

INFORMATION SECURITY WITH AI

Why we should adopt AI now?



ENGINEER SHORTAGE

▲ 2.8M

Cybersecurity roles remain unfilled [1]



FALSE POSITIVES &
PRECISION

▲ 86 %

False alarm rate [2]

Analysts suffer from overloading due to high false alarm rates



CYBER ATTACK
INCREASING

▲ 5263

large and successful ransomware attacks [3]

One-third (1,424) targeting industrial enterprises—a 15% increase over 2023



INFORMATION
SECURITY POLICIES

▼ 12 %

Employees actually know their organization's security policies and rules [Kasp]

Reference [1] CG & Global Cybersecurity Forum (2024 report)

Reference [2] arXiv.org

Reference [3] Wikipedia

Usage 1 Policy & Procedure Issue

- **No one likes to read policies & procedure**

We are moving towards a prompt-based policy flow, where anyone can ask about a specific policy section and instantly receive an answer.

Setup PostgreSQL with pgvector

- PostgreSQL
- Pgvector extension
- Database configuration

Ingested local policies (YAML ➔ DB)

- Parse YAML
- Transform data
- Load into PostgreSQL

Created embeddings using OpenAI mode (1536 dims)

- Fetch text from DB
- Call OpenAI embedding API
- Store vectors in DB

Implemented hybrid search (vector + keyword)

- Define vector search logic
- Add keyword filtering
- Combine result

Usage 1 Policy & Procedure Result

- Formulate the relevant questions that must be asked, then provide clear answers based on the established scoring criteria.

```
(.venv) dashzeveg.b@Kafka-MAB:~/policy$ python3 search.py --q "Нууц үгийн бодлого" --k 3 --mode fts_only --debug
[DEBUG] DB_DSN: postgresql://unitel:unitelpass@localhost:5432/policies?sslmode=disable&gssencmode=disable
[DEBUG] MODEL/DIM: text-embedding-3-small 1536
[DEBUG] mode: fts_only
[DEBUG] phrases: ['нууц үг']
[DEBUG] q_ts (to_tsquery): (нууц & үг)
[DEBUG] conn: ('unitel', 'policies')
1) UNT-P-007-02 §3.0 – Гурав. Нэр томъёо – Нууц үг
vec=None fts=0.3000
Нууц үг нь мэдээллийн систем, сүлжээ, төхөөрөмжид нэвтрэхэд ашиглагдах, ажилтан өөрөө мэдэх нууцлалтай мөр юм.
-----
2) UNT-P-007-02 §7.1 – Долоо. Хамгаалалт ба хяналт – Нууц үгийн сахилга
vec=None fts=0.1500
Системийн түвшний давуу эрхтэй ажилтан бүр систем тус бүрт [redacted] нууц үг ашиглана; нууц үгийг дамжуулах, бичиж хадгалахыг х
ориглоно; гэнэтийн шалгалтууд хийнэ.
-----
3) UNT-P-007-02 §5.4.1 – Нууц үгийн хэрэглээ – Ерөнхий
vec=None fts=0.1333
Хэрэглэгчийн түвшний нууц үг нь “Маш нууц” ангилалд хамаарна; задруулахгүй, алдагдахаас сэргийлж хамгаална; [redacted] тутам сольж
шинэчилнэ.
-----
(.venv) dashzeveg.b@Kafka-MAB:~/policy$
```

Usage 1 Policy & Procedure Result

- Formulate the relevant questions that must be asked, then provide clear answers based on the established scoring criteria.

```
(.venv) root@Kafka-MAB:/home/dashzeveg.b/policy# ^C
(.venv) root@Kafka-MAB:/home/dashzeveg.b/policy# python3 search.py --q "хамгийн гоё пицца" --k 3 --mode fts_only --debug
[DEBUG] DB_DSN: postgresql://unitel:unitelpass@localhost:5432/policies?sslmode=disable&gssencmode=disable
[DEBUG] MODEL/DIM: text-embedding-3-small 1536
[DEBUG] mode: fts_only
[DEBUG] phrases: ['хамг гоё пицца']
[DEBUG] q_ts (to_tsquery): (хамг & гоё & пицца)
[DEBUG] conn: ('unitel', 'policies')
⚠ This is not related to Information Security Unit policy.
(.venv) root@Kafka-MAB:/home/dashzeveg.b/policy#
```

```
params = {
    "q_ts": q_ts,
    "q_vec": q_vec,
    "wv": args.wv, "wt": args.wt,
    "prefilter": args.prefilter,
    "threshold": args.threshold,
    "k": args.k,
    "sim_gate": args.sim_gate,
    "fts_gate": args.fts_gate, - full text search
}
```

Usage 1 Policy & Procedure Code Section

- Responses will be systematically normalized and fine-tuned to ensure clarity, accuracy, and consistency

```
FALLBACK_MESSAGE = "⚠ This is not related to Information Security Unit policy."

# ----- Deterministic Mongolian normalizer + curated synonyms (NO AI) -----
_SUFFIXES = ("ийн", "ын", "ний", "ны", "ууд", "үүд", "аар", "ээр", "тай", "тэй", "дээр", "руу", "рүү", "аас", "ээс", "ад", "эд", "д", "т")

def strip_suffix(t: str) -> str:
    for s in _SUFFIXES:
        if t.endswith(s) and len(t) > len(s) + 1:
            return t[:-len(s)]
    return t

def normalize_mn(text: str) -> str:
    t = (text or "").strip().lower()
    # "...-ийн/-ын/-ний/-ны бодлого" → base word only
    t = re.sub(r"\b([а-яөүёа-z0-9]+?)(ийн|ын|ний|ны)\s+бодлого\b", r"\1", t)
    # drop bare "бодлого"
    t = re.sub(r"\bбодлого\b", " ", t)
    toks = [strip_suffix(tok) for tok in t.split()]
    return " ".join(tok for tok in toks if tok)

# Curated SAFE synonyms
CURATED_SYNONYMS = {
    "цэвэр ширээ": ["цэвэр ширээ", "цэвэр дэлгэц", "clean desk", "clean desk policy"],
    # add more curated keys → list of phrases as needed
}
```

Usage 1 Policy & Procedure Fine-Tuning

- Normalization + synonym expansion (args.q) → builds the FTS query and embedding text.Embedding generation → produces q_vec

Layer	Model choice	Fine-tune	Method	Why it helps
Embedding (ANN recall)	Strong multilingual embedder	No	—	Stable, robust recall; no need to customize early
Reranker (CE)	bge-reranker / MiniLM CE	Yes	Pairwise/listwise ranking	Largest accuracy gain ; fixes “wrong top-1”
Query Rewriter	7–8B instruct LLM	Yes	LoRA SFT (+ optional DPO)	Handles Mongolian/English, synonyms, acronyms

Usage 2 Incident Response

- Stage-1 classifier: XGBoost exported to ONNX, served via FastAPI + ONNX Runtime
- Stage-2 LLM (conditional): call OpenAI with your strict JSON schema; keep prompts short, use RAG from Postgres/Feast

Data Ingestion

- "Data Ingestion" {
Splunk ► Kafka :
stream alerts
Socradar ► Kafka
: TI feed}

Processing & ML

- Preprocessor :
normalize &
enrich
- FeatureStore :
store features
- MLService :
invoke
classification
API

Orchestration

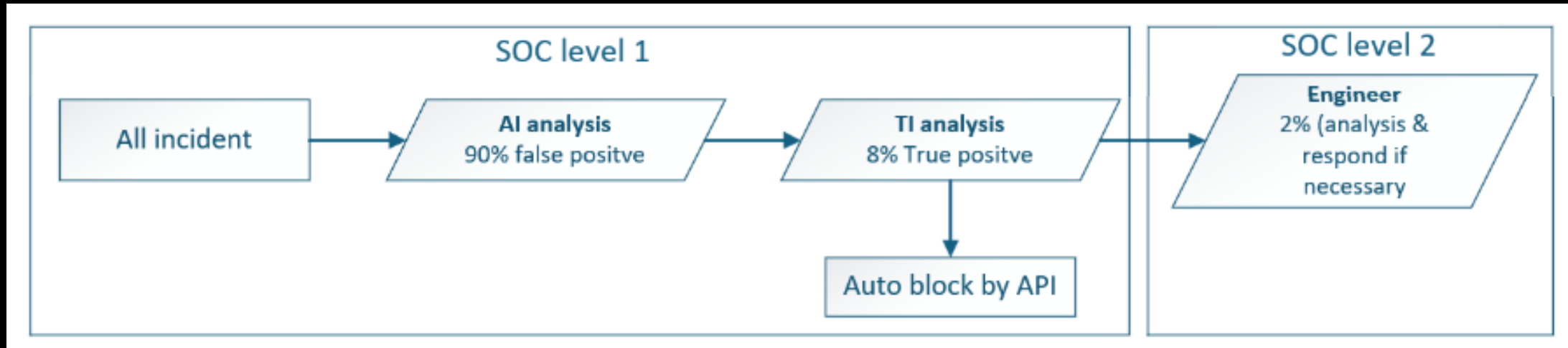
- Firewall : API
block calls
- EDR : hash/IP
block & isolate
- EmailGW :
sender/URL
block
- BotService :
escalate
indeterminate

Monitoring & Feedback

- log actions &
metrics
- label alerts
FeedbackUI
- ModelRepo :
add labeled
data

Usage 2 Incident Response Goal

To automate SOC level 1 with SOAR



Usage 2 Incident Response Result False Positive analysis

✓ False Positive

Incident

```
1750206375.92, search_name="Threat - Microsoft EDR threat alert. - Rule", orig_time="1750206375.92",
category="LateralMovement", computerDnsName="uni-lw-09985.corp.unitel.mn", evidence{}.fileName="null",
evidence{}.fileName="null", evidence{}.fileName="Termius.exe", evidence{}.filePath="null",
evidence{}.filePath="null",
evidence{}.filePath="C:\\Users\\battogtokh.b\\AppData\\Local\\Programs\\Termius",
evidence{}.processCommandLine="null", evidence{}.processCommandLine="null",
evidence{}.processCommandLine="\"Termius.exe\" --type=renderer --user-data-
dir=\"C:\\Users\\battogtokh.b\\AppData\\Roaming\\Termius\" --standard-schemes --secure-schemes=sentry-ipc --
bypasscsp-schemes=sentry-ipc --cors-schemes=sentry-ipc --fetch-schemes=sentry-ipc --service-worker-schemes --
streaming-schemes --app-user-model-id=electron.app.Termius --app-
path=\"C:\\Users\\battogtokh.b\\AppData\\Local\\Programs\\Termius\\resources\\app.asar\" --no-sandbox --no-
zygote --lang=en-US --device-scale-factor=1 --num-raster-threads=4 --enable-main-frame-before-activation --
renderer-client-id=6 --time-ticks-at-unix-epoch=-1750201794868030 --launch-time-ticks=1279180192 --mojo-
platform-channel-handle=3004 --field-trial-handle=1804,i,18044596466559428189,8396216546098597990,131072 --
disable-features=SpareRendererForSitePerProcess,WinRetrieveSuggestionsOnlyOnDemand /prefetch:1",
info_max_time="1750209600.000000000", info_min_time="1750206000.000000000",
info_search_time="1750209610.827395000", severity="Low", orig_status="New", threatFamilyName="null",
threatName="null", title="Suspicious connection to remote service"
```

AI Analysis

1. Reason:

The script is a structured log or alert message, not an executable script. It appears to originate from an EDR (Endpoint Detection and **Response**) system-specifically, Microsoft EDR. The alert reports the execution of "Termius.exe," a legitimate SSH client, from a typical install path with standard command-line arguments (no obvious signs of obfuscation, exploitation, or privilege escalation). The category "LateralMovement" and the alert title "Suspicious connection to remote service" indicate that the EDR system detected a potentially risky action (possible lateral movement using SSH), but there is no evidence in this alert of file modification, data exfiltration, persistence mechanisms, or explicit malicious activity. The rendered process command line and severity rating ("Low") suggest suspicious, but not definitively malicious, behavior.

2. Malicious: False

False positive rate was 98 %

Usage 2 Incident Response Result True Positive analysis

True positive rate was 90 %

✖ Malicious

🔔 Incident

```
1750226789.396666, search_name="Threat - Microsoft EDR threat alert. - Rule", orig_time="1750226789.396666",
category="Execution", computerDnsName="uni-lp-eb063.corp.unitel.mn", evidence{}.fileName="Viber.exe",
evidence{}.fileName="ViberPC.xEbwb", evidence{}.fileName="null", evidence{}.fileName="launch_offer.cmd",
evidence{}.filePath="C:\\Users\\gantuya.ga\\AppData\\Local\\Viber",
evidence{}.filePath="C:\\Users\\gantuya.ga\\AppData\\Local\\Temp", evidence{}.filePath="null",
evidence{}.filePath="C:\\Users\\gantuya.ga\\AppData\\Local\\Temp",
evidence{}.processCommandLine="\"Viber.exe\" AfterInstallation BurnInstaller",
evidence{}.processCommandLine="null", evidence{}.processCommandLine="null",
evidence{}.processCommandLine="null", info_max_time="1750227600.000000000",
info_min_time="1750224000.000000000", info_search_time="1750227610.675945000", severity="High",
orig_status="New", threatFamilyName="null", threatName="null", title="susp"
```

🤖 AI Analysis

1. **Reason:**

The script isn't provided directly but instead, you provided a log entry from what appears to be an EDR (Endpoint Detection and **Response**) system alerting on suspicious behavior. Here are the key suspicious indicators:

- The detection category is **Execution** and the **severity is High**.
- The execution involves multiple files, including executables with anomalous naming (e.g., "ViberPC.xEbwb", where ".xEbwb" is not a standard extension).
- Presence of files like **launch_offer.cmd** in temp directories (CMD files commonly used for scripting and potentially for executing malicious actions).
- The Viber-related executables are running from **user-writable locations** (AppData\\Local and Temp), which is unusual for trusted software installations and a common characteristic of malware or trojanized versions.
- The command line for execution (e.g., "\"Viber.exe\" AfterInstallation BurnInstaller`) may indicate post-installation scripts, which can be abused to run arbitrary code.
- The use of null entries in filenames and file paths may be obfuscation or an artifact of log collection, but it can also hide malicious activity.

Feedback

Feedback JSON (attach to every closed ticket)

```
{
  "alert_id": "A-2025-09-001234",
  "final_label": "FP | TP | BENIGN_BUT_NOISY | NEEDS_PLAYBOOK",
  "model_decision": "FP | NEEDS_ANALYST | LIKELY_TP",
  "model_confidence": 0.87,
  "correct": true,
  "error_code": "MISSED_BASELINE | TI_OUTDATED | ASSET_ROLE_MISREAD | RULE_TOO_BROAD | INSUFFICIENT_CONTEXT | OTHER",
  "missing_enrichment": ["process_tree","dns_reverse","geoip","identity_risk"],
  "allowlist_action": { "added": true, "scope": "rule+asset", "ttl_days": 30 },
  "rationale_note": "Weekly backup caused SMB burst",
  "model_version": "cls-2025-09-01_xgb; llm-2025-08-28",
  "timestamp_closed": "2025-09-02T10:12:33Z"
}
```

Feedback

Store feedback cleanly to Postgres and **LLM fine-tune** (slower loop, monthly/quarterly)

Training rows looks like

```
{  
  "input": "<SYSTEM+USER PROMPT WITH [ALERT_JSON] + [TOP-K CONTEXT]>",  
  "output": "{\"classification\":\"FP\",\"confidence\":0.84,\"key_factors\":[\"known  
baseline\"],\"required_enrichment\":[]}"  
}
```

Effects

- FP + high confidence + non-critical → propose auto-suppress with TTL.
- New allowlist entry (scope + TTL) → update store.
- TI_OUTDATED error → refresh the specific feed.