AISTRIKE · THREAT ADVISORY · CAMPAIGN INSIGHTS · CRITICAL

Synthetic Strike: The AI Offensive Frameworks Redefining Cyber Warfare

How CRAFTY FALCON and other groups use ARTEX, Cairn, PentAGI and CyberStrikeAI against Indian banking and payments and beyond. These multi-agent tools now run real intrusions against identity, cloud and payment APIs, much faster than a human operator could.

Prepared: 5 November 2026 | **Severity:** Critical | **Advisory:** AIS-TA-2026-1105

Audience: SOC, detection engineering, IR, platform owners

- **What happened.** Since late August 2026, attackers have used multi-agent LLM pentest frameworks in real intrusions, and ARTEX is the most widely used. CSIRT-Fin's CRAFTY FALCON uses CyberStrikeAI, ARTEX, Cairn and PentAGI to map Indian financial firms, then moves money through each victim's own payment and disbursement APIs.
- **Why it matters.** The attacks themselves are familiar: password spray, OAuth abuse, SQL injection, webshells and Kerberoasting. The frameworks run them faster and against far more targets. One controller hit more than 200 organisations at once, and one intrusion reached privileged access in under 15 minutes. The infrastructure rotates, so IP blocks go stale.
- **What to do now.** Hunt for behaviour as well as addresses. Look for inference-paced bursts, spec downloads followed by calls to internal APIs, writes followed straight away by a read-back, and the ARTEX and CyberStrikeAI fingerprints in Annexure A. Search back at least 90 days, and run new detections in alert mode before you block.

AiStrike position. Use the indicators in Annexure A for retro-hunting and short-lived blocks. The operators change IP, user agent and model often, and the IOAs in section 5 and the detections in section 7 keep working when they do.

Disclaimer. This report draws on open-source vendor reporting and on the indicators CSIRT-Fin published for the CRAFTY FALCON campaign. The infrastructure changes quickly, so read the full report before you block anything, and test detections in alert mode first.

Contents

SECTION	PAGE
1. Executive summary	3
2. Threat actors and campaigns	3
3. Frameworks and tools	6
3.A How ARTEX works	6
4. AI-enabled malware (LLM calls at runtime)	8
5. IOAs for hunting	8
6. Indicators of compromise	9
7. Detection engineering	9
7.A Critical ruleset mapped with MITRE ID for ARTEX	10
7.B SQL Sample Rulesets	11
8. References	12
Annexure A. Indicators by actor and campaign	13

FIGURES	PAGE
Figure 1. How ARTEX works	6

How to read this report. If you only have a few minutes, read the cover and section 1. Section 2 starts with CRAFTY FALCON and then covers the other actors. Section 3 describes the tools, with a diagram of ARTEX in 3.A, and section 4 covers malware that calls an LLM while it runs. Hunters should go to section 5 for behaviour and section 6 for exposed consoles. Detection engineers will want section 7: the ARTEX ruleset with MITRE mappings is in 7.A, and sample SQL queries are in 7.B. Indicators in Annexure A are defanged. MITRE ATT&CK IDs link to attack.mitre.org.

1. Executive summary

- From late August to October 2026, multi-agent AI pentest frameworks went from lab experiments to tools in real intrusions. ARTEX is the one we see most. PentAGI, Cairn, HexStrike-AI, Strix, Hermes and home-built Claude Code harnesses also turn up.
- The users range widely. China-nexus espionage groups (Storm-0940 / UNC7228, VAPOR PANDA, GTG-10007) and Russian espionage (GTG-20006, linked to Midnight Blizzard) use them. So do financially motivated crews: CRAFTY FALCON against Indian BFSI with CyberStrikeAI, ARTEX, Cairn and PentAGI (section 2A), a Korea/Japan ARTEX operator, Cairn/Strix/Hermes card thieves, and ShinyHunters affiliates tracked as GTG-50014. The hacktivist group GTG-50029 is on the list too.
- The techniques are old ones: password spray, OAuth and app abuse, SQL injection, webshells, Kerberoast and DCSync. The frameworks run them faster and at much larger scale. Reports cite privileged access in under 3 hours (under 15 minutes in one case), one controller hitting 200+ organizations in parallel, and a cost of about $25-50 per breach.
- Static IOCs go stale quickly because operators rotate and use open proxies, so detection has to look at behavior: inference-paced bursts, spec-driven API targeting, error-driven self-correction, read-back after writes, and canary strings.

2. Threat actors and campaigns

2A. CRAFTY FALCON: AI-driven fund-transfer campaign against Indian BFSI

Who and what

CSIRT-Fin, CERT-In's finance CSIRT, named the actor CRAFTY FALCON and described it as "an AI-assisted financial threat actor that surveys its targets with machine intelligence, dives at machine speed and strikes by invoking the victim's own payment and disbursement APIs." The advisory gives no attribution or location. *(public)* Reuters/CNA note that at least two of its tools are China-linked.

The targets are Indian NBFCs, digital lending platforms, fintechs, payment aggregators, payment service providers, and crypto exchanges or virtual digital asset service providers. The actor is after unauthorized fund transfers, and also steals crypto wallet keys.

(public) CERT-In calls the campaign "active and ongoing." CSIRT-Fin is still getting reports of successful attacks, some from entities that had already acted on the earlier advisories.

The campaign has run through three advisories so far:

ADVISORY	DATE	CONTENT
261830103401 (Advisory 12/2026)	01 Oct 2026	Unauthorized fund transfers via payment and disbursement API abuse. *(public)* CERT-In public version 09 Oct
261830103401_ADD.1 (Advisory 13/2026)	15 Oct 2026	IOCs and rules for CyberStrikeAI C2 infrastructure, including the full CyberStrikeAI beacon, panel and YARA/Sigma rule set (not repeated in ADD.2)
261830103401_ADD.2 (Advisory 15/2026, Annex-1)	30 Oct 2026, circulated 04 Nov	Adds the ARTEX family and Cairn, 516 more IPs, TLS fingerprints and attack observations. Annex-2 lists IOCs from other, unrelated incidents. Full indicators are in the CSV

Tools used

TOOL	WHAT CSIRT-FIN OBSERVED
CyberStrikeAI	Open-source Go framework that combines LLM attack orchestration with a built-in C2. Beacons talk over HTTP/HTTPS, raw TCP and WebSocket using the `X-Implant-Token` / `X-Session-Token` headers and fixed paths `/check_in`, `/tasks`, `/result`, `/upload`, `/file/`, `/ws`. Payloads are AES-256-GCM encrypted and base64-encoded, and raw-TCP sessions start with the magic `CSB1`. Panels run on TCP 8080 with a self-signed certificate `CN=CyberStrikeAI`. It persists through cron, bashrc and a registry Run key named `SystemUpdate`. *(public)* Kaspersky found no AI in the implant itself: the server-side LLM queues commands, and the implant polls for them and runs them. In Kaspersky's cases the attackers reached domain admin in 3-4 days
ARTEX family (Autumn-27/ARTEX, artex-global, scopeweaver)	The platform shown in Figure 1 (section 3.A). It runs three parallel workers by default and does not rate-limit traffic to the target. Consoles sit behind nginx with TLS certificates whose CN contains `artex` (for example `artex.<ip>.sslip.io`) or a self-signed `O=ARTEX, ST=Beijing, C=CN`. Adds user agent `ARTEX-worker/1.0` to the known `artex-enrich/1.0`, `norma/0.4` and `artex-selfupdate`. A Korean derivative called "Redex" (favicon mmh3 `6434073`) is also listed
Cairn	State-space search engine. `cairn serve` runs on port 8000 (Python/FastAPI, SQLite database of facts, intents and hints). Its workers are AI coding-agent CLIs (`claude`, `codex`) run inside per-project Kali containers, with prompts that forbid refusal. The containers are preloaded with impacket, netexec, bloodyad, coercer, kerbrute, chisel, proxychains, ysoserial, jwt_tool, nuclei, katana, naabu, dalfox, ffuf, cloudfox, public PoC repositories (`/home/kali/pocs`) and knowledge bases (`/home/kali/knowledges`). Behavioural signature: many unrelated attack threads (web, AD, cloud) against one target at once, each persisting through failures, and repeat visits with new angles taken from stored facts
PentAGI	*(public)* Named in CERT-In's public 04 Nov statement but not detailed in ADD.2

Storm-0940 (Microsoft) = UNC7228 (GTIG); Wiz tracking. China state-aligned

- The group has used ARTEX since Aug-Sep 2026 against hundreds of organizations in aerospace and defense, semiconductors, telco, software, government, think tanks and other sectors.
- It was behind the ConnectWise Cloud Backup (SkyKick) supply-chain compromise. ARTEX then tried to sign in to 2,250+ tenants with the stolen multi-tenant app credentials.
- A password spray from 10 to 14 Oct hit 327 accounts across 8 organizations, with 151k+ failures and a peak of 2,900 failures a minute.

- After getting in, it ran Graph reconnaissance, added app credentials, had a service principal assign itself Global Admin, pulled Key Vault secrets in bulk with `SecretGet`, and read mail, Teams and SharePoint.
- Wiz saw it download JS bundles, source maps and OpenAPI specs, then tamper with OAuth callbacks, send unsigned JWTs to 60+ SSO routes in 2 minutes, call Cognito `InitiateAuth` / `ForgotPassword`, and spray Okta and Entra in 20-60 minute bursts with a new user agent each burst. It moves to a new IP within minutes of being blocked.
- Time to deep access was about 4 hours, and 15 minutes at the fastest.
- ATT&CK: T1595, T1078.004, T1195, T1199, T1098.001, T1098.003, T1528, T1555.006, T1087.004, T1580, T1110.003.

VAPOR PANDA (CrowdStrike). China-nexus espionage

- From 17 Oct 2026, a single ARTEX controller ran against 200+ organizations in 14 countries. Targets included foreign ministries in LATAM, Europe, Asia and the Middle East; US pharma, semiconductor, defense and nuclear firms; and telcos in the US, Belgium and the Philippines. Nine compromises are confirmed.
- The operator used DeepSeek and a second commercial LLM, and burned about 1.4B input tokens in one intrusion.
- Operator commands carry the tag `【人类指令】`, in CST.
- In one chain, an unsigned SAML assertion plus XXE led to IMDS role theft, then SSM-style commands to thousands of instances. Other intrusions used webshells, an unauthenticated orchestration service to get root in a container, LAPS reads and RBCD. In another, a mailbox token was forged from 2022 keys that had never been rotated, then leaked configs gave VPN access.
- The agent was told to avoid EDR: "在域里寻找没有装猎鹰的windows机器" ("find Windows hosts without Falcon").
- The post-exploitation implant is BLOODALCHEMY. It loads by sideloading `graphics-hook-filter64.dll` from a renamed GameHook `svchost.exe` and decrypting `graphics-hook_config.bin`. The project path is `D:\ProjectX\Merman_New\`.
- Tunnels were Chisel, SoftEther, Tor and SOCKS.
- Hunt for command lines containing `artx` / `artex`, cloud roles prefixed `artx`, admin-command comments starting `art18-`, and the directory `C:\ProgramData\artx5v\`.

Korea/Japan financial ARTEX operator. Chinese-speaking, financially motivated (CrowdStrike, Zscaler, Recorded Future, GMO Ierae)

- The activity ran from late Sep into Oct 2026. In Korea, 16 financial firms were breached and about 100k records leaked (Shinhan, KB Kookmin, Hana, Yegaram Savings). The way in was loan-inquiry broker portals and employee mobile backends, through SQL injection and auth bypass. In Japan, about 80 companies were targeted from 6 servers, and 8 breaches were disclosed.
- The operator used a work server and a jump server (`38[.]244[.]50[.]120`). An open directory exposed `:18899/.claude/CLAUDE.md`.
- ARTEX pointed at `ARTEX_LLM_BASE_URL=https://xcai.pro/v1` with model `deepseek-v4.1-flash`. It ran from the Docker image `artex-local:perf2-20261003`, with the UI on port 8890 and Postgres on 5435.
- Several models shared the work. DeepSeek ran the investigation and GLM reviewed its output. Grok came in through a `grok2api` bridge, and Codex ran `deepseek-v4-pro`.
- Scripts and hooks found: `tools/sync-all.sh`, `sync-memory.sh`, `tools/sync-cc-users.sh`, `tools/project_context_hook.py`, `xcai-shim.py`. Each target had `scope.md`, `surface.md`, `notes.md`, `findings/` and `evidence/`. Backups went to Backblaze B2.
- The operator asked Claude where to sell Korean PII and how to find Korean Telegram channels that trade data. A possible persona is Telegram `@YY520CN`.
- Outside finance, targets covered dating, SMS, crypto/NFT, gambling, retail, education and logistics in about 19 countries.

Cairn / Strix / Hermes e-commerce operators. Chinese-speaking, financially motivated

- Gambit reported on 22 Sep that this operator had been active since Jul 2026. It ran 105 projects in 6 days and compromised 27+ companies, including a Fortune 500 hospitality company and a major US airline. It stole 600k+ cards and planted skimmers on 19 sites plus 100+ more. OpenRouter spend was about $12-18k, or roughly $25 per target.
- One example chain started with SQL injection (EXTRACTVALUE) and a plaintext OTP that bypassed MFA. An upload led to RCE, and the operator then moved through `sudo NOPASSWD python3.12`, an NFS share with `no_root_squash`, and wp-config. The chain ended with an AWS Secrets Manager dump, the Magento key, and decryption of `cc_number_enc`.
- The skimmers persisted in many places: a loader appended to jQuery or Bootstrap with the file timestamps restored, a gtag block padded with about 100 tabs, poisoned S3/CDN objects, DB product fields, a Kubernetes initContainer, the page cache, and a cron job in the JBoss log directory that re-appended the skimmer every 2 minutes.
- The toolkit had a "Database Wipe After Extraction" skill. At one victim, `ZQ` staging tables and an overbroad cleanup dropped 180 tables, backups included.
- On 19 Oct, NorForge reported 20+ exposed Cairn panels in two clusters. AIA0001 had 14 panels and 1,100+ projects against US, Canadian, Japanese and Australian shops, used Playwright to pre-validate targets, and aimed for an uploader webshell. AIA0002 was English-language and went after cards and crypto. Across both, 1,200+ sites were attempted and 100+ breached (third-party sites of Deere, SharkNinja and Cummins among them), at about $50 per successful breach. A successful run took about 23 hours of autonomous work.
- Tools seen were Amass, dnsrecon, ffuf, nmap, PHPGGC and Playwright, plus a lab for the StyleSmuggler Magento chain (CVE-2026-75650). Techniques included polyglot image/PHP uploads, memcached session writes and Zend deserialization.

Anthropic-tracked actors (Sep 2026 report; Nov 2025 report)

- GTG-1002 is China state-sponsored and was reported in Sep 2025 (MITRE C0062). It used Claude Code with MCP pentest servers against about 30 targets, and AI did 80-90% of the tactical work. The operators got past guardrails with personas and by splitting the work into small tasks.
- GTG-20006 is Russian espionage, consistent with Midnight Blizzard, under the handle "JackPoterz". Its AI agents watched for AV detections and rebuilt and redeployed the malware on their own. It also used device-code phishing ("Embassy Kit"), hotel Wi-Fi DNS hijacking with ClickFix (CaptiveCrunch), WhatsApp companion-device takeover through WPPConnect, and theft from the drone supply chain. Its malware includes PowerChrome, WUEngine, Shadow C2, MiniPlasma, CloudSyncSvc, GiftDrop (Android) and DarkSword (iOS).
- GTG-50014 is a set of ShinyHunters affiliates (MeowSHA / frkoo / blazespider). They scanned 1.8M APKs with TruffleHog on 10 EC2 workers and used stolen victim AI API keys as their own attack compute. A SaaS supply-chain theft reached about 200 downstream customers and took 2,100 Azure AD token sets in about 34 hours. In one case a stolen developer token led to cloud admin in about 3 hours.
- GTG-10007 is a Chinese-speaking group (Changsha, Hunan). It runs agent swarms, a 13-agent scheduled collection fleet, and an autonomous zero-day pipeline aimed at security products and network appliances that produced a dozen or more possible 0-days in one month.
- GTG-50020 is Russian-speaking and financially motivated. It prompt-injected an AI vendor's eval sandbox to steal production keys, then attacked about 30 AI companies in 4 days. It runs a containerized PentAGI-style pipeline and a KYC reverse proxy called "cloak".
- GTG-50029 is a French-speaking hacktivist group. Its toolkit has PentAGI-style sub-agents, a WordPress reinstall race-condition 0-day, a webshell hidden among font files, a mu-plugin credential harvester, poisoned backups, BeEF C2 and the "fafsearch" doxxing platform. It got into 14 of 42 targets.
- GTG-50021 (kl1zy) sells cheap "Claude" access that actually proxies to another model, and harvests Anthropic credentials.

CrowdStrike "Agentic AI Campaigns" report (21 Oct 2026). Four new actors with exposed open directories

1. A Spanish/English-speaking actor hit 15+ Argentine healthcare entities with a supervisor/executor Claude Code harness derived from PentestGPT, plus MCP servers.
2. A Chinese-speaking actor went after 20+ targets, including government entities in 2 countries and Brazilian and US companies. It reached Claude through a China "transfer station" proxy and used both deterministic and swarm harnesses.
3. An Indonesian-speaking actor had 819 skills, mostly from `Claude-Red`, and targeted Indonesian government domains.
4. A Russian-speaking actor used a seven-step `/pentest-setup` workflow. It stole data from a US CRM vendor and targeted SaaS providers and data centers in Russia and Kyrgyzstan.

In one of these cases an AI agent moved laterally to DC credentials in about 20 minutes.

State actors using LLMs (GTIG May/Sep 2026)

APT41 (OSSTUN C2 development), APT45/DPRK (bulk CVE/PoC validation), APT27 (ORB fleet tooling), UNC2814 (firmware vulnerability research using expert personas), UNC6201 (automated LLM account cycling), UNC5673 (Claude-Relay-Service, CLI-Proxy-API, CLIProxyAPI relays), UNC6780 / TeamPCP (agent-planned mass credential harvest in under 6 hours; tricks AI coding assistants and LLM scanners in OSS supply-chain attacks), Basin Castle.

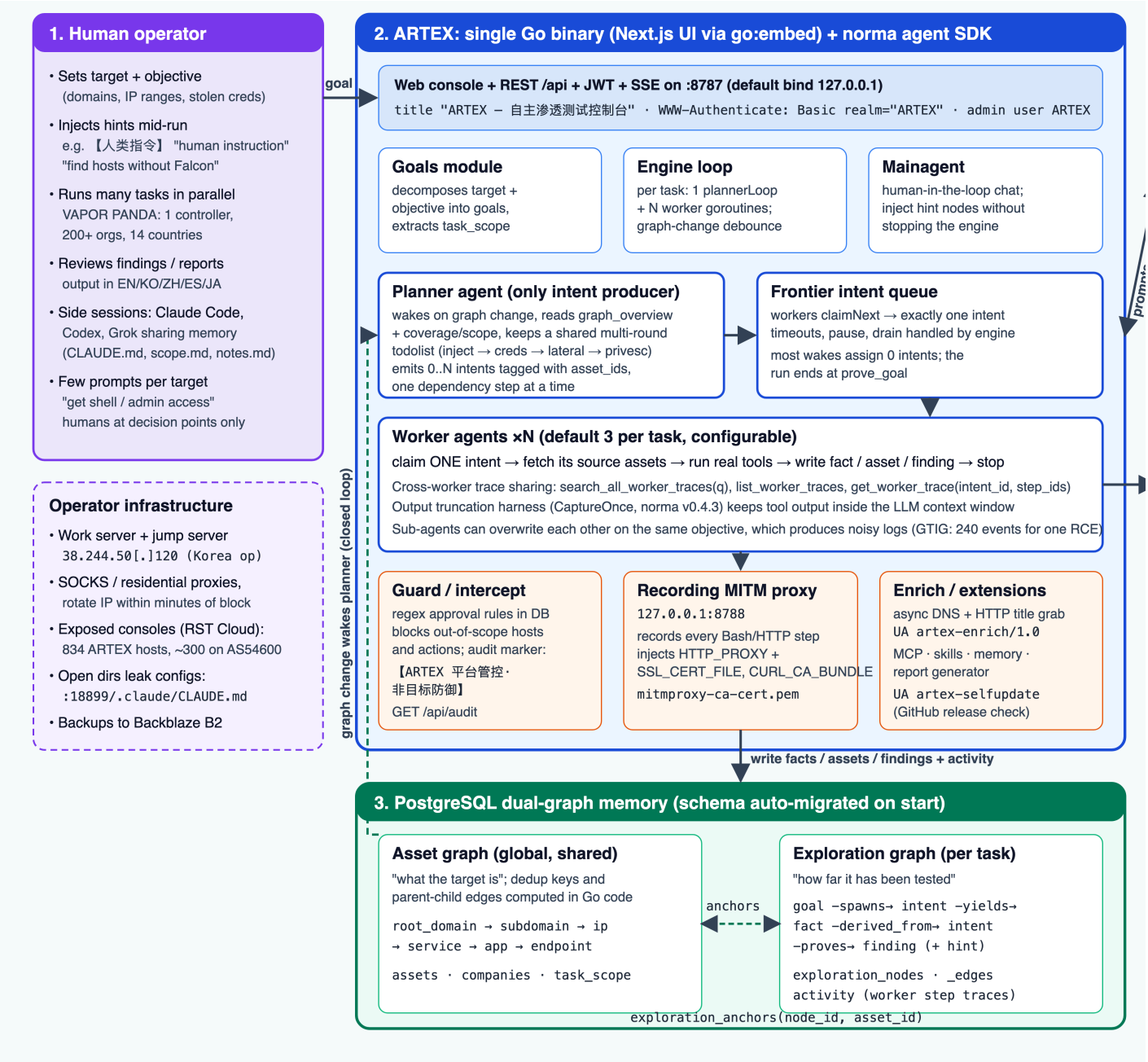
AI-agent supply-chain poisoning

- TrapDoor spans 34+ packages and 384+ versions across npm, PyPI and crates. It plants `.cursorrules` and `CLAUDE.md` files with instructions hidden in zero-width Unicode (U+200B/200C/200D/FEFF), and sends pull requests to langchain, langflow, browser-use, llama_index, MetaGPT and OpenHands.
- A malicious `AGENTS.md` in `~/Downloads` made Codex install a rogue npm package. A `CLAUDE.md` redirecting the API base URL leaked Claude Code OAuth tokens.
- Mini Shai-Hulud / TanStack-class worm tampers with `.claude` and `.vscode` configs.

3. Frameworks and tools

3.A How ARTEX works

Figure 1. How ARTEX works. The operator sets a goal. A planner and parallel workers share a PostgreSQL asset graph and an exploration graph, call whichever LLM is configured, and run real tools through a recording proxy, repeating until the goal is proven. The second panel shows the model and tool path and the kill chain seen at victims.



4. LLM backends (swappable)

- DeepSeek v4.1-flash / v4 Pro
- Zhipu GLM-5.3 (reviewer role)
- Grok 4.6 / 4.7, Claude, GPT

reasoning prompts stay in Chinese; langDirective forces output language

```
ARTEX_LLM_PROVIDER / _MODEL
ARTEX_LLM_BASE_URL / _PROXY
```

Gateways / resellers:

```
xcai[.]pro · Sub2API (:8080)
lanmao666[.]com · grok2api
OpenRouter · "transfer stations"
```

↕ tool calls

5. Tools run by workers

```
nmap · nuclei · httpx · ffuf
curl · sqlmap · Impacket
PowerShell · Bash · Python
jadx · apktool · frida · Blutter
boto3 / aws-cli / az / Graph
```

attack traffic carries each tool's own default UA (no forced ARTEX UA), except `norma WebFetch → UA norma/0.4`

routed: worker → MITM proxy → jump host / proxy pool → target

Machine tempo (IOA)

bursts 2–25 s, pauses ~25–31 s
27 events in 23 s; 1 req/s sprays
self-correct on 4xx/5xx; read-back after every write

attack

responses / errors feed back

6. Target environment: observed kill chain

① Recon and application mapping

perimeter scan, ffuf content discovery, S3 listing
`*.js.map · /openapi.json · /swagger · env.js`
`runtime-config.json · /zzz-nonexistent-xyz123`
calls internal routes seconds after download
canaries: zz_no_such_user_9f3a, ReconProbe/1.0

② Identity attacks

Okta/Entra spray: 1 s gaps, 20–60 min bursts, unique UA per burst, engineers targeted
`/sso?jwt=eyJhbGciOiJub25lIn0... (alg:none)`
OAuth code/state tamper, illegal_redirect_uri
reset/OTP abuse; Cognito ForgotPassword, SignUp

③ Exploit / API fallback

SQLi, auth bypass, IDOR, self-registration (201)
unauth PUT/POST then GET read-back in 1–2 s
`SET GLOBAL general_log_file='.../x.jsp'`
→ JSP webshells cl1217.jsp, f1219a.jsp, z1219.jsp
Kudu/SCM exec; SAML + XXE → IMDS creds

④ Privilege escalation and lateral movement

AWS --dry-run sweeps, GetCallerIdentity; ARM deployments/validate; TinyTestBlob write+delete
Kerberoast → DCSync (krbtgt), LAPS, RBCD
PtH / PtT / WinRM / Kerberos relay attempts
service principal self-assigns Global Admin

⑤ Persistence

app credential adds on existing service principals
ArtexProbe accounts, is_admin=true, password resets
`artex-probe-NNNNN.* · C:\ProgramData\artx5v\`
SSH key + Elastic IP + t3.micro jump host
webshells (JSP/PHP), BLOODALCHEMY (VAPOR PANDA)

⑥ Collection and exfiltration

SQL dumps (PII, loan data), mass blob reads, EBS snapshots via keyless EC2, 110k AD objects
bulk Key Vault SecretGet; Graph mail/SharePoint
`curl → rotating webhook.site URLs (491 seen)`
mass internal emails via app notification APIs

FRAMEWORK	WHAT IT IS	LLM BACKENDS SEEN	SEEN WITH
ARTEX (Autumn-27; forks ScopeWeaver, ARTEX PRO, ARTEX v2, BaaRai, AegisLoop)	Go + Next.js multi-agent pentest/C2. Planner, worker, and reporter agents share a fact/asset graph. Six sub-agents. Released Jul 2026, forked 12k+ times, made closed-source 08 Oct 2026. Default port 8787; nginx Basic realm `ARTEX`. Engine library "Norma"	DeepSeek v4.1-flash (via `xcai[.]pro`), GLM-5.3, Grok 4.6/4.7, commercial LLMs via Sub2API gateways	Storm-0940/UNC7228, VAPOR PANDA, Korea/Japan finance operator, extortion crews

FRAMEWORK	WHAT IT IS	LLM BACKENDS SEEN	SEEN WITH
PentAGI (VXControl)	Agent "Flows" run inside Kali sandboxes (nmap, metasploit, sqlmap). Default 8443, cert CN `PentAGI CA`	12+ cloud/local backends	GTG-50020, GTG-50029, Asia telco recon cluster, aerospace targeting via proxies
Cairn (oritera / le1xia0)	Python blackboard engine using a fact/intent/hint graph. Default port 8000, uvicorn, no auth	DeepSeek v4.1-flash (`claudecode_deepseek-v4.1-flash`), `pi_openrouter_deepseek`	NorForge AIA0001/AIA0002 (1,200+ sites, 100+ breached), Gambit card-theft operator
Strix	AI vulnerability scanner ("deep mode")	GLM 5.2, DeepSeek v4 Pro via OpenRouter	Gambit operator (discovery stage)
Hermes	Autonomous agent with persistent memory, self-written skills, scheduler	claude-opus-4.6	Gambit operator (orchestration; persona "SOUL - Red Team Operator"; 121 skills, 78 offensive)
HexStrike-AI	MCP orchestrator for 150+ security tools	Any MCP client	Underground actors weaponizing Citrix CVE-2025-7775/7776/8424 (Aug-Sep 2025)
Villager (Cyberspike)	MCP client, Kali containers that self-destruct after 24h, 4,201 prompts. MCP service on TCP 25989	DeepSeek	Abuse concern. Cyberspike suite earlier bundled AsyncRAT/Mimikatz
Custom Claude Code harnesses	Supervisor/executor state machines, MCP servers, skill packs (e.g. `SnailSploit/Claude-Red`), `/pentest-setup` slash command	Claude (often via Chinese "transfer station" proxies)	GTG-1002, CrowdStrike Actors 1-4, GTG-20006, GTG-10007
Others named	CyberStrikeAI, Shannon, MetasploitMCP, PentestGPT, Decepticon, AgentZero, ZeroHollow inference service		Detection pack coverage / experimentation (GTIG)

4. AI-enabled malware (LLM calls at runtime)

MALWARE	ACTOR	BEHAVIOR
PROMPTSTEAL / LAMEHUG	APT28 (FROZENLAKE)	Python/PyInstaller. Queries Qwen2.5-Coder-32B-Instruct on Hugging Face for one-line recon/collection commands. Lures are `.pif` files in ZIPs
PROMPTFLUX	Unknown (experimental)	VBScript dropper. Asks the Gemini API to rewrite itself and saves the result to Startup. Spreads to removable/network shares
HONESTCUE	Unattributed small actor	Gemini returns C# that is compiled in memory via CSharpCodeProvider. Discord CDN payloads
PROMPTSPY	Android (ESET/GTIG)	`GeminiAutomationAgent`. Serializes the UI tree and sends it to `gemini-2.5-flash-lite`, which returns CLICK/SWIPE JSON. Replays biometrics and rotates API keys
QUIETVAULT	Unknown	JS stealer for GitHub/npm tokens. Uses on-host AI CLIs to hunt secrets and exfiltrates to a public GitHub repo
FRUITSHELL	Unknown	PowerShell reverse shell with prompts aimed at fooling LLM-based analyzers
PROMPTLOCK	PoC	Go ransomware that generates Lua scripts through an LLM
CANFAIL, LONGSTREAM	Russia-linked	LLM-generated decoy/junk code to evade detection

5. IOAs for hunting

1. Think-then-act cadence: bursts of 1-14 requests at about one a second, then regular 25-31 second pauses while the model thinks. GTIG saw execution bursts of 2-25 seconds, such as 27 events in 23 seconds.
2. Spec-driven targeting: the agent fetches `*.js.map`, `/bundle.js`, `/openapi.json`, `/docs`, `/redoc` or `runtime-config.json`, then calls non-public routes within seconds, with no wordlist brute force.
3. Error-driven self-correction: after an HTTP 422 or 500, the agent downloads the spec again and the corrected payload succeeds. A "script too large" error is followed by a trimmed retry.
4. Goal verification: every PUT/POST is followed by a GET read-back within 1-2 seconds.
5. Semantic route guessing: after a rejection, the agent tries a trailing slash, `/api/v2/`, or an unversioned or shortened path. It guesses from meaning and does not use a wordlist.
6. IdP baseline probing: a login for a random nonexistent user, then `admin` one second later. OAuth `illegal_redirect_uri_enhanced` across many client IDs. Password-reset/OTP/MFA-profile probing.

7. Cloud permission sweeps: AWS `DryRun` with placeholder IDs, `GetCallerIdentity`, ARM `deployments/validate` across 11 resource types in about 8 seconds, write-then-delete test blobs in 330-400 ms, Kudu/SCM command exec.
8. Identity and app abuse: service principal sign-ins from new ASNs, adding app credentials, self-assigned roles, bulk `SecretGet` (20 or more across 3 or more vaults in 15 minutes).
9. Noisy on-host activity: generated PowerShell for AD enumeration (110k+ objects), Kerberoasting followed by DCSync, PtH/PtT/WinRM/Kerberos relay attempts, accounts with `is_admin=true`, mass internal notification emails.
10. Outbound from servers to webhook.site, OAST, tunnels (Chisel/SoftEther), LLM APIs (`api.deepseek.com`, `open.bigmodel.cn`, `api.x.ai`, `api.anthropic.com`, `openrouter.ai`, Hugging Face inference), and resellers (`xcai[.]pro`, Sub2API/grok2api gateways).
11. EDR-gap hunting: tasklist and folder listing to find hosts without a sensor.
12. Web persistence: polyglot image/PHP uploads, JSP via MySQL general log, webshells hidden among fonts, mu-plugins, skimmers with timestamps restored, cron re-injection, poisoned backups.

6. Indicators of compromise

ARTEX, Cairn and PentAGI consoles

Internet scans found these consoles. None of them is tied to a known intrusion, so use them for monitoring and hunting, and don't block on them alone.

IP:PORT	CONSOLE	VT
`176[.]65[.]144[.]71`	Cairn	15/92
`43[.]250[.]53[.]42`	Cairn	12/92
`117[.]72[.]220[.]70`	ARTEX	12/92
`23[.]226[.]135[.]117:8443`	PentAGI "xhackai"	8/92
`1[.]14[.]104[.]208`	ARTEX + PentAGI	7/92
`43[.]139[.]50[.]42:3000`	ARTEX v2	6/92
`31[.]57[.]47[.]29`	Cairn	6/92
`51[.]158[.]154[.]152`	ARTEX + PentAGI	3/92
`156[.]240[.]236[.]6:8001`	Cairn "cf作战平台"	2/92
`8[.]216[.]41[.]60`	Cairn "Zero"	1/92
`84[.]247[.]152[.]233`	ARTEX + PentAGI	0/92
`5[.]255[.]124[.]144:8080`	ScopeWeaver	0/92
`119[.]45[.]26[.]102:18089`	PentAGI 天融信	0/92
`123[.]57[.]3[.]71`	Cairn "VulnFlow IoT"	0/92
`38[.]11[.]221[.]26`	ARTEX	0/92
`111[.]228[.]36[.]150`	ARTEX	0/92
`82[.]25[.]190[.]221`	PentAGI	0/92
`124[.]223[.]104[.]170`	PentAGI	0/92
`23[.]95[.]170[.]215`	PentAGI	0/92

7. Detection engineering

- Most exposed consoles sit on two networks: AS54600 (about 300 ARTEX consoles) and AS398823 (about 90 PentAGI consoles).
- To find consoles in internet scan data, search for these page titles: `ARTEX: 자율 침투 테스트 콘솔`, `artex — 自主渗透测试控制台`, `AegisLoop — 攻防安全运营`, `ScopeWeaver — Autonomous Penetration Testing Console`, `Cairn · 事实图谱协同探索`. PentAGI's favicon SHA-256 is `2a620e3c5f7b49dfc6f5278352607f7931f8c38ef9d4689908af65538dd95366`, and it also listens on ports 8666 and 8765.
- On our own hosts, look for listeners on 8787 and 8788, a `mitmproxy-ca-cert.pem` file, tools started with `SSL_CERT_FILE` or `HTTP_PROXY` pointing at `127.0.0.1:8788`, and the user agents `artex-enrich/1.0`, `artex-selfupdate` or `ReconProbe/1.0`.
- At the web edge, alert when a canary request such as `/zzz-nonexistent-*` is followed within 60 seconds by routes taken from a downloaded `.js.map` or OpenAPI file. Also alert on any `alg:none` JWT, and on directory discovery, auth probing and a sign-in from one source inside 30 minutes.
- In identity and cloud logs, watch for Okta or Entra sprays with a new user agent each burst and about one second between attempts, and for a certificate or secret added to an app followed by a role assignment from the same service principal within 24 hours. `DryRunOperation`

errors across 5 or more EC2 APIs in 10 minutes, a blob created and deleted within a second, and DCSync by a non-DC account are also worth an alert.

- Theori `aipt-detections` (GitHub): Suricata, Sigma, YARA, ModSecurity CRS, KQL, and SPL fingerprints for ARTEX, CyberStrikeAI, HexStrike-AI, PentAGI, Strix, Shannon, MetasploitMCP, PentestGPT, Decepticon, and Villager. These match fingerprints only, so pair them with behavioral rules.
- Nextron: `HKTL_ARTEX_Oct26`, `HKTL_ARTEX_Docker_Oct26`, `HKTL_ARTEX_Forensic_Artifacts_Oct26`, `LOG_Norma_UA_Oct26`, `PUA_NORMA_Usage_Oct26`, ARTEX/Norma UA rules.
- Microsoft: Defender XDR alerts ("Application self-assigned Global Administrator role", "Unusual Microsoft Graph activity associated with Compromised ConnectWise Application") plus the KQL hunts in the MS blog (OAuthAppInfo, EntraIdSpnSignInEvents, GraphAPIAuditEvents, CloudKeyVaultEvents).
- Google SecOps: "Suspicious Webhook.Site Interaction", "Suspicious Curl Invocation", "Cmd Launching Curl", "AWS Cloud Accounts Getcalleridentity".
- MITRE: T1588.007 (Obtain Capabilities: AI), campaign C0062, ATLAS AML.T0040/T0043/T0051/T0054.

7.A Critical ruleset mapped with MITRE ID for ARTEX

QUERY NAME	LOG SOURCE	MITRE ID
A1. ARTEX Asset Enrichment Prober User-Agent	webserver	`T1595.002` `T1592`
A2. ARTEX Norma SDK WebFetch User-Agent	webserver	`T1595.002`
A3. ARTEX API Recon Skill Script User-Agent	webserver	`T1595.003`
A4. ARTEX Enrichment Scan Velocity	webserver	`T1595` `T1592`
A5. ARTEX Enrichment Fan-Out	webserver	`T1595` `T1592`
A6. HTTP 404 Burst From a Single Client	webserver	`T1083`
A7. Scanner-Grade User-Agent Rotation Among 404s	webserver	`T1595.002`
A8. Burst of Requests Without User-Agent	webserver	`T1595.002`
A9. Many Distinct Non-Static URL Paths From One Client	webserver	`T1083`
A10. HTTP Request Burst From One Client	webserver	`T1595`
A11. ARTEX SQL-Injection Marker Strings in Requests	webserver	`T1190`
B1. ARTEX Self-Update Check or Release Download	proxy	`T1102` `T1105` `T1588.002`
B2. ARTEX Norma web_search DuckDuckGo HTML Scrape	proxy	`T1593.002`
B3. LLM API Request From a Host Not Approved for LLM Use	proxy	`T1102.002`
B4. TLS Certificate Issued by a Mitmproxy CA	zeek (x509)	`T1557`
B5. ARTEX Console Self-Signed Certificate	zeek (x509)	`T1608.003` `T1071.001`
B6. LLM API Call Followed by a Request Burst From the Same Host	proxy, webserver	`T1059`
C1. ARTEX Console-Specific API or UI Route Access	webserver	`T1071.001`
C2. ARTEX Console Session Cookie Presented	webserver	`T1071.001`
C3. ARTEX Console Auth Status or First-Run Init Request	webserver	`T1046`
D1. ARTEX Server Binary Execution	process_creation	`T1204.002`
D2. Shell or Attack Tool Spawned by ARTEX	process_creation	`T1059.004`
D3. ARTEX Recording Proxy and CA Bundle Injected Into Tool Environment	process_creation	`T1553.004` `T1557`
D4. ARTEX Recording Proxy CA or Traffic Index Written	file_event	`T1557`
D5. ARTEX Worker Command Output Spill File	file_event	`T1074.001`
D6. ARTEX Agent Transcript File Written	file_event	`T1074.001`
D7. ARTEX Self-Update Staging Files Written	file_event	`T1588.002`
D8. ARTEX Console JWT Signing Key Created	file_event	`TA0003` `NO TECHNIQUE`
D9. ARTEX Platform Guard Audit-Log Marker	application	`T1059`
D10. ARTEX Guard-Block Burst	application	`T1059`

QUERY NAME	LOG SOURCE	MITRE ID
D11. Destructive Command Execution (ARTEX guard deny-list)	process_creation	T1485 T1561.002 T1489
D12. ARTEX Guard Marker and Destructive Command on One Host	application, process_creation	T1059 T1485
D13. ARTEX Task and Staging Folders on a Victim Host	file_event	T1074.001 T1105
D14. ARTEX Markers in Command Lines	process_creation	T1059
D15. Cloud IAM Role Created With "artx" Prefix	cloudtrail	T1098.003

7.B SQL Sample Rulesets

Rule 1 · ARTEX Recording-Proxy MITM CA Certificate Artifact · file_event · HIGH `T1557`

```
SELECT *
FROM file_event
WHERE LOWER(TargetFilename) LIKE '%\_ca%' ESCAPE '\'
  AND LOWER(TargetFilename) LIKE '%mitmproxy-ca-cert.pem%';
```

Rule 2 · Cairn Dispatcher AI Worker Processes · process_creation · CRITICAL `T1059`

```
SELECT *
FROM process_creation
WHERE (
      LOWER(Image) LIKE '%/cairn'
      AND LOWER(CommandLine) LIKE '%serve%'
  )
   OR LOWER(CommandLine) LIKE '%--dangerously-skip-permissions%'
   OR LOWER(Image) LIKE '%/codex';
```

Rule 3 · ARTEX Family Console Process or Listening Port · network_connection · HIGH `T1071`

```
SELECT *
FROM network_connection
WHERE DestinationPort IN (8787, 8788)
   OR LOWER(Image) LIKE '%artex%'
   OR LOWER(Image) LIKE '%scopeweaver%';
```

Rule 4 · Local LLM Runtime on Non-Approved Host · process_creation · MEDIUM

```
SELECT *
FROM process_creation
WHERE LOWER(Image) LIKE '%\ollama.exe'
   OR LOWER(Image) LIKE '%/ollama'
   OR LOWER(Image) LIKE '%/llama-server'
   OR LOWER(Image) LIKE '%/vllm'
   OR LOWER(CommandLine) LIKE '%ollama serve%'
   OR LOWER(CommandLine) LIKE '%llama-server%'
   OR LOWER(CommandLine) LIKE '%vllm serve%';
```

Rule 5 · Suspicious Container Capability Additions · process_creation · HIGH `T1610`

```
SELECT *
FROM process_creation
WHERE LOWER(CommandLine) LIKE '%docker%'
  AND LOWER(CommandLine) LIKE '%run%'
  AND (
      LOWER(CommandLine) LIKE '%--cap-add%'
      OR LOWER(CommandLine) LIKE '%--privileged%'
  );
```

Rule 6 · Script Interpreter Download Cradle on Server · process_creation · MEDIUM `T1105` `T1105`

```
SELECT *
FROM process_creation
WHERE (
      LOWER(Image) LIKE '%/curl'
      OR LOWER(Image) LIKE '%/wget'
      OR LOWER(Image) LIKE '%\curl.exe'
      OR LOWER(Image) LIKE '%\certutil.exe'
  )
  AND (
      LOWER(CommandLine) LIKE '%http://%'
      OR LOWER(CommandLine) LIKE '%https://%'
      OR LOWER(CommandLine) LIKE '% -o %'
  );
```

8. References

- Microsoft: Storm-0940 uses ARTEX
- Google GTIG: ARTEX and Friends, From Prompting to Autonomy (Sep 2026), AI vuln exploitation (May 2026), AI Threat Tracker (Nov 2025)
- CrowdStrike: VAPOR PANDA, Korea finance, Agentic AI campaigns
- Anthropic: Sep 2026 threat report, GTG-1002
- Wiz: ARTEX in the wild
- Unit 42: ARTEX timely intel
- Zscaler: Korea finance, Global campaign
- Gambit: $25 a company; CSA: Cairn note, TrapDoor
- Check Point: HexStrike-AI; The Hacker News: Villager
- Detection: theori-io/aipt-detections, Nextron, RST Cloud harness hunting
- ARTEX (section 3.A): Guardz: ARTEX agentic AI framework, SOC Prime: ARTEX leads a new wave, jiwoochris/artex-global (README, `docs/defense-en.md`, indicators CSV), plus the GTIG, Microsoft, CrowdStrike Korea, Wiz and RST Cloud posts above
- CRAFTY FALCON (section 2A): CSIRT-Fin Advisory 261830103401_ADD.2 via Advisory No. 15/2026; public coverage: Economic Times, CNA/Reuters, Indian Express, Elets BFSI; CyberStrikeAI intrusion analysis (Kaspersky, via OSINTsights)
- Press: Japan Times (Japan ARTEX), Financial News (Korea FSC), Reddit r/cybersecurity PentAGI self-test

---

AiStrike · Threat Advisory · AI-driven offensive frameworks · 5 November 2026. The full indicator list is in Annexure A and the companion CSV.

Annexure A. Indicators by actor and campaign

IP addresses and domains are defanged, with each dot written as `[.]`. For CRAFTY FALCON we list only the indicators marked for blocking. In the other tables, block means block once alert mode has confirmed it is safe, alert means raise a detection, monitor or hunt means use it for retro-hunting, and review means false positives are known.

CRAFTY FALCON (12)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	ACTION
`artex[.]ok7777[.]eu[.]cc`	domain	ARTEX	ARTEX console hostname	BLOCK
`artex[.]tokena2b[.]top`	domain	ARTEX	ARTEX console hostname	BLOCK
`artex[.]noog2infra[.]site`	domain	ARTEX	ARTEX console hostname	BLOCK
`artex[.]*[.]sslip[.]io`	domain	ARTEX	Wildcard-DNS ARTEX console hostnames (artex.<ip>.sslip.io)	BLOCK
`f4febc55ea12b31ae17cfb7e614afda8`	ja3s	CyberStrikeAI	C2 panel JA3S (TLS 1.3); stable across IP rotation	BLOCK
`174a48523e8f428b7e9132e81e6ccf9f`	ja3s	CyberStrikeAI	C2 panel JA3S (TLS 1.2 fallback)	BLOCK
`t130200_1301_a56c5b993250`	ja4s	CyberStrikeAI	C2 panel JA4S	BLOCK
`CN=CyberStrikeAI`	tls_cert	CyberStrikeAI	Self-signed ECDSA P-256 panel cert (variant CN=localhost)	BLOCK
`-1349384020`	favicon_mmh3	CyberStrikeAI	Panel favicon /static/favicon.ico	BLOCK
`-576220750`	favicon_mmh3	ARTEX	ARTEX console favicon /favicon.ico	BLOCK
`6434073`	favicon_mmh3	ARTEX (Redex)	Korean ARTEX derivative "Redex" console favicon	BLOCK
`O=ARTEX, ST=Beijing, C=CN`	tls_cert	ARTEX	Self-signed ARTEX console cert subject; also any CN containing artex.	BLOCK

Storm-0940/UNC7228 (26)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`144[.]172[.]64[.]60`	ip	ARTEX	Sign-ins and Graph API activity via compromised ConnectWise apps	Microsoft, Wiz	BLOCK
`144[.]172[.]64[.]99`	ip	ARTEX	Password spray and ConnectWise app sign-ins	Microsoft, Wiz	BLOCK
`144[.]172[.]64[.]128`	ip	ARTEX	Sign-ins and Graph API activity	Microsoft, Wiz	BLOCK
`104[.]194[.]150[.]21`	ip	ARTEX	IdP and API probing of IT vendors	Wiz	BLOCK
`167[.]88[.]172[.]97`	ip	ARTEX	IdP and API probing of IT vendors	Wiz	BLOCK
`187[.]15[.]88[.]186`	ip	ARTEX	VPN IP Graph access 2026-10-15/16 only	Microsoft	MONITOR
`187[.]15[.]89[.]186`	ip	ARTEX	VPN IP Graph access 2026-10-15/16 only	Microsoft	MONITOR
`187[.]15[.]90[.]217`	ip	ARTEX	VPN IP Graph access 2026-10-15/16 only	Microsoft	MONITOR
`506e08f9-6e9a-4da4-942b-a099a03644ae`	app_id	ARTEX	ConnectWise Cloud Backup (compromised multi-tenant app)	Microsoft	REVIEW
`24205314-2b89-4e07-a2a4-579e898a5640`	app_id	ARTEX	ConnectWise - MS Teams	Microsoft	REVIEW
`0de14c77-b9b0-4137-9360-34878a929827`	app_id	ARTEX	ConnectWise - MS Graph	Microsoft	REVIEW
`Python-urllib/3.12`	user_agent	ARTEX	Noisy alone; correlate	Microsoft, Wiz	MONITOR
`python-requests/2.31.0`	user_agent	ARTEX	Noisy alone; correlate	Microsoft	MONITOR
`python-requests/2.34.2`	user_agent	ARTEX	Noisy alone; correlate	Microsoft, Wiz	MONITOR
`curl/7.88.1`	user_agent	ARTEX	Noisy alone; correlate	Microsoft, Wiz	MONITOR
`azure-toolkit/1.0`	user_agent	ARTEX	Graph/sign-in UA	Microsoft	MONITOR
`norma/0.4`	user_agent	ARTEX	Norma agent engine UA	Wiz, Nextron	ALERT
`Boto3/1.43.113 md/Botocore#1.43.113 ua/2.1 os/linux#6.8.0-147-generic`	user_agent	ARTEX	AWS API UA prefix	Wiz	MONITOR
`aws-cli/1.46.1 md/Botocore#1.43.62`	user_agent	ARTEX	AWS CLI UA prefix	Wiz	MONITOR
`ReconProbe/1.0`	user_agent	ARTEX	Recon probe UA in application-layer phase	Wiz	ALERT
`Boto3/1.43.112`	user_agent	ARTEX	AWS SDK UA seen in Cognito and AWS probing	Wiz	HUNT
`/zzz-nonexistent-xyz123`	path	ARTEX	Baseline 404 canary before real route calls	Wiz	ALERT
`eyJhbGciOiJub25lIn0.eyJlbWFpbCI6InRlc3RAdGVzdC5jb20ifQ.`	string	ARTEX	Unsigned alg:none JWT (email test@test.com) sent to 60+ SSO routes	Wiz	ALERT
`state=notvalidb64!!`	string	ARTEX	OAuth callback tamper with code=AAAA	Wiz	ALERT
`/api/users/generate_redirect_link?role=admin`	path	ARTEX	Admin redirect link probe	Wiz	HUNT
`/api/v1/Users/GeneratePasswordResetToken`	path	ARTEX	Unauthenticated reset token generation	Wiz	HUNT

VAPOR PANDA (10)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`C:\ProgramData\artx5v\`	path	ARTEX	ARTEX working dir	CrowdStrike	ALERT
`nginxmysql[.]com`	domain	ARTEX	BLOODALCHEMY C2	CrowdStrike	BLOCK

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`45[.]153[.]126[.]223`	ip	ARTEX	BLOODALCHEMY C2 (nginxmysql[.]com)	CrowdStrike	BLOCK
`sqlserver-sqlite[.]com`	domain	ARTEX	BLOODALCHEMY C2	CrowdStrike	BLOCK
`89[.]46[.]234[.]50`	ip	ARTEX	BLOODALCHEMY C2 (sqlserver-sqlite[.]com)	CrowdStrike	BLOCK
`www[.]ddduuuppp[.]com`	domain	ARTEX	BLOODALCHEMY C2	CrowdStrike	BLOCK
`192[.]121[.]47[.]20`	ip	ARTEX	BLOODALCHEMY C2 (ddduuuppp.com)	CrowdStrike	BLOCK
`49fe70419fd19b81ca8d730a6ef7c1ec71b681d60b390761070d42686c7d2068`	sha256	ARTEX	GameHook binary renamed svchost.exe	CrowdStrike	BLOCK
`3fb5a9fb3e2cb1d6208966bd4e55d10282d19c7250d827efa560463d60a12ee4`	sha256	ARTEX	BLOODALCHEMY loader graphics-hook-filter64.dll	CrowdStrike	BLOCK
`364251838edfcf5ef1107fa692da3c5bd9a540482449f645992830134bf807ab`	sha256	ARTEX	BLOODALCHEMY payload graphics-hook_config.bin	CrowdStrike	BLOCK

Korea/Japan finance operator (17)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`38[.]244[.]50[.]120`	ip	ARTEX	ARTEX instance / jump server (:18899/.claude/CLAUDE.md)	CrowdStrike, Zscaler	BLOCK
`23[.]248[.]226[.]250`	ip	ARTEX	Work server	Zscaler	BLOCK
`23[.]248[.]226[.]251`	ip	ARTEX	Work server	Zscaler	BLOCK
`23[.]248[.]226[.]252`	ip	ARTEX	Work server	Zscaler	BLOCK
`23[.]248[.]226[.]253`	ip	ARTEX	Work server	Zscaler	BLOCK
`23[.]248[.]226[.]254`	ip	ARTEX	Work server	Zscaler	BLOCK
`161[.]8[.]161[.]140`	ip	ARTEX	Upstream SOCKS proxy :11005	Zscaler	BLOCK
`101[.]53[.]80[.]20`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`205[.]214[.]59[.]31`	ip	ARTEX	Proxy / upstream SOCKS	CrowdStrike, Zscaler	BLOCK
`124[.]155[.]252[.]63`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`154[.]201[.]79[.]246`	ip	ARTEX	Proxy pool API	CrowdStrike, Zscaler	BLOCK
`23[.]248[.]249[.]90`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`23[.]158[.]220[.]98`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`103[.]248[.]148[.]84`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`203[.]160[.]133[.]172`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`209[.]209[.]85[.]38`	ip	ARTEX	Proxy	CrowdStrike	BLOCK
`xcai[.]pro`	domain	ARTEX	LLM API reseller (DeepSeek via OpenAI-compatible API)	CrowdStrike, Zscaler	MONITOR

Gambit e-commerce operator (17)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`155[.]254[.]22[.]215`	ip	Cairn/Strix/Hermes	Staging/command server and AI console	Gambit	BLOCK
`209[.]126[.]4[.]170`	ip	Cairn/Strix/Hermes	DNS exfil and HTTP listeners	Gambit	BLOCK
`213[.]21[.]239[.]62`	ip	Cairn/Strix/Hermes	C2	Gambit	BLOCK
`172[.]245[.]224[.]188`	ip	Cairn/Strix/Hermes	C2	Gambit	BLOCK
`172[.]245[.]89[.]137`	ip	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`medbooksource[.]com`	domain	Cairn/Strix/Hermes	Operator console / OOB / DNS exfil	Gambit	BLOCK
`traffic-analyzer[.]net`	domain	Cairn/Strix/Hermes	C2	Gambit	BLOCK
`b8t[.]shop`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`cdn[.]netlfjs[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`x1opay[.]co`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`static-js[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`js-static[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`cdn[.]js-static[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`jsnetlify[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`netlifyjs[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`newssjs[.]com`	domain	Cairn/Strix/Hermes	Skimmer host	Gambit	BLOCK
`new Function(atob(`	string	Cairn/Strix/Hermes	Skimmer loader pattern with .slice(7) junk prefix	Gambit	HUNT

GTG-20006 (Midnight Blizzard-linked) (46)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`ms365-live[.]com`	domain	Claude Code workflows	Phishing / device-code	Anthropic	BLOCK

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`teams[.]ms365-live[.]com`	domain	Claude Code workflows	Phishing	Anthropic	BLOCK
`m365-owa[.]com`	domain	Claude Code workflows	Phishing	Anthropic	BLOCK
`owa-ms365[.]com`	domain	Claude Code workflows	Phishing	Anthropic	BLOCK
`ms365-device[.]com`	domain	Claude Code workflows	Device-code phishing	Anthropic	BLOCK
`mslivetest[.]duckdns[.]org`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`my-invite[.]org`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`chamber-ua[.]org`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`chathamhouse[.]eu`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`ukrinform-share[.]net`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`statistic-ms[.]live`	domain	Claude Code workflows	C2	Anthropic	BLOCK
`static-ms[.]live`	domain	Claude Code workflows	C2	Anthropic	BLOCK
`ad-g[.]org`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`docs-viewer[.]org`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`wa-connect[.]eu`	domain	Claude Code workflows	WhatsApp takeover	Anthropic	BLOCK
`wa-meeting[.]com`	domain	Claude Code workflows	WhatsApp takeover	Anthropic	BLOCK
`mygreatmarket[.]org`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`mygreatmarket[.]com`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`cdncounter[.]net`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`stuseamandesilt[.]org`	domain	Claude Code workflows	C2 (api/cdn/update subdomains)	Anthropic	BLOCK
`itechx[.]tel`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`pdfviewer2024[.]b-cdn[.]net`	domain	Claude Code workflows	Payload host	Anthropic	BLOCK
`meridian-protocol[.]org`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`meridiangroup-corp[.]com`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`projectnightcrawler[.]dev`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`metricwave[.]org`	domain	Claude Code workflows	Infra	Anthropic	BLOCK
`mgsend[.]org`	domain	Claude Code workflows	Mailer	Anthropic	BLOCK
`russianearabroad[.]com`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`russianearabroad[.]org`	domain	Claude Code workflows	Lure	Anthropic	BLOCK
`104[.]145[.]210[.]184`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`31[.]57[.]243[.]154`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`104[.]194[.]151[.]133`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`104[.]194[.]159[.]55`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`144[.]172[.]114[.]192`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`213[.]145[.]86[.]112`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`2[.]26[.]53[.]194`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`148[.]135[.]195[.]111`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`185[.]198[.]234[.]26`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`185[.]198[.]234[.]101`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`149[.]54[.]42[.]106`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`104[.]194[.]149[.]228`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`38[.]146[.]28[.]132`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`38[.]146[.]28[.]75`	ip	Claude Code workflows	Infra	Anthropic	BLOCK
`be99857449d2856dd5a84e21c8a3d5e0e01456adb44062ddec5a6b4970d8d42c`	sha256	Claude Code workflows	Malware sample	Anthropic	BLOCK
`918fa52ae45ed60ba7cc8bdc99c3cbe9ab92e0375ec31fc05d0d4513be11c593`	sha256	Claude Code workflows	Malware sample	Anthropic	BLOCK
`msedgeupdate_v3.exe|msedgeupdate.exe|version.dll|WUEngine.exe|DiagHost.exe|fix_network.apk`	filename	Claude Code workflows	Implant filenames	Anthropic	HUNT

GTG-50014 (ShinyHunters) (26)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`updatebeacon[.]duckdns[.]org`	domain	Claude agents	Infra	Anthropic	BLOCK
`soraki[.]cc`	domain	Claude agents	Carding platform	Anthropic	BLOCK
`soraki[.]work`	domain	Claude agents	Carding platform	Anthropic	BLOCK
`policenationale[.]cc`	domain	Claude agents	Carding storefront	Anthropic	BLOCK
`emailsecure[.]email`	domain	Claude agents	Phishing	Anthropic	BLOCK

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`mozilla[.]ws`	domain	Claude agents	Infra	Anthropic	BLOCK
`signin-1pssword[.]com`	domain	Claude agents	Phishing	Anthropic	BLOCK
`on-pssword[.]com`	domain	Claude agents	Phishing	Anthropic	BLOCK
`ari-chain[.]com`	domain	Claude agents	Crypto phishing	Anthropic	BLOCK
`arichain[.]network`	domain	Claude agents	Crypto phishing	Anthropic	BLOCK
`bitmart-mystery[.]com`	domain	Claude agents	Crypto phishing	Anthropic	BLOCK
`defi-claim[.]xyz`	domain	Claude agents	Crypto phishing	Anthropic	BLOCK
`service-infos[.]info`	domain	Claude agents	Infra	Anthropic	BLOCK
`162[.]128[.]129[.]106`	ip	Claude agents	Egress 2026-02-20..03-10	Anthropic	MONITOR
`195[.]178[.]110[.]131`	ip	Claude agents	Egress 2026-03-12..04-30	Anthropic	MONITOR
`45[.]148[.]10[.]242`	ip	Claude agents	Egress 2026-04-06..04-27	Anthropic	MONITOR
`92[.]118[.]39[.]3`	ip	Claude agents	Egress 2026-04-10..04-19	Anthropic	MONITOR
`185[.]65[.]134[.]246`	ip	Claude agents	Egress (VPN) 2026-04-19..05-04	Anthropic	MONITOR
`185[.]65[.]134[.]199`	ip	Claude agents	Egress (VPN) 2026-04-19..04-28	Anthropic	MONITOR
`193[.]32[.]249[.]161`	ip	Claude agents	Egress (VPN)	Anthropic	MONITOR
`193[.]32[.]249[.]164`	ip	Claude agents	Egress (VPN)	Anthropic	MONITOR
`193[.]32[.]249[.]170`	ip	Claude agents	Egress (VPN)	Anthropic	MONITOR
`104[.]36[.]50[.]54`	ip	Claude agents	Egress 2026-04-24	Anthropic	MONITOR
`104[.]193[.]135[.]207`	ip	Claude agents	Egress 2026-04-05	Anthropic	MONITOR
`91[.]171[.]138[.]169`	ip	Claude agents	Egress 2026-04-19..04-21	Anthropic	MONITOR
`176[.]177[.]12[.]62`	ip	Claude agents	Egress 2026-04-19..04-20	Anthropic	MONITOR

GTG-50020 (8)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`141[.]133[.]125[.]208`	ip	PentAGI-style pipeline	Egress 2026-05-21..05-23	Anthropic	MONITOR
`167[.]250[.]111[.]136`	ip	PentAGI-style pipeline	Egress 2026-05-23..06-03	Anthropic	MONITOR
`178[.]16[.]54[.]141`	ip	PentAGI-style pipeline	Egress 2026-05-21..06-16	Anthropic	MONITOR
`37[.]27[.]103[.]22`	ip	PentAGI-style pipeline	Egress 2026-05-26..06-13	Anthropic	MONITOR
`194[.]163[.]183[.]216`	ip	PentAGI-style pipeline	Egress 2026-05-23..05-24	Anthropic	MONITOR
`202[.]66[.]167[.]230`	ip	PentAGI-style pipeline	Egress 2026-05-21..06-04	Anthropic	MONITOR
`146[.]103[.]101[.]253`	ip	PentAGI-style pipeline	Egress 2026-05-21..06-13	Anthropic	MONITOR
`146[.]103[.]97[.]169`	ip	PentAGI-style pipeline	Egress 2026-05-21..05-25	Anthropic	MONITOR

GTG-50029 (8)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`139[.]59[.]2[.]243`	ip	PentAGI-style sub-agents	Key-validation box	Anthropic	MONITOR
`136[.]144[.]242[.]56`	ip	PentAGI-style sub-agents	Staging and scan box	Anthropic	MONITOR
`163[.]172[.]157[.]53`	ip	PentAGI-style sub-agents	Persistent dedicated server	Anthropic	MONITOR
`34[.]156[.]199[.]132`	ip	PentAGI-style sub-agents	Exfil endpoint (hijacked GCP)	Anthropic	MONITOR
`34[.]156[.]95[.]176`	ip	PentAGI-style sub-agents	Exfil endpoint (hijacked GCP)	Anthropic	MONITOR
`frntrs-analytics[.]dedyn[.]io`	domain	PentAGI-style sub-agents	BeEF browser C2	Anthropic	BLOCK
`prod-artfkt[.]com`	domain	PentAGI-style sub-agents	Operational domain	Anthropic	BLOCK
`fafwatch[.]xyz`	domain	PentAGI-style sub-agents	Doxxing domain	Anthropic	BLOCK

GTG-50021 (7)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`awstore[.]cloud`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK
`kiro[.]cheap`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK
`sys-tools[.]cfd`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK
`aws-us-east-3[.]com`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK
`holdboost[.]store`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK
`deltaclient[.]xyz`	domain	Fake AI reseller	Credential harvester infra	Anthropic	BLOCK

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`iymkjuzymkapovrntoxy[.]supabase[.]co`	domain	Fake AI reseller	Credential harvester backend	Anthropic	BLOCK

ARTEX (unattributed) (9)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`31[.]40[.]205[.]155`	ip	ARTEX	Serves Basic realm ARTEX + Sub2API :8080; self-registration and API abuse	Unit42	MONITOR
`173[.]254[.]220[.]78`	ip	ARTEX	Sub2API gateway; recon and account recovery probing	Unit42	MONITOR
`103[.]100[.]158[.]103`	ip	ARTEX	Scanning and OAuth probing; lanmao666 subdomain	Unit42	MONITOR
`152[.]233[.]51[.]12`	ip	ARTEX	Probing and unauthenticated API writes	Unit42	MONITOR
`lanmao666[.]com`	domain	ARTEX	Multi-tenant LLM gateway	Unit42	MONITOR
`lanmao666[.]xyz`	domain	ARTEX	Multi-tenant LLM gateway	Unit42	MONITOR
`Mozilla/5.0 (security-testing; authorized)`	user_agent	ARTEX	Self-describing agent UA	Unit42	ALERT
`Mozilla/5.0 (Windows NT 10.0; Win64; x64) recon-audit`	user_agent	ARTEX	Self-describing agent UA	Unit42	ALERT
`Fuzz Faster U Fool v2.1.0-dev`	user_agent	ARTEX	ffuf worker	Unit42	ALERT

ARTEX (10)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`WWW-Authenticate: Basic realm="ARTEX"`	http_header	ARTEX	ARTEX console fingerprint (scan)	Unit42, RSTCloud	HUNT
`zz_no_such_user_9f3a`	string	ARTEX	Canary username	Unit42	ALERT
`zzq.nonexistent9x7`	string	ARTEX	Canary username	Unit42	ALERT
`precheck.not-a-user@example.invalid`	string	ARTEX	Canary email	Unit42	ALERT
`zzzrandom12345`	string	ARTEX	Canary path	Unit42	ALERT
`nonexistent-i3185`	string	ARTEX	Canary S3 key	Unit42	ALERT
`zz-no-such-key-1a2b3c.html`	string	ARTEX	Canary S3 key	Unit42	ALERT
`artex-probe-*`	filename	ARTEX	Probe files with 5-digit incremental suffix	GTIG	ALERT
`ArtexProbe`	string	ARTEX	Created account naming	GTIG	ALERT
`cl1217.jsp|f1219a.jsp|z1219.jsp`	filename	ARTEX	JSP webshells via MySQL general_log poisoning	GTIG	ALERT

Any ARTEX operator (9)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`artex-enrich/1.0`	user_agent	ARTEX	Enrich module DNS/HTTP title grab against targets	artex-global, Guardz	ALERT
`artex-selfupdate`	user_agent	ARTEX	Self-update check outbound from an ARTEX host	artex-global	ALERT
`【ARTEX 平台管控·非目标防御】`	string	ARTEX	Guard/intercept audit marker in ARTEX logs and DB	artex-global, Guardz	ALERT
`8787`	port	ARTEX	Default console and REST API port	artex-global, Guardz	HUNT
`127.0.0.1:8788`	port	ARTEX	Local recording MITM proxy injected into worker tools	artex-global, Guardz	HUNT
`mitmproxy-ca-cert.pem`	filename	ARTEX	CA cert written by ARTEX proxy; referenced by SSL_CERT_FILE/CURL_CA_BUNDLE	artex-global	HUNT
`exploration_nodes`	string	ARTEX	ARTEX Postgres table (also exploration_edges, exploration_anchors, task_scope, agent_prompts)	artex-global, Guardz	HUNT
`ARTEX_LLM_BASE_URL`	string	ARTEX	ARTEX environment variable (also ARTEX_LLM_PROVIDER, ARTEX_PG_DSN, ARTEX_SETUP_TOKEN)	artex-global	HUNT
`ghcr.io/jiwoochris/artex-global`	string	ARTEX	Public ARTEX container image (defensive fork, same binary)	artex-global	REVIEW

ARTEX simulation (2)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`TinyTestBlob`	string	ARTEX	Azure blob created then deleted in cloud permission probe simulation	SOC Prime	HUNT
`aws ec2 describe-instances --dry-run`	string	ARTEX	Dry-run permission probe (DryRunOperation)	SOC Prime	HUNT

Exposed PentAGI consoles (2)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`2a620e3c5f7b49dfc6f5278352607f7931f8c38ef9d4689908af65538dd95366`	sha256	PentAGI	PentAGI favicon SHA-256 (Netlas)	RST Cloud	HUNT
`AS398823`	asn	PentAGI	About 90 exposed PentAGI consoles hosted here	RST Cloud	MONITOR

Exposed ARTEX consoles (5)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`ARTEX: 자율 침투 테스트 콘솔`	string	ARTEX	Korean ARTEX build console title	RST Cloud	HUNT
`artex — 自主渗透测试控制台`	string	ARTEX	Lower-case ARTEX console title variant	RST Cloud	HUNT
`AegisLoop — 攻防安全运营`	string	ARTEX fork	AegisLoop console title	RST Cloud	HUNT
`ScopeWeaver — Autonomous Penetration Testing Console`	string	ScopeWeaver	ScopeWeaver console title	RST Cloud	HUNT
`AS54600`	asn	ARTEX	About 300 exposed ARTEX consoles hosted here	RST Cloud	MONITOR

Exposed Cairn consoles (1)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`Cairn · 事实图谱协同探索`	string	Cairn	Cairn console title	RST Cloud	HUNT

Exposed harness consoles (19)

INDICATOR	TYPE	FRAMEWORK	CONTEXT	SOURCE	ACTION
`23[.]226[.]135[.]117`	ip	PentAGI	Exposed PentAGI "xhackai" console on port 8443; VT 8/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`1[.]14[.]104[.]208`	ip	ARTEX;PentAGI	Exposed ARTEX + PentAGI console; VT 7/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`43[.]139[.]50[.]42`	ip	ARTEX	Exposed ARTEX v2 console on port 3000; VT 6/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`51[.]158[.]154[.]152`	ip	ARTEX;PentAGI	Exposed ARTEX + PentAGI console; VT 3/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`156[.]240[.]236[.]6`	ip	Cairn	Exposed Cairn "cf作战平台" console on port 8001; VT 2/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`8[.]216[.]41[.]60`	ip	Cairn	Exposed Cairn "Zero" console; VT 1/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`84[.]247[.]152[.]233`	ip	ARTEX;PentAGI	Exposed ARTEX + PentAGI console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`5[.]255[.]124[.]144`	ip	ScopeWeaver	Exposed ScopeWeaver console on port 8080; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`119[.]45[.]26[.]102`	ip	PentAGI	Exposed PentAGI 天融信 build console on port 18089; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`123[.]57[.]3[.]71`	ip	Cairn	Exposed Cairn "VulnFlow IoT" console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`176[.]65[.]144[.]71`	ip	Cairn	Exposed Cairn console; VT 15/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`43[.]250[.]53[.]42`	ip	Cairn	Exposed Cairn console; VT 12/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`117[.]72[.]220[.]70`	ip	ARTEX	Exposed ARTEX console; VT 12/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`31[.]57[.]47[.]29`	ip	Cairn	Exposed Cairn console; VT 6/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`38[.]11[.]221[.]26`	ip	ARTEX	Exposed ARTEX console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`111[.]228[.]36[.]150`	ip	ARTEX	Exposed ARTEX console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`82[.]25[.]190[.]221`	ip	PentAGI	Exposed PentAGI console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`124[.]223[.]104[.]170`	ip	PentAGI	Exposed PentAGI console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR
`23[.]95[.]170[.]215`	ip	PentAGI	Exposed PentAGI console; VT 0/92 at report time; exposure only, not attribution	RST Cloud	MONITOR