TECHNICAL WHITE PAPER

Healthcare Threat Monitor

Version 0.9 · November 2026 · All results are from simulated data

Summary

Healthcare Threat Monitor is an open source platform that detects ransomware, improper access to patient records and compromised medical devices from a hospital's security logs, and keeps a human analyst in charge of every response.

In evaluation on a simulated hospital, it detected 36 of 36 scripted attacks before analyst feedback and 36 of 36 after, across three independent runs. Analyst feedback cut false incidents in the analyst queue from 16 to 18 per day down to 7 to 8 per day. No false incident ever became eligible for isolation.

A ransomware intrusion reached the analyst queue 100 seconds after its first malicious event and reached high confidence in 6 to 7 minutes. The scripted attack deletes shadow copies at 15 minutes and starts encrypting at 16.

All results come from simulated data with scripted attacks. They show that the method works and can be measured. They are not a claim about performance in a live hospital.

The problem

A hospital cannot simply switch off a system that is under attack, because clinical care depends on it. Security tooling built for offices assumes it can.

Three features of hospitals make detection harder than in most enterprises:

- **Patient records are opened by many people for legitimate reasons.** Improper access looks like normal access, only more of it or to the wrong patients. Fixed rules cannot tell the difference.
- **Medical devices cannot run security agents.** A pump or a monitor can only be watched from the network.
- **False alarms carry a real cost.** Isolating a clinical workstation on a false alarm interrupts care, so automated response must be rare and well founded.

The platform is designed around these three constraints.

Design

Every log record follows one path: collect, detect, correlate, review, respond, record.

1. **Collect.** Signed collectors send raw logs from ten kinds of systems: EHR access records, endpoint detection, Windows, Linux, firewall, VPN, DNS, email security, medical device traffic and cloud audit. Each record is mapped into one common event format.
2. **Detect.** Rules catch known attack behaviour. Anomaly models flag activity that departs from normal.
3. **Correlate.** Alerts about the same machine or person become one incident with one confidence score.
4. **Review.** An analyst sees the evidence and decides.
5. **Respond.** On a high confidence incident the analyst may approve isolation. Incidents are forwarded to a SIEM as CEF and signed JSON.
6. **Record.** Every alert, decision and action is written to a hash chained audit log.

Four principles shaped the design:

- **A human approves every disruptive action.** Nothing is isolated automatically.
- **Detection logic is data.** Rules, thresholds and bands are rows an analyst can edit, not code.
- **Every alert explains itself.** A rule names its ATT&CK technique. A model names the features that drove its score.
- **Everything is measurable.** Ground truth for scripted attacks is stored apart from detection, so results can be scored without detection ever seeing a label.

Deployment. The platform runs in two modes from the same code. The demonstration runs the simulated hospital on one computer with one command. A hospital installation is delivered as a ready made virtual machine for VirtualBox, VMware or Hyper-V, or as a USB installer for a spare computer, so it installs with clicks: the installation's screen shows its address and a one time setup code, and the first administrator is created in the browser with that code. It runs on a single computer inside the hospital network with the simulator, test attacks and public demo login switched off, HTTPS on by default, and sign in required for every page. A setup wizard in the browser loads departments, staff and computers from spreadsheets. Access records come from a read only connection to the patient record database (PostgreSQL, MySQL or MariaDB), where the platform itself opens every session as read only, from the record system's FHIR server using SMART Backend Services sign in, or from an uploaded access report. Routers, firewalls and servers send syslog to a built in receiver, which keeps logs only from devices an administrator accepted and maps zones from the inventory. Windows computers run a small collector, installed from one downloaded file, that signs its batches with a per group secret and trusts only the installation's own certificate authority. Every path passes through the same signed ingestion path as every other source.

Learning period. For a configurable period after each kind of log first arrives, rules that report new values record them as normal instead, and anomaly models stay silent on windows that include that log. Rules for unmistakable attack steps report from the first record. When the period ends the models retrain at once.

Detection method

Detection has three layers: rules, anomaly models, and a scoring step that combines them.

Rules. Twenty one rules ship with the platform, each mapped to a MITRE ATT&CK technique. A rule is one of three kinds: a single event match, a count inside a time window, or a value an entity has never used before. Examples are a command that deletes shadow copies, thirty file renames on one machine in a minute, and a medical device contacting an unfamiliar host.

Anomaly models. Three Isolation Forest models each watch one kind of entity over fixed windows. They train on recent normal activity and need no labelled attacks.

Model	Entity	Window	What it summarises
Record access	Member of staff	15 minutes	Patients opened, how many outside their own department, comparison with their own history and their peers, unusual hours
Endpoint behaviour	Machine	1 minute	Files written, renamed and deleted, file types touched, programs new to that kind of machine
Device traffic	Medical device	5 minutes	Hosts contacted, hosts never contacted before, ports, bytes, peak packet rate

A window raises an alert only when two tests pass. It must be unusual, and at least one feature must sit far outside the normal range for its peer group. The first test has two parts because an Isolation Forest cannot see a value outside the range it was trained on, so a separate check catches values beyond anything seen in training.

Confidence and bands. Detectors are treated as independent witnesses. If one is 60% confident and another 50%, the chance that at least one is right is 80%. Several alerts from the same detector count once. The score maps to a band:

Band	Confidence	What happens
Low	Below 50%	Kept for tuning only
Medium	50% to below 80%	Analyst queue and SIEM
High	80% and above	As medium, and isolation may be approved

One safeguard sits on top: an incident cannot be high unless at least two independent log sources agree. A single faulty or spoofed sensor can therefore never lead to an isolation offer.

Human oversight and learning

An analyst decides every incident, and each dismissal makes the detectors more accurate for the entity involved.

Isolation is offered only on a high confidence incident that names the machine. The analyst approves it, a connector carries it out on the endpoint, and only the connector's acknowledgement changes the asset's status. Release works the same way.

When an analyst dismisses an incident as expected activity, each detector that contributed learns, for that entity only:

Detector	What changes
New value rule	The value joins that entity's baseline. After three entities, it becomes normal for all
Threshold rule	That entity needs one and a half times the dismissed count before the rule fires again
Match rule	The rule is silenced for that entity
Model	That entity must sit further from normal than the dismissed window, and the window joins later training as a heavily weighted normal example

A confirmed incident does the opposite: its activity is kept out of future training. Every change made by feedback is listed in the console and can be undone.

Audit and compliance

Every alert, decision and action is written to an append only log in which each entry stores the SHA 256 hash of the entry before it.

Changing or removing any past entry breaks the chain from that point on. A database trigger refuses updates and deletes in the first place, so the chain is a second line of defence. An exported log can be verified with a standalone script that needs nothing from the platform. Each message forwarded to the SIEM carries the newest audit hash, so a copy held outside the platform exposes a log that has been cut short.

The platform maps its functions to three published sets of safeguards and shows a live count as evidence for each row.

Framework	Rows mapped	Examples
HIPAA Security Rule	6	Information system activity review, 45 CFR 164.308(a)(1)(ii)(D). Audit controls, 164.312(b)
NIST Cybersecurity Framework 2.0	9	DE.AE-03, information is correlated from multiple sources. RS.MI-01, incidents are contained
HHS 405(d) practices	6	Security operations and incident response. Network connected medical device security

The platform supports these safeguards. It does not make an organisation compliant by itself.

Evaluation method

One command runs the whole evaluation against ground truth, through the same signed ingestion path that live data takes.

The simulated hospital. A simulator models one mid sized hospital: 11 departments, 197 staff on day and night shifts, 118 workstations, 10 servers, 40 medical devices and 600 synthetic patients. It writes logs in each source system's own format, about 115,000 events per simulated day. It includes the ordinary noise a real network has: mistyped passwords, account lockouts, staff who float between departments, software installs, a scheduled archive job, and devices checking for firmware.

Scripted attacks. Three scenarios write the log records an attack would leave behind. No malware is involved.

Scenario	What it does	Duration
Ransomware intrusion	Phishing attachment, script execution, credential theft, lateral movement, shadow copy deletion at 15 minutes, encryption from 16 minutes	20 minutes
Insider record snooping	A member of staff opens 45 to 75 records outside their department, including a flagged patient	40 minutes
Medical device compromise	An infusion pump probes its neighbours, calls out to a known malicious domain, then floods its gateway	13 minutes

Protocol. Each run has three parts:

1. 24 hours of normal activity, on which the models train.

2. Phase A: 24 hours of normal activity with 12 attacks mixed in, four of each kind.
3. A simulated analyst rules on every queued incident using ground truth, the detectors learn, and Phase B repeats phase A with new attacks on new targets.

Scoring. An incident counts as a detection only when it shares a machine or person with the attack, overlaps it in time, and holds at least one malicious event as evidence. Any other incident in the analyst queue counts as false. Time to detect runs from the first malicious event to the moment the incident reached the analyst queue, in event time.

Results

Every scripted attack was detected in all three runs, and analyst feedback roughly halved the false incidents an analyst had to review.

Each run used a different random seed, and so different attack targets and timing. Each run processed about 342,000 events.

Measure	Seed 11	Seed 23	Seed 37
Attacks detected, before feedback	12 of 12	12 of 12	12 of 12
Attacks detected, after feedback	12 of 12	12 of 12	12 of 12
False incidents per day, before feedback	17	16	18
False incidents per day, after feedback	8	7	7
False incidents eligible for isolation, either phase	0	0	0
Share of queued incidents that were real, before feedback	41%	43%	40%
Share of queued incidents that were real, after feedback	60%	63%	63%

Times below are medians after feedback, counted from the first malicious event.

Attack	Measure	Seed 11	Seed 23	Seed 37
Ransomware intrusion	Reached the analyst queue	100 s	100 s	100 s
Ransomware intrusion	Reached high confidence	6.3 min	7.2 min	6.2 min
Medical device compromise	Reached high confidence	6.2 min	7.1 min	6.2 min
Insider record snooping	Reached the analyst queue	13.2 min	10.9 min	8.2 min

Three observations:

- **Ransomware is caught early in the chain.** The incident reaches the queue when the document starts a script interpreter, and reaches high confidence about eight minutes before the scripted deletion of shadow copies.
- **Record snooping stays at medium by design.** Only the EHR audit log can see it, and one source cannot produce a high incident. One snooping run of 24 reached high, when other alerts joined the same incident.
- **Feedback helps most where false alarms repeat.** Device firmware checks and the file server's archive job stopped raising incidents after one or two dismissals. Account lockouts did not, because each one involves a different person.

In a live run at ten times speed, a ransomware incident reached high confidence about 20 seconds after launch. An analyst approved isolation, the simulated endpoint was isolated 4 seconds later, and the attack stopped before lateral movement or encryption.

Scope of the evaluation

The evaluation measures the method on a simulated hospital, where every attack is known, so detection and false alarms can be counted exactly and anyone can reproduce them.

- **Scripted attacks.** Each scenario follows a published attack pattern. Rules and models are kept in tables, so a hospital adds or tunes detections for the techniques it sees.
- **A simulated hospital.** In a real hospital the platform first learns that hospital's normal activity during its learning period, so false alarm rates reflect each hospital's own systems.
- **A consistent reviewer.** The simulated analyst rules the same way every time, so the effect of feedback can be measured. Every real analyst decision is recorded in the audit log and can be undone on the Rules page.
- **Seven to eight incidents a day in the queue** after feedback, a workload one person can review in a shift.
- **Isolation through a connector.** The platform requests isolation through a documented contract, which an endpoint product's webhook or API carries out.

- **Sample size.** Twelve attacks per phase across three seeds, 36 in all, each caught. Larger runs are one command away.
- **Connections.** The database, FHIR and upload paths, router and firewall formats, and the Windows collector are each tested end to end against the published standards and vendors' documented formats, and the virtual machine and USB installer for every release.

Hospitals that run the platform are invited to share measurements on their own, properly governed log data, so results from many settings can be published alongside these.

Reproducing the results

The same seed gives the same result on any machine, because the evaluation uses a fixed simulated calendar.

With the console running against an empty database, run this from the `engine` folder:

```
python -m evaluate --seed 11
```

It takes about two minutes and writes a full report with every number in this paper, broken down by attack and by detector. Seeds 23 and 37 reproduce the other two columns.

Sources

- HIPAA Security Rule, 45 CFR 164.308
- HIPAA Security Rule, 45 CFR 164.312
- NIST Cybersecurity Framework
- HHS 405(d) program
- MITRE ATT&CK
- CICIoMT2024 dataset, Canadian Institute for Cybersecurity
- Synthea synthetic patient generator

Open source under the Apache License, Version 2.0.