Wk2

Offline alternative: **WSUS Offline** or **WuPackage**; schedule monthly via Prefect (Week 15).

| **Layer** | **Hardening action** | **Exact command / setting** |
| --- | --- | --- |
| **1 Service integrity** | Lock service configuration so only Administrators can stop or reconfigure it. | powershell\n  sc.exe sdset SysmonDrv D:(A;;CCLCSWRPWPDTLOCRRC;;;BA)(A;;CCDCLCSWRPWPDTLOCRSDRC;;;SY)S:  \n  *(SY = SYSTEM full; BA = Built-in Admins full; everyone else = no control)* |
| **2 Executable integrity** | Set ACLs on Sysmon binaries and config so even Admins have *read-only* access during normal operation. | powershell\n  icacls \"C:\\Tools\\Sysmon\" /inheritance:r /grant:r \"SYSTEM:F\" \"Administrators:R\"  \n  # Break-write only when you intentionally update config  \n |
| **3 Config tamper alerts** | Tell Sysmon to log its own configuration changes to the Event Log. | The SwiftOnSecurity template already enables “Sysmon configuration change” (Event ID 16). Make sure you **forward ID 16** to ELK and alert on it. 4. **Configure** winlogbeat.yml Edit the configuration file winlogbeat.yml in your preferred text editor (e.g., Notepad++, VS Code):   * Configure what logs to collect (e.g., Application, Security, System) * Set output to either Elasticsearch, Logstash, or a file.   Example minimal config for Elasticsearch:  yaml  CopyEdit  winlogbeat.event\_logs:  - name: Application  - name: Security  - name: System  output.elasticsearch:  hosts: ["http://localhost:9200"] 5. **Test the Configuration** powershell  CopyEdit  .\winlogbeat.exe test config -c .\winlogbeat.yml -e  Make sure it reports Config OK. 6. **Start the Winlogbeat Service** powershell  CopyEdit  Start-Service winlogbeat  To check the status:  powershell  CopyEdit  Get-Service winlogbeat (Optional) 7. **Enable Logging for Debugging** In winlogbeat.yml, you can increase the logging level during testing:  yaml  CopyEdit  logging.level: debug  logging.to\_files: true  logging.files:  path: C:\Program Files\Winlogbeat\logs  name: winlogbeat  Would you like help customizing the winlogbeat.yml file for Logstash or Elasticsearch, or are you using a different pipeline (e.g., Kafka, file output)?   1. **Why you must forward & alert on it**  | **Risk** | **Mitigation via ID 16** | | --- | --- | | Attacker (or accident) reloads a new, noisy config to hide malicious processes. | Event 16 fires → shipped to ELK → Grafana alert “⚠ Sysmon config changed on host X”. | | Well-meaning admin edits the XML directly on a prod box, breaking feature parity with your ML model. | You see the Event 16 alert and can re-apply the repo-controlled config. |  1. **Minimal steps to wire the alert**    1. **Forwarder/Filebeat rule**: include the channel Microsoft-Windows-Sysmon/Operational and filter event\_id:16.    2. **Kibana index pattern** already shows event\_id.    3. **Grafana / Prometheus alert**:   yaml  CopyEdit  expr: rate(sysmon\_event\_total{event\_id="16"}[5m]) > 0  for: 1m  labels:  severity: critical  annotations:  summary: "Sysmon config changed on {{ $labels.host }}"   * 1. Test by running:   powershell  CopyEdit  Sysmon64.exe -c sysmonconfig-gold.xml  **In a sentence for your documentation:**  *“Sysmon Event ID 16 (configuration change) is forwarded to ELK and triggers a Grafana alert to ensure any tampering with the monitoring configuration is detected within one minute.”* |
| **4 Centralised config control** | Keep the master sysmonconfig-gold.xml in your Git repo. Push updates via Ansible/WinRM and reload with Sysmon64.exe -c <file>. Never edit in-place on production hosts. |  |
| **5 Signed updates** | Get in the habit of verifying Microsoft’s SHA-256 and digital signature before replacing Sysmon64.exe. | powershell\nGet-FileHash Sysmon64.exe -Algorithm SHA256\nGet-AuthenticodeSignature Sysmon64.exe\n |
| **6 Event-log resilience** | Increase log size (you already did) **and** lock Security/EventLog ACLs so attackers can’t clear them. | powershell\nwevtutil sl \"Microsoft-Windows-Sysmon/Operational\" /ms:102400\nwevtutil sl Security /e:true\n# ACL lock via local policy → Security Options → Audit: Force audit policy subcategory settings\n |
| **7 Service recovery traps** | Configure service-recovery options to restart on failure and trigger a custom action. | powershell\nsc.exe failure SysmonDrv reset= 86400 actions= restart/60000\n |
| **8 AppLocker / WDAC** | Create a rule that **allows** the specific Sysmon path & hash, **denies** any other copy of Sysmon in %TEMP% or user folders. Blocks LOLBin-style replacements. | Add path + hash rules in AppLocker’s *Executable Rules* or WDAC policy. |
| **9 Defender Attack-Surface Reduction** | Enable the ASR rule “Block abuse of exploited vulnerable signed drivers” | Set-MpPreference -AttackSurfaceReductionRules\_Ids 56a863a9-875e-4185-98a7-b882c64b5ce5 -AttackSurfaceReductionRules\_Actions Enabled |
| **10 Continuous monitoring** | Alert if Event ID 255 (Sysmon service stopped) or if Sysmon event flow ceases for >N minutes in Grafana. | Prometheus exporter already collects event counts → set a *dead-man* alert. |

| **Task** | **Concrete actions** |
| --- | --- |
| **4 Git repo tasks** | Pre-commit hooks → .pre-commit-config.yamlCI skeleton → .github/workflows/ci.ymlgit add + commit + push. |
| **5 SICP 1.15 – 1.29** | Create sicp/ch1\_partB.scm, push after tests pass. |

### What goes into Git this week

docs/

└── auditpol\_backup\_benign-ws01\_2025-06-02.csv # tiny

.github/

└── workflows/ci.yml

.pre-commit-config.yaml

sicp/ch1\_partB.scm

Large VM disks stay in *vms/* (git-ignored).

Wk3

### Week 3 – “Data Pipe ⚙ Warm-up” **(≈ 15 h total)**

| **Block** | **What you’ll do** | **Concrete outputs to commit / archive** |
| --- | --- | --- |
| **1 – ELK stack (6 h)** | Objective → see Sysmon events in Kibana.1. git clone the official Elastic Docker Compose template (or use the lightweight elogstash setup).2. Edit **elk.yml** → expose ports 9200, 5601 only on host-only subnet.3. Install **Filebeat** on benign-ws01; enable the sysmon module → point to ELK.4. Confirm in **Kibana Discover** that Event ID 1 rows arrive. | Repository: infrastructure/elk.yml Screenshot: docs/img/elk\_first\_event.png |
| **2 – Python env (2 h)** | Create a **Python 3.12 virtual-env** on the host (not inside VM).• python -m venv .venv && source .venv/bin/activate• pip install river pandas jupyter ipykernel.• Freeze: pip freeze > requirements.txt. | requirements.txt committed to Git |
| **3 – Benign 5-min trace (4 h)** | 1. On benign-ws01 run scripted workload: open Notepad, Edge, copy files (can just click manually).2. After 5 min stop Filebeat; copy Sysmon.evtx to host via shared folder.3. Store under data/raw/benign\_2025-06-08/. | data/raw/.../Sysmon.evtx (ignored by Git but logged in README) |
| **4 – Blog / LinkedIn note (1 h)** | Short post **“SICP §1 vs Python list-comps”** summarising exercises 1.01-1.14 and why functional transforms matter for feature engineering. | URL linked in README.md |
| **5 – Git housekeeping (1–2 h)** | • git add infrastructure/elk.yml requirements.txt README.md• Commit: “Week-3: ELK online, venv + first benign trace”• Push; CI should still pass. | Green GitHub check mark |

**Success criteria**

* Kibana Discover shows live Event ID 1 from benign-ws01.
* requirements.txt exists and installs cleanly in CI (placeholder test).
* A 5-minute Sysmon .evtx sample is archived (Git-ignored path).
* Blog post link appears in repo README and/or LinkedIn feed.

After Week 3 you’ll be ready to write the parser and feed real data into River (Week 4).