FRAME TO BIRD

HOST DEFENSE

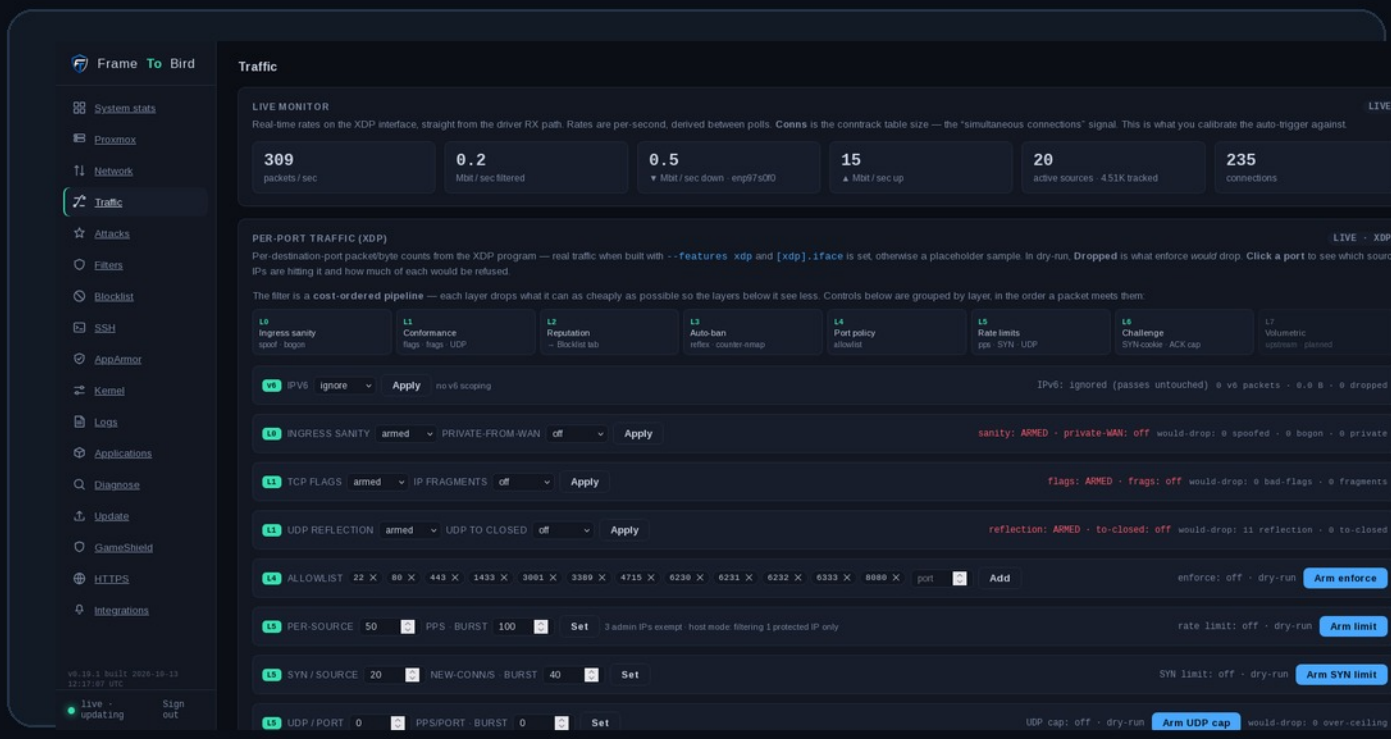
See the threat. Control the response.

A practical security console for Linux servers: XDP packet filtering, firewall operations, attack visibility, and guided response in one place.

XDP
early packet-path controls

FIREWALL
guarded rule changes

VISIBILITY
host, traffic and events



PRODUCT CAPABILITIES OVERVIEW • OCTOBER 2026

Feature availability depends on build, host, and operator configuration.

A focused console for a hard problem

FrameToBird combines a Linux control plane with an optional XDP/eBPF data plane. The data plane is designed to discard unwanted packets close to the network interface, while the control plane gives operators a place to configure policy, watch counters, and manage related host defenses.

1 Filter early

XDP can inspect and drop selected traffic before the normal socket-buffer, connection-tracking, and firewall path. The filter is organized as a cost-ordered pipeline, from basic packet checks through reputation, limits, and challenges.

LINUX / XDP

2 Operate safely

Dry-run and observe modes help operators inspect would-drop counters. Firewall changes use a confirmation window with automatic rollback if not kept. Update installation also has a rollback path.

GUARDED CHANGES

3 Understand activity

Live views combine host metrics, interface rates, per-port traffic, attack timelines, source activity, and durable event or ban records. Diagnostics and journal output support troubleshooting.

OBSERVABILITY

4 Extend response

Blocklists, feeds, source categories, honeypots, scan and SSH detection, notifications, and upstream OVH mitigation settings connect signals to operator workflows.

CONFIGURABLE

DESIGNED FOR

Game and application servers, small hosting operators, and infrastructure teams that need a visible, locally managed first line of defense on a Linux gateway or protected host.

Single-host deployment is the current operating model.

PRODUCT POSITIONING

FrameToBird helps a server absorb and manage traffic that reaches its host. It does not increase the physical capacity of an uplink or replace an upstream scrubbing provider when an attack saturates the network before packets reach the server.

What operators can manage

XDP policy & counters

Allowlisted ports, per-source limits, SYN handling, blocklist matching, auto-bans, protocol sanity/conformance controls, IPv6 mode, and live layer counters. Availability varies by kernel, build feature, NIC/driver, and configuration.

Attack and source response

DoS/DDoS monitoring, counter-based port-scan detection, durable temporary-ban history, reputation scoring, source categories, and SSH brute-force/honeypot event capture.

GameShield

Game-oriented per-port controls, including query packet-size/signature concepts and rate limits, with profiles and independent counters. Filters can be staged in dry-run before enforcement.

Firewall operations

Structured firewall editing, rule import/export, backend support and live-state views. Applying a change is paired with a timed confirm/revert workflow and lifeline allowances.

Host and network visibility

CPU, memory, disks, block devices, interfaces, traffic history, active connections, logs, AppArmor, kernel settings, and optional Proxmox host details.

Integrations & lifecycle

Threat-feed refresh, Telegram/Discord/webhook/e-mail notification configuration, optional OVH API mitigation lever, update bundles with checks and rollback, and diagnostics such as ping, DNS, TCP, path, and MTU.

Optional integrations and controls are configured by the operator; they are not automatically enabled merely because they appear in the panel.

A workflow built around visibility

1

Observe

Start with counters, event history, and dry-run modes. Confirm which traffic would be affected.

2

Tune

Set service ports, thresholds, protected addresses, administrator exemptions, and host-specific settings.

3

Arm

Enable enforcement deliberately. Some high-impact operations have confirmation prompts and safety notes in the UI.

4

Review

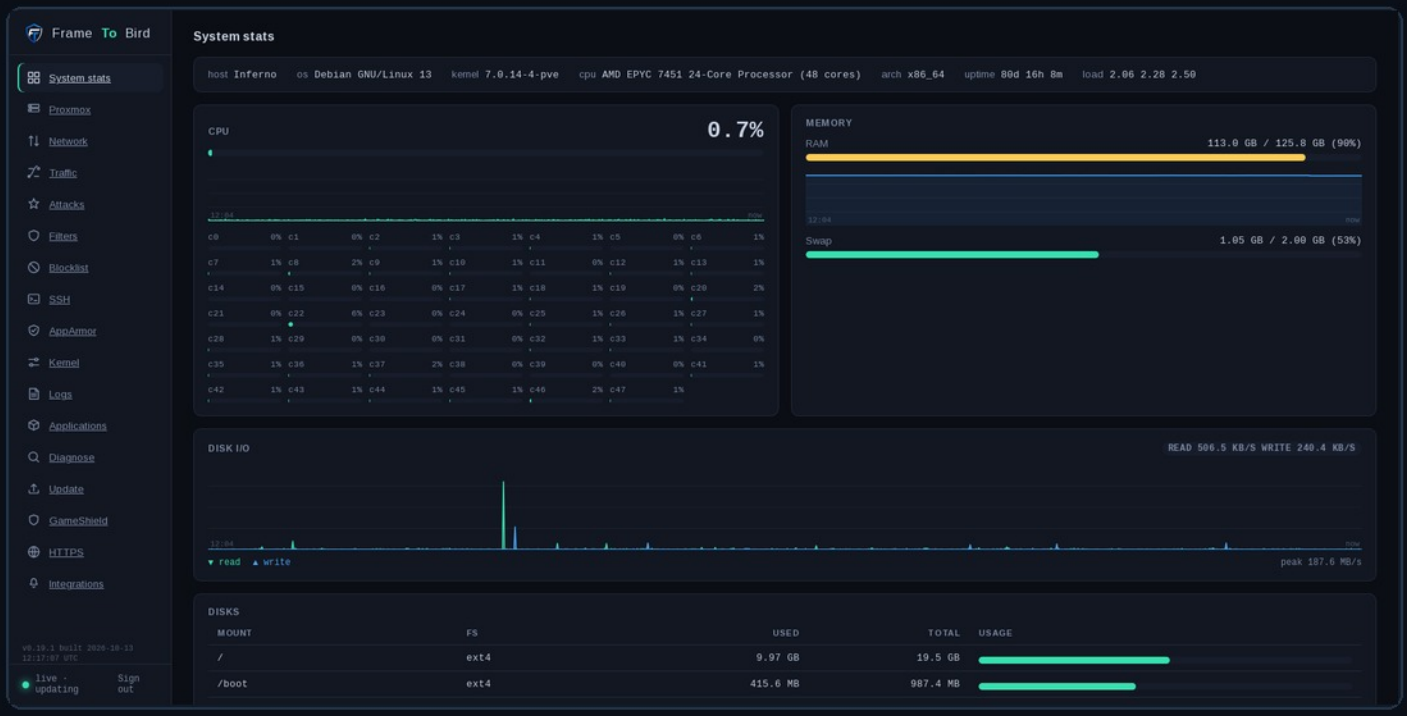
Follow live rates, per-layer drops, source events, and journal diagnostics. Revert or adjust when traffic behavior changes.

DEPLOYMENT & SECURITY NOTES

- Panel access uses a login; the configured password is stored as a hash. TLS can be configured for the panel.
- The default bind in current config code is loopback. Keep access on a trusted management network or VPN; do not publish the control surface openly.
- Firewall/XDP controls can disrupt service if misconfigured. Plan changes with console access and a recovery route.
- XDP is Linux-specific and optional at build time. The default non-XDP mode does not provide packet filtering at the NIC.

IN THE PANEL

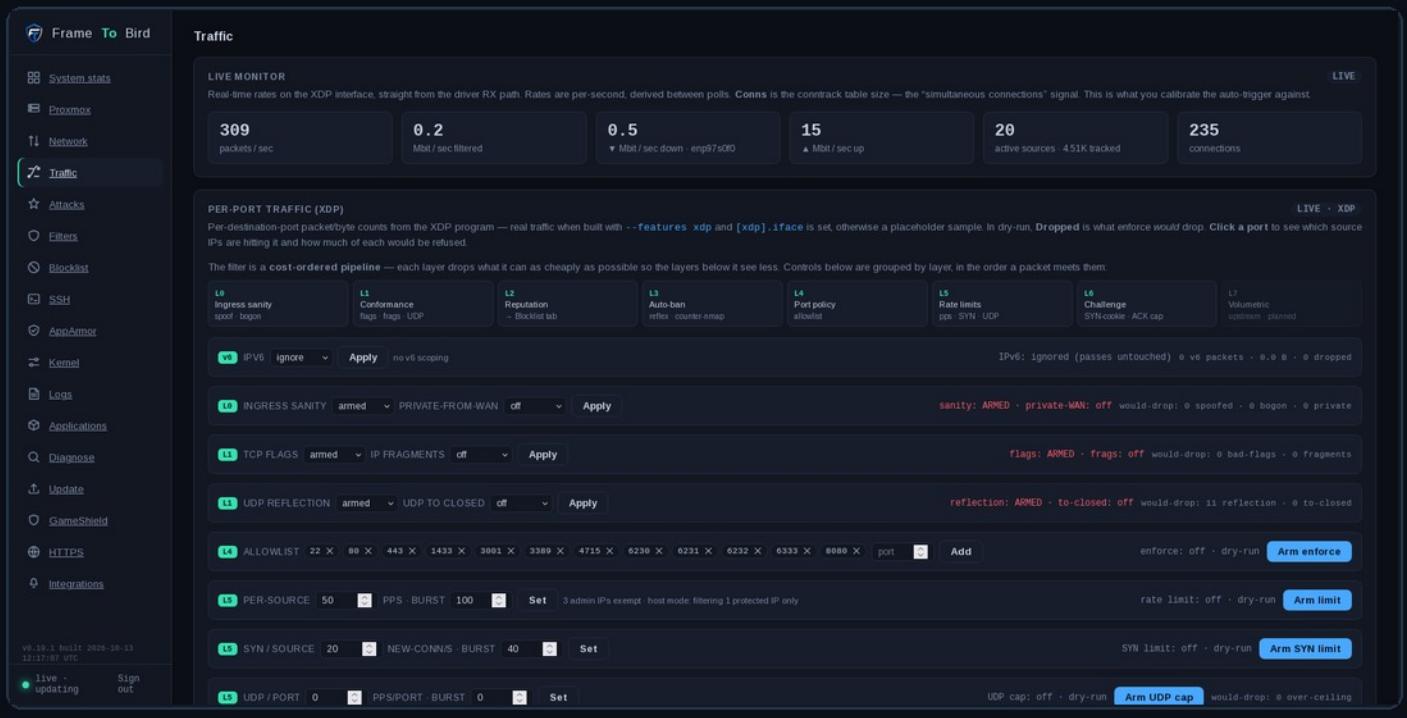
Host health at a glance



System stats brings CPU, memory, swap, storage and disk activity together. The optional Proxmox view is host-dependent; the screenshots show it alongside system health.

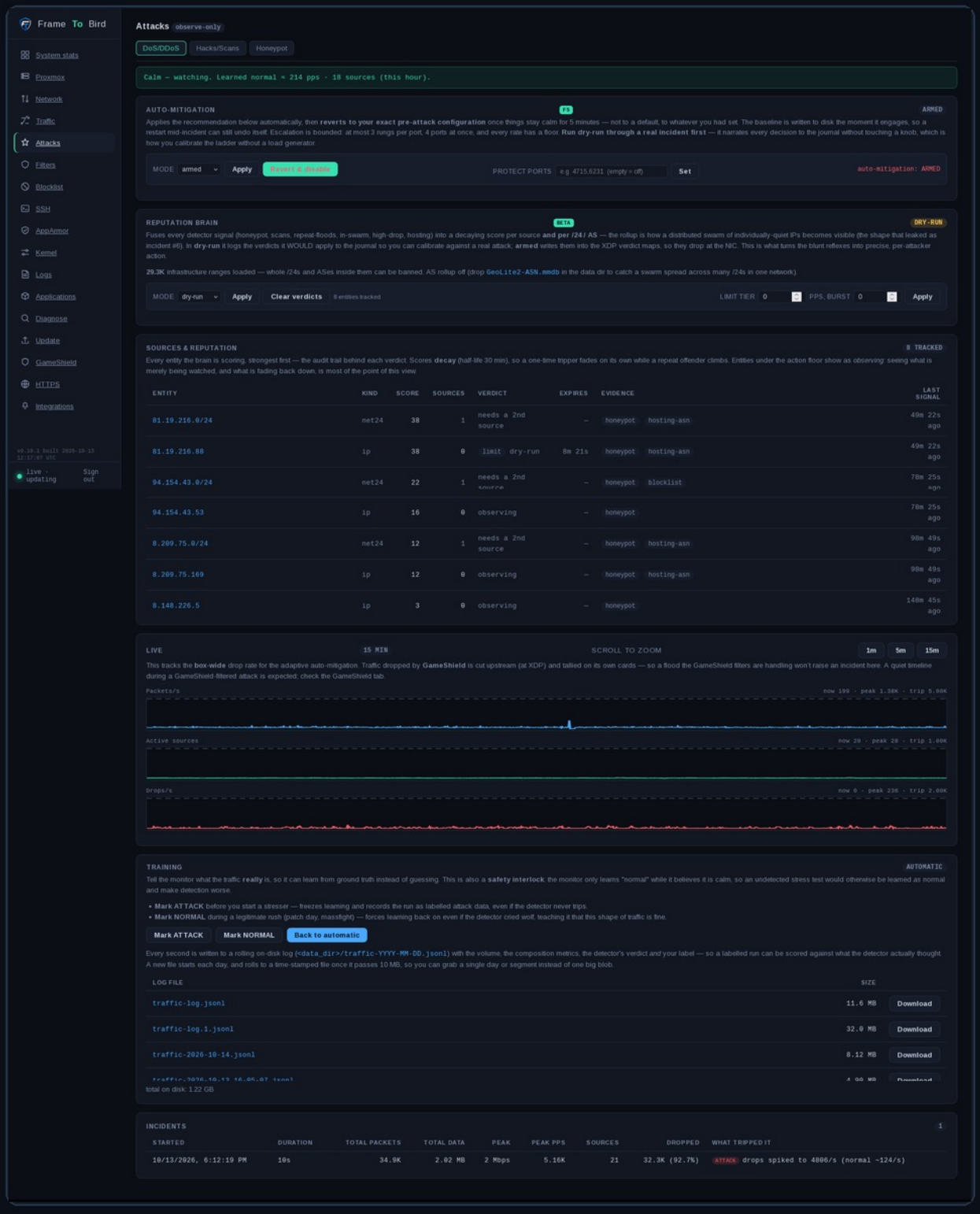
IN THE PANEL

Traffic controls with context



The Traffic view surfaces live rates and per-port activity, then groups defensive controls by pipeline layer. Dry-run status and would-drop counters help operators calibrate before enabling packet drops.

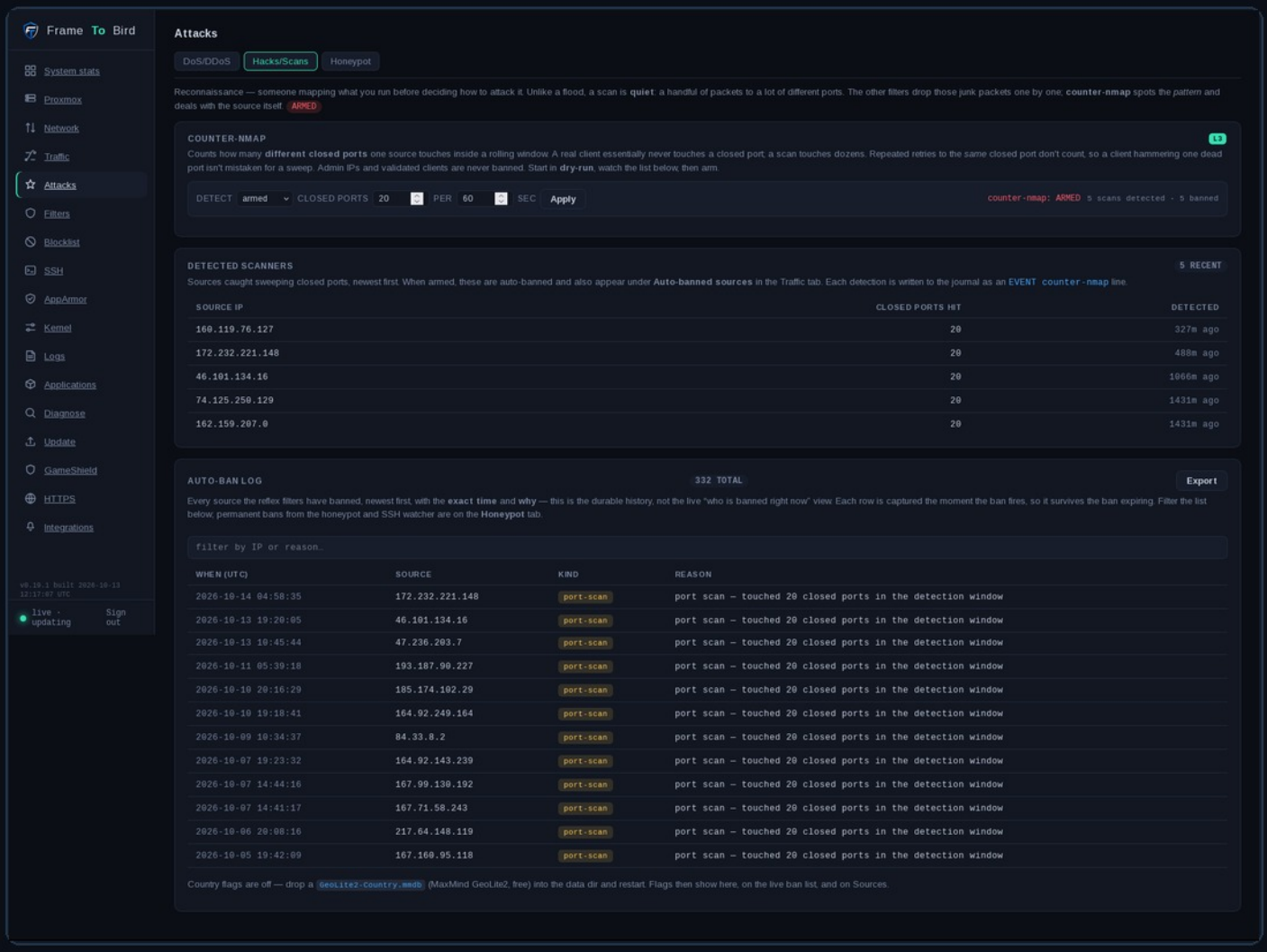
Attack monitoring and response



The DoS/DDoS view combines learned baseline context, attack timeline, reputation signals, auto-mitigation settings, training labels, traffic logs, and incident records. Displayed data is from the supplied screenshot.

IN THE PANEL

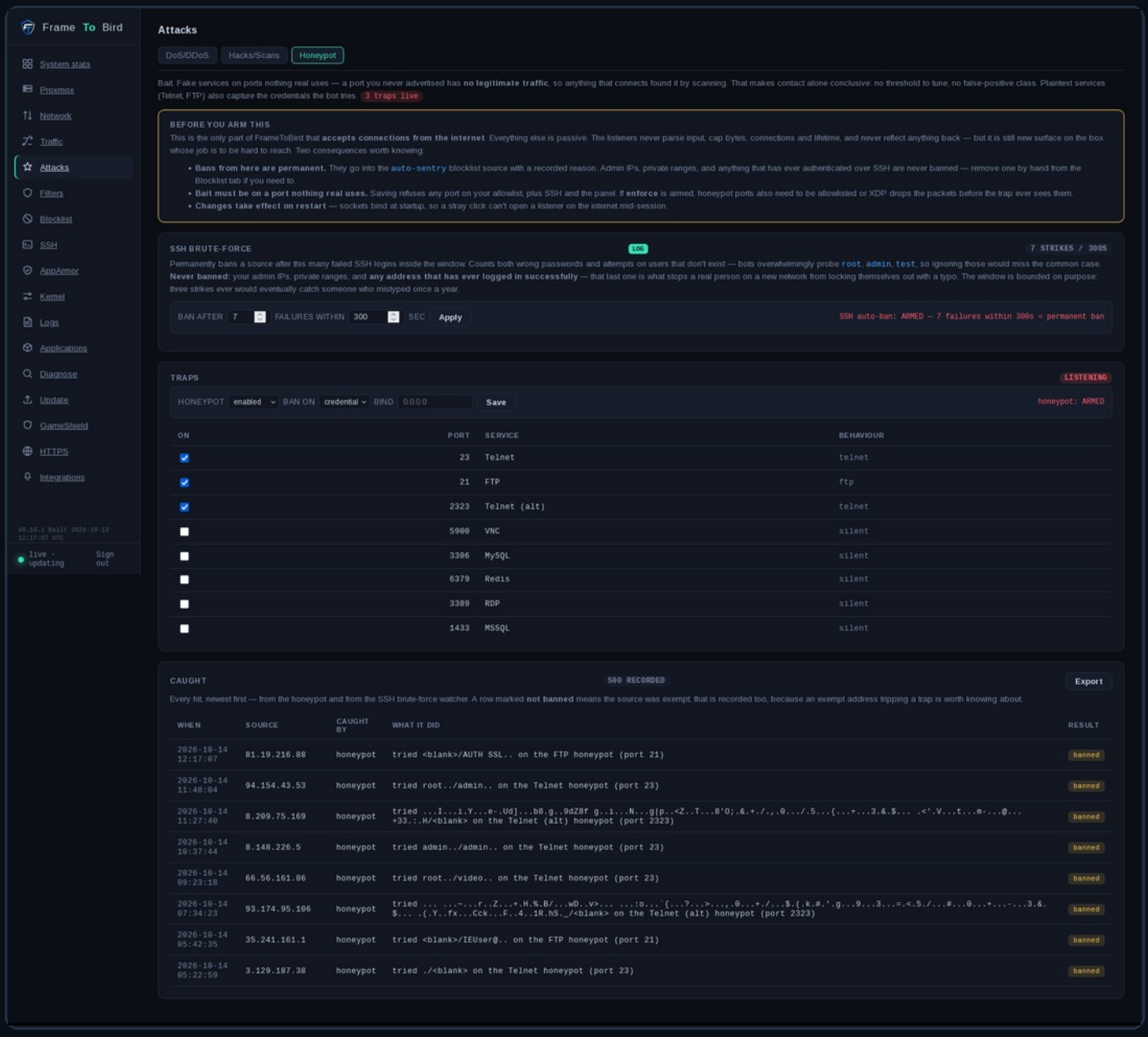
Scan detection with an audit trail



The scan view exposes counter-nmap settings, detected scanners, and a durable auto-ban log with timestamp and reason. The screenshot shows the detector armed on that host.

IN THE PANEL

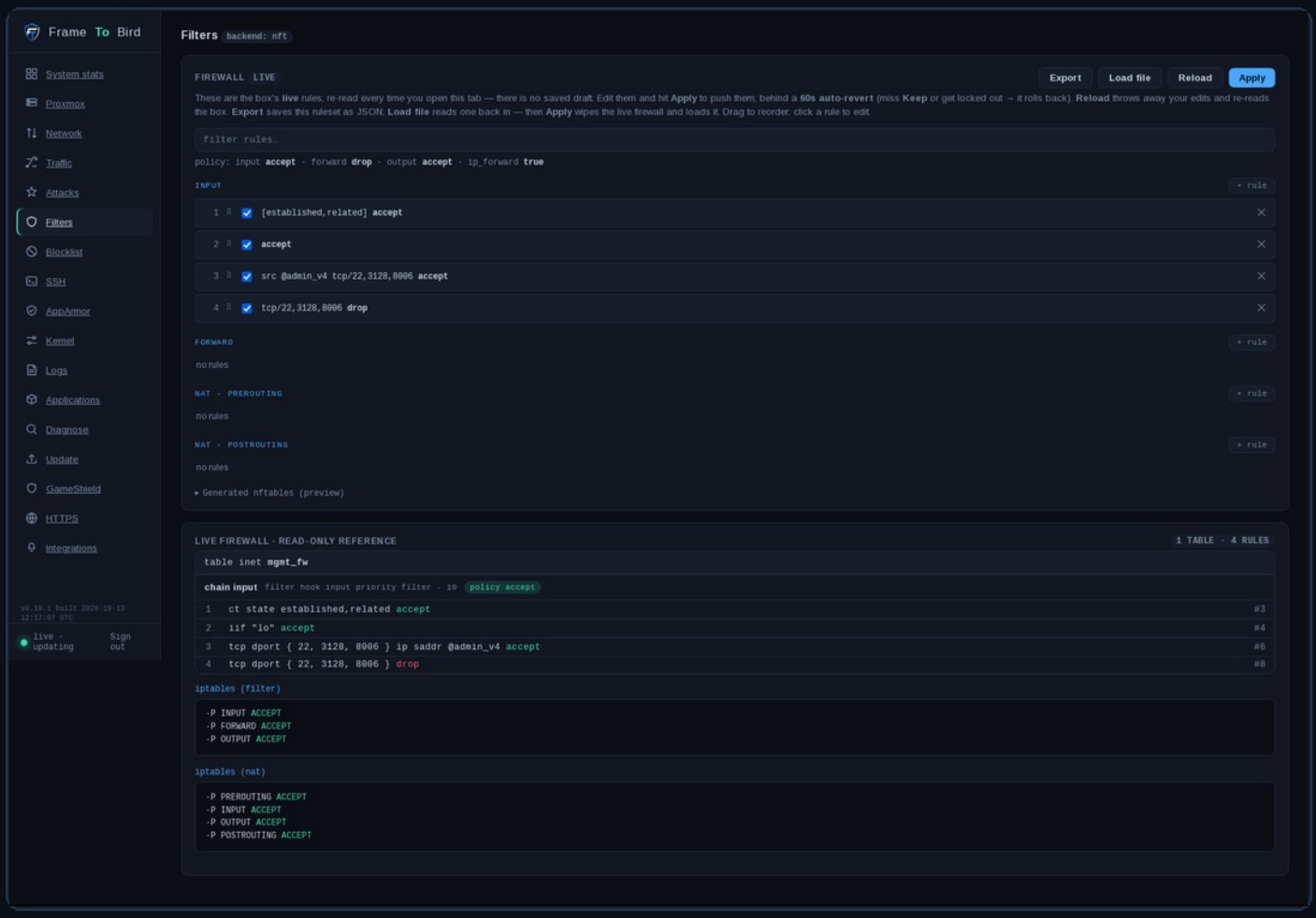
Honeypots and SSH signals



The honeypot view configures bait listeners and shows captured activity alongside SSH brute-force policy. These listeners create additional inbound services and should be selected deliberately.

IN THE PANEL

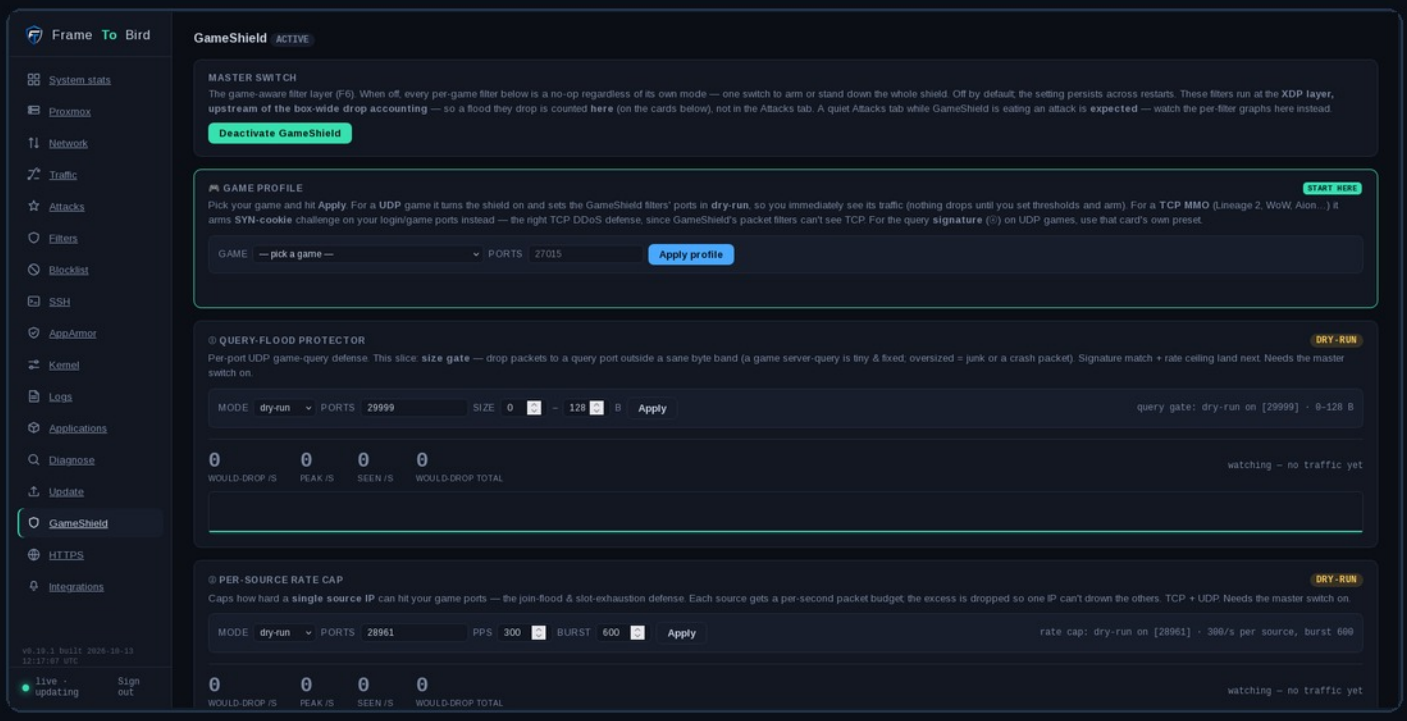
Firewall editing with rollback



The Filters view models rules and shows the live firewall reference. Apply operations use a timed confirmation workflow so an unconfirmed change can be reverted.

IN THE PANEL

Game-aware filtering



GameShield provides a separate switch and per-game or per-port controls, including query-size and per-source rate policies. The screen explains that these controls are staged in dry-run and require deliberate arming.

Reported test figures & built-in guardrails

Reported controlled-test figures

The companion overview PDFs supplied for this update report a controlled test against a live production game server. They state that the XDP data plane remained attached and the server stayed online during the event.

2.76M
packets / second peak

~8.6 Gbit/s
inbound traffic peak

295,000
unique attacking sources

HOW TO READ THESE NUMBERS

These are reported results for a particular test environment, not an independently verified benchmark or a guaranteed capacity. The supplied PDFs do not provide raw telemetry or key reproduction details such as hardware/NIC, packet mix, test duration, attack-generation method, and per-layer counters. Validate the exact release on representative hardware before using these figures for sizing or procurement.

Concrete implementation details

Automatic mitigation bounds

Auto-mitigation has 3 escalation rungs and can manage up to 4 ports at once. It waits at least 60 seconds between escalation steps and reverts after 300 seconds of calm. The pre-engagement baseline is persisted.

Game profiles

21 named presets plus custom. UDP examples: Source, Rust, ARK, Valheim, FiveM, Minecraft and Quake III. TCP examples: Lineage 2, WoW, Aion, Metin2 and Terraria. Review preset ports for your server.

Layer-specific, operator-controlled protections

The eBPF program contains controls for ingress sanity, TCP flag and fragment checks, UDP reflection/closed-port policy, reputation and CIDR blocklists, scan auto-bans, per-source and SYN limits, per-port UDP/ACK ceilings, SYN-cookie challenges, web-service IP/port policy, and GameShield query-size, rate, payload-signature, and repeat-packet filters. Observe/dry-run modes are available for many controls; arming remains an operator decision.

Scope, requirements & next steps

Typical deployment

A Linux server or gateway runs the FrameToBird service and panel. The service reads operator configuration from TOML and stores its mutable state, logs, and backups in a data directory. The panel is served from the binary with its static UI embedded.

Requirements depend on the selected features

Panel	Rust-built service; supports Linux, macOS, and Windows for panel/control-plane development. Host-management operations are platform-specific.
XDP filtering	Linux kernel and driver support; an XDP-enabled build and configured interface are required. eBPF toolchain requirements apply to the XDP build.
Firewall control	Linux tools and privileges for the selected backend; systemd is used by rollback and service packaging flows.
Optional data	GeoIP annotations need an operator-supplied MaxMind database. Threat feeds, SMTP, chat/webhook channels, and OVH API require operator configuration.

HONEST LIMITS

XDP can reduce host-side packet-processing cost, but it cannot stop traffic from consuming a saturated upstream link. Upstream scrubbing or provider mitigation may still be needed. The screenshots identify a v0.19.1 build, while this checkout declares v0.1.0; confirm the exact release, feature set, kernel compatibility, and operational behavior during technical evaluation.

A sensible evaluation

Agree on a test host and recovery path; validate the XDP interface and driver mode; observe dry-run counters against normal workloads; test the exact policies and services you plan to protect; then review rollback, logs, and support expectations.