Sentrilite EDR/XDR for Windows — Threat-Detection-as-Code, Observability, Runtime-Security, Live Telemetry, Misconfig Scanner with AI/LLM insights.

Installation Steps

In the Zip FIle, open Sentrilite.exe OR In a Powershell Terminal run:

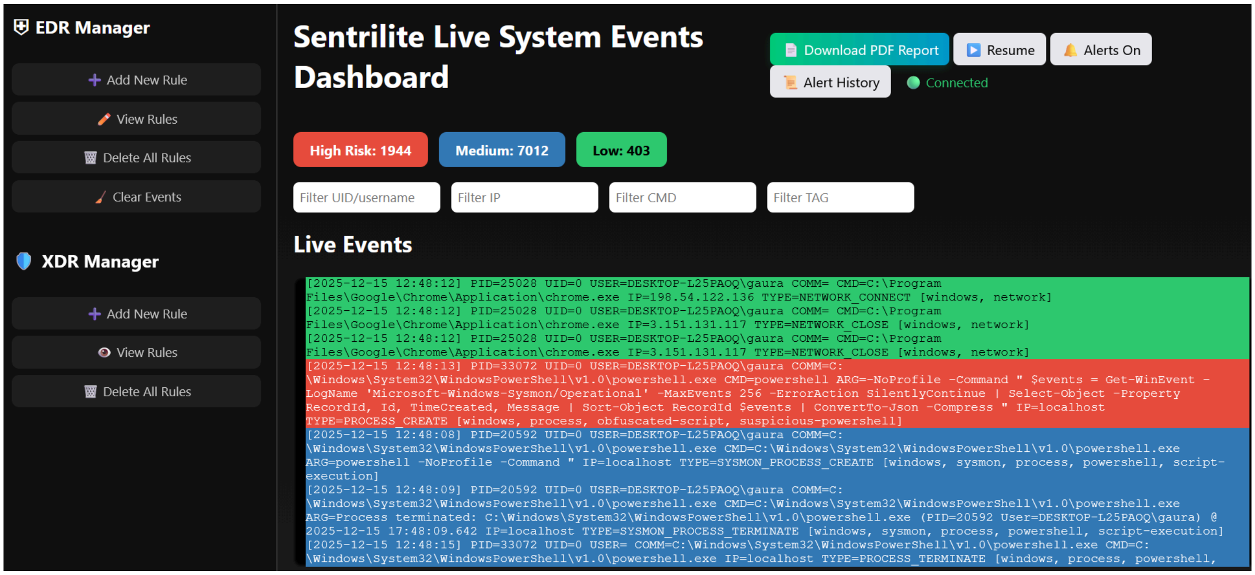
Start Sentrilite Service: `\sentrilite-service.bat start`

Check Service Status: `\sentrilite-service.bat status`

Stop Sentrilite Service: `\sentrilite-service.bat stop`

Live System Telemetry

Open the dashboard.html to check live telemetry:



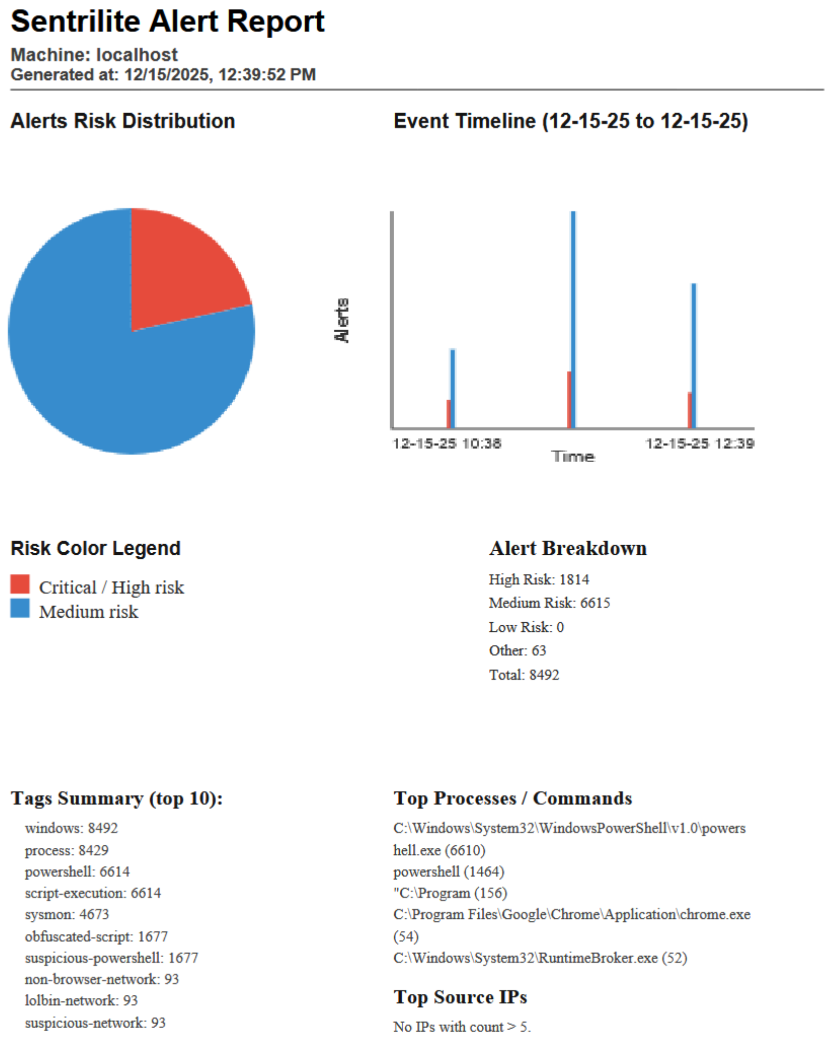
```
[2025-12-15 12:48:12] PID=25028 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM= CMD=C:\Program
Files\Google\Chrome\Application\chrome.exe IP=198.54.122.136 TYPE=NETWORK_CONNECT [windows, network]
[2025-12-15 12:48:12] PID=25028 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM= CMD=C:\Program
Files\Google\Chrome\Application\chrome.exe IP=3.151.131.117 TYPE=NETWORK_CLOSE [windows, network]
[2025-12-15 12:48:12] PID=25028 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM= CMD=C:\Program
Files\Google\Chrome\Application\chrome.exe IP=3.151.131.117 TYPE=NETWORK_CLOSE [windows, network]
[2025-12-15 12:48:13] PID=33072 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM=C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe CMD=powershell ARG=-NoProfile -Command " $events = Get-WinEvent -
LogName 'Microsoft-Windows-Sysmon/Operational' -MaxEvents 256 -ErrorAction SilentlyContinue | Select-Object -Property
RecordId, Id, TimeCreated, Message | Sort-Object RecordId $events | ConvertTo-Json -Compress " IP=localhost
TYPE=PROCESS_CREATE [windows, process, obfuscated-script, suspicious-powershell]
[2025-12-15 12:48:08] PID=20592 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM=C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe CMD=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ARG=powershell -NoProfile -Command " IP=localhost TYPE=SYSMON_PROCESS_CREATE [windows, sysmon, process, powershell, script-
execution]
[2025-12-15 12:48:09] PID=20592 UID=0 USER=DESKTOP-L25PAOQ\gaura COMM=C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe CMD=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ARG=Process terminated: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe (PID=20592 User=DESKTOP-L25PAOQ\gaura) @
2025-12-15 17:48:09.642 IP=localhost TYPE=SYSMON_PROCESS_TERMINATE [windows, sysmon, process, powershell, script-execution]
[2025-12-15 12:48:15] PID=33072 UID=0 USER= COMM=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe CMD=C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe IP=localhost TYPE=PROCESS_TERMINATE [windows, process, powershell,
```

Configuration

- license.key — place in the current directory (baked in image or mounted as Secret).
- sys.conf — network config, placed in the current directory (baked in image or mounted as ConfigMap).
- Rule files - (custom_rules.json, sensitive_files.json, windows_security_rules.json) reside in the working dir; rules can be managed via the dashboard.

Alert Report

From the Dashboard, click Download PDF Report to generate the Alert Summary report.



Sentrilite Alert Report

Machine: localhost
Generated at: 12/15/2025, 12:39:52 PM

Alerts Risk Distribution

Event Timeline (12-15-25 to 12-15-25)

Risk Color Legend

🟥 Critical / High risk
🟦 Medium risk

Alert Breakdown

High Risk: 1814
Medium Risk: 6615
Low Risk: 0
Other: 63
Total: 8492

Tags Summary (top 10):

windows: 8492
process: 8429
powershell: 6614
script-execution: 6614
sysmon: 4673
obfuscated-script: 1677
suspicious-powershell: 1677
non-browser-network: 93
lolbin-network: 93
suspicious-network: 93

Top Processes / Commands

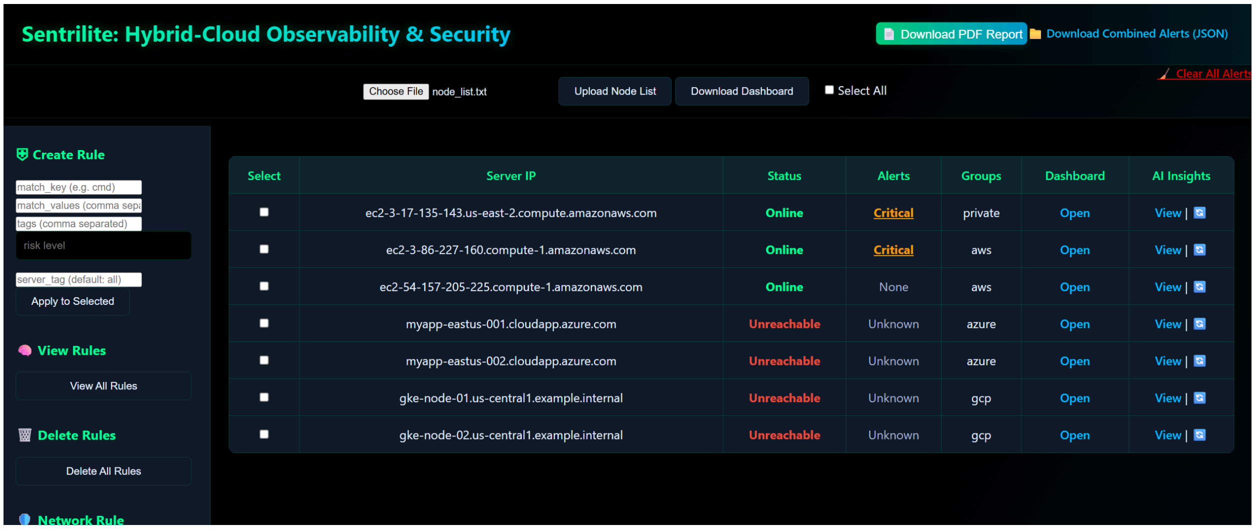
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe (6610)
powershell (1464)
"C:\Program (156)
C:\Program Files\Google\Chrome\Application\chrome.exe (54)
C:\Windows\System32\RuntimeBroker.exe (52)

Top Source IPs

No IPs with count > 5.

Main Dashboard (for all the servers)

Open the main.html, upload node_list file (csv of nodes,groups) to monitor entire infra and generate Alert Summary Reports:



Select	Server IP	Status	Alerts	Groups	Dashboard	AI Insights
□	ec2-3-17-135-143.us-east-2.compute.amazonaws.com	Online	Critical	private	Open	View |
□	ec2-3-86-227-160.compute-1.amazonaws.com	Online	Critical	aws	Open	View |
□	ec2-54-157-205-225.compute-1.amazonaws.com	Online	None	aws	Open	View |
□	myapp-eastus-001.cloudapp.azure.com	Unreachable	Unknown	azure	Open	View |
□	myapp-eastus-002.cloudapp.azure.com	Unreachable	Unknown	azure	Open	View |
□	gke-node-01.us-central1.example.internal	Unreachable	Unknown	gcp	Open	View |
□	gke-node-02.us-central1.example.internal	Unreachable	Unknown	gcp	Open	View |

Sentrilite EDR/XDR for Windows

Sentrilite EDR/XDR for Windows is a lightweight **Detection-as-Code (DAC) & Automated Response**, real-time endpoint security and observability platform. It streams structured system events to a live dashboard where JSON rules drive risk scoring, tagging, alerting, and reporting.

It provides a low-overhead endpoint security layer for Windows servers and workstations without requiring heavyweight EDR agents. If Sysmon is present, Sentrilite can automatically enrich coverage by ingesting Sysmon logs; if not, it falls back to its own native collectors.

What Sentrilite Collects on Windows

Process Activity Monitoring

Sentrilite captures all process creation and termination and normalizes them into a unified event model:

- Full executable path (cmd / comm)
- Parent PID / child PID
- User / SID context (e.g., NT AUTHORITY\SYSTEM, local users)
- Timestamps
- Tags (e.g., windows, process, powershell, lolbin-network)

You can write rules for:

- Suspicious binaries (e.g., powershell.exe, wscript.exe, certutil.exe)
- LOLBins and lateral-movement tools (psexec.exe, wmic.exe, wmiprvse.exe)
- Obfuscated or encoded script execution (e.g., -EncodedCommand, FromBase64String()
- Unexpected parent-child chains (e.g., winword.exe → powershell.exe)

File Access Monitoring (Rule-Driven)

The Windows agent detects sensitive file usage via process arguments and custom file rules, using custom_rules.json and sensitive_files.json:

- High-risk alerts for reads/writes to sensitive paths (credentials, config, keys, etc.)
- Tag events with categories such as:
- exfiltration
- credential-access
- custom tags like “test-new-tag” for your own watch files

Network Activity Monitoring

Sentrilite monitors outbound connections via Windows networking APIs (GetExtendedTcpTable), producing events that include:

- Local address / port
- Remote address / port
- Protocol (TCP)
- Owning process (image path)
- User context
- Basic connection state (LISTEN, ESTABLISHED, etc.)
- Rules can differentiate between:
- Browser baseline traffic vs. non-browser processes making external connections
- System services vs. unexpected user processes
- Access to special IPs (e.g., cloud metadata 169.254.169.254)

Optional Sysmon-Aware Enrichment

If Sysmon and the Microsoft-Windows-Sysmon/Operational log are available, Sentrilite starts a Sysmon reader loop that:

- Polls Sysmon events via Get-WinEvent
- Maps them into the same Event structure as native events
- Adds a sysmon tag plus category tags:
- process (Event ID 1)
- network (ID 3)
- driver (ID 6)

- module-load (ID 7)
- file (ID 11)
- registry (IDs 12, 13, 14)
- wmi (IDs 19, 20, 21)
- dns, network (ID 22)
- Keeps Arg1 concise and structured (short summaries rather than raw multi-line blobs)

Key point:

Sentrilite works without Sysmon, but if Sysmon is installed, you automatically get richer coverage with the same rule engine, same WebSocket pipeline, and same alert model.

Detection-as-Code (DAC)

Detection logic is fully programmable using JSON:
Rule files:

- custom_rules.json
- Windows_security_rules.json
- Sensitive_files.json
- regex_rules.json

Hot reload:

Rule files are reloaded on change — no rebuilds, no restarts.
Match on any event field, including:

- cmd, comm
- arg1 (first argument / summarized payload)
- user
- ip
- msg_type_str (e.g., PROCESS_CREATE, SYSMON_DNS_QUERY)
- tags
- aliases like file, iid mapped into shared fields
  Rules can:
- Assign risk levels: 1 = high, 2 = medium, 3 = low
- Add custom tags for later correlation / dashboards
- Trigger alerts automatically when conditions match
  (e.g., high-risk PowerShell with encoded commands, LSASS access, non-browser outbound network, WMI-based lateral movement)

This gives Windows administrators full programmability over detection logic without touching code.

🔐 Licensing

The project is currently using a trial license.key . Contact info@sentrilite.com to get a new license.

🛠️ Third-Party Integrations (PagerDuty & Alertmanager)

- PagerDuty
- Alertmanager (Prometheus ecosystem)
- SIEM forwarding (JSON events)

Alerts

When a rule marks an event as high-risk, Sentrilite:

- Creates a structured alert (JSON)
- Pushes it in real time to the dashboard
- Saves it to alerts.json
- Marks the node as “high risk” (risk-level = 1)
- Can forward to external systems (PagerDuty, AlertManager)

Alerts include:

- Process info
- User identity
- Risk reasoning via tags
- File paths or network destinations
- Human-readable summaries

Support

For licensing, troubleshooting, or feature requests:

- 📧 info@sentrilite.com
- 🌐 https://sentrilite.com