

Forum:[Hack Forums](#)

Thread: [\[Feedback\] White Falcon Self-Hosted Multi-Agent Internet Inventory & SOC/Endpoint O](#)

itcorp - August 8, 2026, 7:18 pm

White Falcon

Self-hosted multi-agent platform for internet host inventory, distributed scanning, and SOC/endpoint operations.

Looking for honest opinions on architecture, detection design, and what to build next - not a sales pitch.

Summary

White Falcon lets you run your own agent fleet, scan authorized IP ranges or country CIDRs, enrich hosts (ports, banners, TLS, tech/device fingerprints, vulns, geo/WHOIS), store everything in your database, search it with Shodan-style filters, and operate from a realtime Windows desktop console. Agents can also collect endpoint inventory, baselines, alerts, and optional live screen stream.

Detection is catalog-driven: the API holds live signature packs; agents pull and cache them, match against banners/headers/body, score confidence, and send unknown/low-confidence patterns into a learning queue for human review.

Built for SOC teams, labs, MSSPs, and anyone who wants on-prem control of the scanner fleet and the inventory - not a cloud-only data feed.

What it is

Think: private internet asset intelligence you build yourself + multi-agent scanner orchestration + operator wallboard + light endpoint/EDR-style modules + a signature learning loop that improves detection over time.

It is not a public Shodan clone and not a "spray the internet" toy. It is an ops platform for authorized scanning and defensive/research use.

High-level architecture

Operator client (Windows WPF) - login, dashboard, tasks, agents, search, endpoint, admin

Central API - auth (JWT), orchestration, REST + SignalR, persistence, detection catalog

Agents (workers) - request work, scan batches, run detection pipeline, upload results, optional endpoint modules

SQL Server - tasks, hosts, signatures, plans, audit, learning queue

Flow in one line: Client/Agents <-> API (HTTP + SignalR + JWT) <-> SQL

Fig. 1 - White Falcon architecture

How a scan works (end-to-end)

Operator signs in on the desktop client

Creates a scan task (IP range or country + ports + optional feature flags/capabilities)

Online agents authenticate with an API key and connect via SignalR

API assigns non-overlapping IP batches (default ~256 IPs)

Agent runs the crawl + detection pipeline on the batch

Results upload via durable chunked ingest (begin -> chunks -> complete)

Dashboard updates live; hosts become searchable

Unknown/low-confidence patterns can enter the learning queue for review

Fig. 2 - End-to-end scan flow

How detection works

Detection is not a single black-box "AI guess." It is a layered catalog + match + score system.

1) Detection catalog (lives on the API / SQL) Port probes - probe payload + expected reply patterns for banners/service hints

Technology signatures - CMS/framework/server tech from headers, HTML, scripts, meta

Web software signatures - product/app fingerprints (+ common paths for discovery)

Device signatures - port-scoped packs for device type/vendor/model (banner + headers + body)

Vulnerability signatures - local CVE-style rules against content and detected components

External vuln feeds - NVD (optional OSV) correlation by protocol/software/tech + version

Agents authenticate, pull the catalog over the configuration API, cache it in memory, and refresh on SignalR catalog updates or on a periodic loop (~5 minutes). Admins grow the catalog over time (including via the learning loop below).

2) Match inputs and pattern engine

For each open port under analysis, the agent builds evidence from:

- Service banner
- HTTP headers (when applicable)
- HTML/body content (when applicable)

Matching uses typed patterns (contains, regex, starts/ends with, banner, hex, fingerprint). Device signatures are port-scoped - a signature only competes if it applies to that port.

3) Capability/feature flags

Every scan task can enable or disable phases (examples): PingSweep, PortScan, BannerGrab, ServiceDetection, WebScan, DeviceDetection, SSLScan, DnsLookup, OsDetection, Whois, ProtocolAnalysis, VulnScan.

If a capability is off, that detection stage is skipped - so you can run light discovery or heavy enrichment depending on the job.

4) Confidence scoring (two paths)

Content/tech/web software: scored 0–100 from match count, pattern length, coverage, header/version bonuses. Weak hits below a minimum threshold are discarded.

Device: runs signature match, vendor resolve, and endpoint discovery in parallel, then model resolve when vendors are known. Overall confidence blends those signals (plus guards like CDN/edge to cut false camera-like hits). Device learning triggers when there is no solid hit or overall confidence is around below 0.5. Content learning triggers on no detection or low max confidence.

5) Policy vs detection

Behavior policies are a separate gate (allow/deny/alert) — they do not own fingerprints. Typical hooks: geo DENY before port scan; content DENY/alert after banner/HTML analysis.

6) Learning loop

Agent posts unknown/low-confidence evidence to the learning collect endpoint

Item lands in an admin review queue

Human reviews and promotes useful patterns into Technology/WebSoftware/Device tables

Fleet gets a catalog update (SignalR) and agents refresh cache

Fig. 5 - Detection engine

Agent detection flow (step by step)

This is the concrete per-IP/per-port path an agent follows during a batch:

Host level

Blacklist skip - drop targets that must not be scanned

Ping/alive check - dead hosts are recorded as inactive and skipped for deep analysis

DNS - PTR/related records when DnsLookup is enabled

Policy/geo gate - early geo can DENY the host before ports are touched

TCP port scan - open/closed ports for the task port set

Enrich - geo, WHOIS, MAC (as available), plus OS fingerprint on open ports

Per open port

Banner/probes - HTTP/SSL grab or raw socket probes (needs BannerGrab and/or ServiceDetection)

Parallel analysis - protocol detect + SSL (on SSL ports) + content/tech/web-software/local vuln signatures + device detection

Device path detail - signature + vendor + endpoints in parallel -> model resolve -> overall confidence -> learning report if unknown/low

Optional path discovery - sensitive/admin-style paths from web-software and device endpoint hints

Extra SSL/TLS pass - when SSLScan is enabled on SSL-looking ports

External vuln correlation - NVD/OSV by protocol + detected software/tech (when VulnScan is enabled)

Content policy check - behavior rules on banner/HTML (allow / deny / alert)

After analysis

Assemble host - merge ports, enrichments, device/tech/vuln into the host record

Chunked ingest - upload to the API (begin -> chunks -> complete)

Learn (if needed) - content and device paths can independently enqueue learning samples

Fig. 3 - Agent crawl pipeline (overview)

Fig. 6 - Agent detection flow (detail)

Fig. 4 - Detection learning loop

Feature overview

Scanning and fleetMulti-agent distributed scanning (API key + SignalR, heartbeat ~15s)

IP ranges and country CIDR tasks

Blacklist, auto/recurring scan profiles

Per-task capability toggles

Chunked durable result ingest

Search and intelligenceHost inventory with Shodan-style filters, e.g.
port,country,city,org,isp,service,banner,software,tech,vuln + free text

Host detail drilldown (ports, enrichments, vulns, geo, etc.)

Operator console (WPF)Realtime wallboard - KPIs, activity, agent status, charts, world map

Tasks/scans management

Agents CRUD, status, commands, telemetry

Search, results, stream wall (multi-agent live screen)

Admin: signatures, learning review, country IP ranges, IPAM, blacklist, audit

Endpoint/EDR-lite (agent modules)Hardware/software inventory, processes and services

FIM, registry/event-oriented collection (where applicable)

Baseline/diff/alert triage

Forensics collect

Optional live screen stream to the operator console

Behavior policies + local rule evaluation

Org/platformJWT auth, roles/permissions (RBAC)

Teams, users, API keys for agents

Subscription/plan and usage hooks (quota gates)

Platform audit log

Tech stack

.NET 9

ASP.NET Core API + SignalR + JWT/Identity

EF Core + SQL Server

WPF MVVM desktop client (realtime hubs, charts, map)

Resilient HTTP (retry/circuit patterns)

External enrichments: geo, RDAP/WHOIS, NVD

Demo (there are still bugs but more on that later)

[Video: <https://www.youtube.com/watch?v=0GtdiDJwlmo>]

Screenshots

[align=center]img

Clumsy - August 9, 2026, 1:41 pm

Vouch

itcorp - August 9, 2026, 1:43 pm

(Aug 9, 2026 01:41 PM)Clumsy Wrote: [\[CITE\]](#) Vouch

This is not sale thread bud,I need your opinion.

Dystortion - August 29, 2026, 8:26 am

One of the most insane projects I've seen by a single person - this is something that large scale development teams could never do. But it's a breeze to this god right here.

itcorp - August 29, 2026, 9:02 am

(Aug 29, 2026 08:26 AM)Dystortion Wrote: [\[CITE\]](#) One of the most insane projects I've seen by a single person - this is something that large scale development teams could never do. But it's a breeze to this god right here.

This hit me right in the feels....thank you bud!! I don't think I'm anything special,I just really love what I do and I'm too stubborn to stop.But when someone like you says something like this,it reminds me why I do it.Truly grateful.