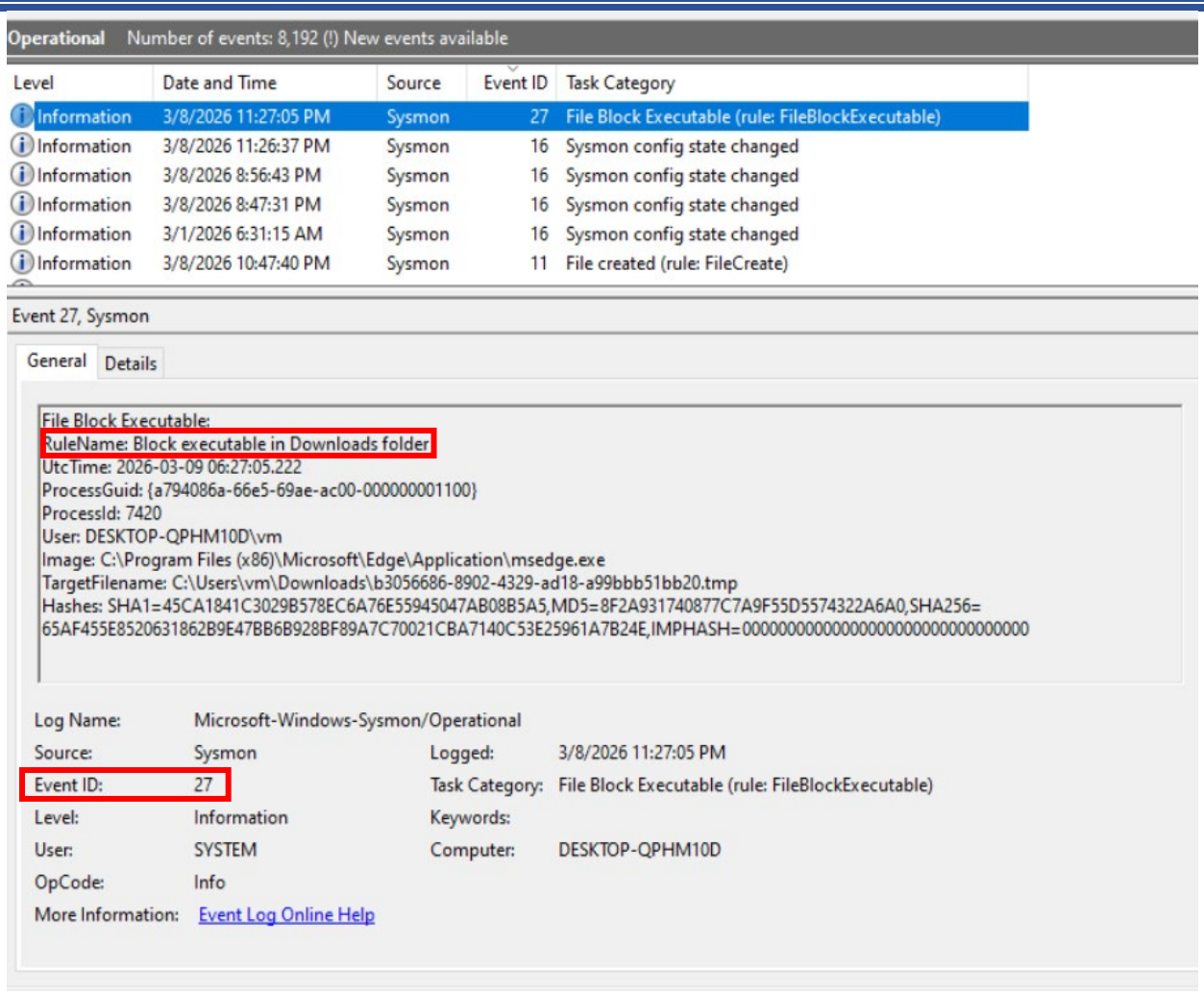

Figure 6: Windows Event showing Event ID 27 (File Block Executable

Understanding Sysmon Rule Logic OR vs AND

Operator	Where Used	Behavior	Example
OR	groupRelation on RuleGroup or Rule	The rule fires if ANY ONE listed condition matches	TargetFilename ends with .exe OR TargetFilename ends with .bat → fires on either
AND	groupRelation on Rule (structural)	ALL conditions inside the Rule must match simultaneously	ParentImage = cmd.exe AND CommandLine contains whoami → both must be true

Practical Tip	In practice, you almost always use OR at the RuleGroup level (cast a wide net, alert on any match) and AND inside individual Rules (require specific co-occurring behaviors to reduce false positives). You mostly edit: the rule name and the condition values.

Why .rule Instead of .xml?

In corporate environments, malicious loaders (like ThreadLoader) use PowerShell to pull down .xml configuration files. A script like 'Invoke-WebRequest -OutFile config.xml' is a known attacker pattern. If your Sysmon rules are saved as .xml, they may blend in with or worse, be confused for attacker infrastructure when log analysts review PowerShell download events. Saving as .rule keeps your config clearly identifiable and separate from suspicious .xml download activity.

Sysmon vs. Microsoft Defender Which Should You Use?

Factor	Sysmon	Microsoft Defender EDR
Cost	Free / Open Source	Paid (Microsoft 365 E5 or standalone license)
Setup	Manual XML configuration, deployed via GPO or script	Out-of-the-box; works immediately after license assignment
Log destination	Windows Event Log (forwards to any SIEM)	Microsoft Sentinel or Defender portal (limited SIEM export)
Customization	Fully customizable — write any rule for any behavior	Limited; predefined detection logic and some custom indicators
Prevention	Detection only (except Event ID 27 file blocking)	Advanced built-in prevention, AV, behavioral blocking
Bypass difficulty	Easier — attackers can patch Sysmon or change config	Harder — tamper protection, kernel-level enforcement
Best for	Learning, small teams, manual SOC operations, custom labs	Enterprise, large teams, mature security operations

Recommendation	Start with Sysmon. Learn it manually. Struggle with the XML. Understand every event. THEN move to Defender or a commercial EDR because when you get there, everything will make sense and you will be able to write better rules, ask better questions, and solve harder problems.

Architecture From Endpoint to SIEM

The full detection pipeline works like this:

1	**Configure GPO (Group Policy)** Enable the right Windows audit policies across all endpoints through Active Directory. This ensures Event IDs 4624, 4688, 4719, etc. are being generated on every machine in the domain.
2	**Install Sysmon on Endpoints** Deploy Sysmon64.exe with a custom XML rule set (e.g., SwiftOnSecurity template) via GPO , All endpoints should run the same base configuration.
3	**Forward Logs to SIEM** Configure Windows Event Forwarding or a log shipper agent to forward the Microsoft-Windows-Sysmon/Operational channel and the Windows Security log to your SIEM in real time.
4	**Create SIEM Correlation Rules** In the SIEM, write detection rules: 'If Event ID 1 fires AND parent process is CMD AND command line contains whoami → generate HIGH alert.' The SIEM lets you join events across time, machines, and log sources.
5	**Analyst Reviews and Escalates** The SOC analyst reviews triggered alerts in the SIEM dashboard. They investigate the alert context, determine if it is a true positive or false positive, contain the threat if needed, and document findings.

Web Application Security WAF and IIS Monitoring

The Challenge with Web Applications

A web application like a company portal is reachable by anyone on the internet not just internal users. That makes it a completely different threat surface compared to an internal endpoint. Perimeter controls alone are insufficient because legitimate HTTP/HTTPS traffic is allowed through by design.

WAF Web Application Firewall

Definition	A WAF (Web Application Firewall) protects web applications specifically. Its job is to understand the OWASP Top 10 vulnerabilities and validate/block requests that look like attacks: SQL injection, XSS, broken access control, command injection, path traversal, etc.

IIS Log Monitoring

For Windows web servers running IIS (Internet Information Services), we must refer to CIS Benchmarks as the reference configuration standard. The key steps are:

- Enable IIS logging and configure it to capture: date, time, client IP, method, URI stem, URI query, HTTP status code, bytes sent/received, user agent, and referrer.
- Set log format to W3C Extended this is the format that SIEMs and log analysis tools expect.
- Forward IIS logs to your SIEM (path: C:\inetpub\logs\LogFiles\W3SVC1\).
- CIS Benchmarks for IIS specify exactly which settings to enable treat them as mandatory for any internet-facing Windows web server.
- In the SIEM, build detection rules for: error spikes (scanning/brute force), unusual URI patterns (SQL injection attempts), large outbound responses (data exfiltration), and user agents associated with known scanners.

Core Principle	Every application generates logs. Your job as a SOC analyst is to ensure those logs go somewhere useful and get analyzed. The tool changes (IIS, Apache, Nginx, WAF) but the principle is always the same: collect, forward, correlate, alert.