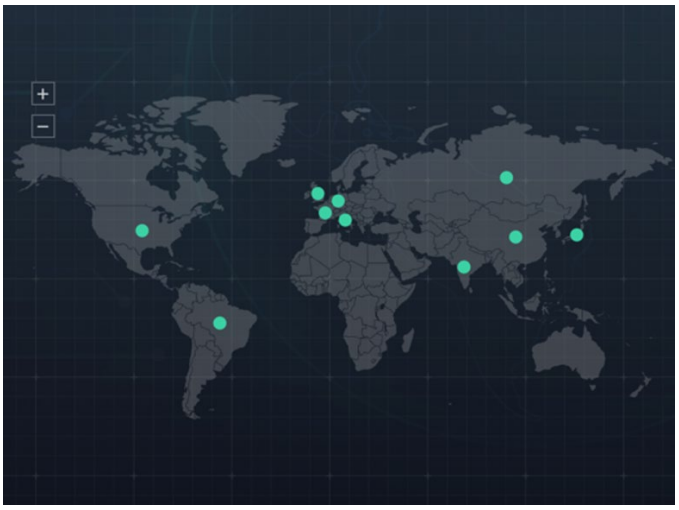
ARA SYSTEM

BOT SIMULATED SOFTWARE (PAL-BSS) DATASHEET



The PALAMYS system (Controlled/Customizable Strategic DDoS) is a purpose-built DDoS simulation and penetration-testing platform designed for special units and cyber-range operators who need the most realistic, operationally relevant attack emulation. Developed from real-world cyber-warfare observations and modern DDoS trends, PALAMYS combines powerful, field-proven attack techniques with an operator-friendly control surface so teams can design, schedule, execute and analyze realistic denial-of-service scenarios without excessive operational overhead.

PALAMYS replicates modern adversary behavior across volumetric, protocol and application layers while giving defenders the observability and repeatability required for high-value training and readiness exercises. The system was architected to reflect cyber threat evolution — from amplified reflection attacks and IoT/VM-based botnets to sophisticated HTTP/2 and browser-emulation application floods — and to let blue teams test detection, mitigation and incident-response playbooks under controlled but realistic pressure.

Key capabilities:

- High-impact volumetric & reflection attacks: generate amplified/reflective traffic against web apps, DNS, NTP, COAP, WSD and other vulnerable services to emulate real amplification campaigns.
- Autonomous discovery & augmentation: continuously scan the Internet to discover additional reflector/attack nodes and automatically incorporate them into campaigns.
- Botnet lifecycle management: add, remove and orchestrate attack nodes (bot/VM/reflector fleets) with fine-grained control over behavior, persistence and geographic distribution.
- Scheduled, automated campaigns: create reusable scenarios and run full attack playbooks on demand or on a calendar to support exercises and long-term readiness validation.
- Target reconnaissance & adaptive strategy: automated target profiling (application fingerprinting, API discovery, CDN/WAF detection, origin IP tracing) and automated selection of the most effective attack vectors.
- Visual attack path mapping: interactive visualizations of attack flows, vectors and impact metrics so operators can explain lessons learned to technical and non-technical stakeholders.

BOT Simulated Software (PAL-BSS)

The Attack Layer provides the high-throughput execution fabric for PALAMYS's DDoS simulation capabilities. It combines purpose-provisioned attack servers and scalable VM-Base BOT software to generate realistic, multi-protocol traffic at production-like scale. Designed for use in controlled cyber-range exercises, the Attack Layer emphasizes predictable performance, seamless horizontal scaling, and operator safety controls to ensure repeatable, auditable stress tests of target infrastructure.

Table 1: VM-Base BOT — Bot Simulated Software specification

Feature	Specification / Capability
Browser-like traffic emulation	Generates realistic, application-layer traffics to targets using a headless/virtual browser engine that reproduces common browser behaviors (JS execution, cookie handling, redirections, headers).
Minimum throughput	Capable of producing **at least 10,000 HTTP requests per second** per attack server instance under standard test profiles.
HTTP protocol support	Full compatibility with HTTP/1.x and HTTP/2 stacks (including connection reuse, multiplexing where supported).
Unlimited horizontal scaling	Capable of scaling the number of threads (equivalent to a VM-based bot) unlimitedly by expanding the server infrastructure (When adding a new service server to the VM-Base BOT server network, the system automatically rebalances threads and incorporates the new server's resources into the existing network, increasing the system's total attack threads).
Browser-behavior simulation to evade WAFs	Emulates browser behavior, simulates realistic access patterns to emulate real user sessions and to help bypass web firewall inspection mechanisms. Simulation patterns may covers user-agents, headers, and other behaviors.
Dynamic thread management	Runtime scaling of simulated clients/threads per server with adaptive ramp-up and ramp-down policies for controlled stress profiles.
Safety & governance controls	Built-in rate-limits, target scoping, approval gates, dry-run mode and emergency stop to prevent accidental or out-of-scope impacts.
Observability	Per-instance telemetry: requests/sec, concurrent connections, error rates, latency histograms, and per-target contribution metrics.
Integration	Native integration with Controller, Connector Management and Command Distribution services for orchestration, scheduling and command delivery.
Licensing	Perpetual or year-base; unlimited/limited installation count; update & maintenance support service.

Performance & scaling behavior

- **Per-server baseline:** Each provisioned attack server meeting the host specifications is validated to support the VM-Base BOT workload at a minimum of 10,000 requests/second under defined test profiles.
- **Aggregate scaling:** Total system throughput grows linearly with the addition of attack servers. When a new host is registered, orchestration automatically redistributes simulated clients and increases global thread capacity. There is no artificial software cap on flows — practical limits are determined by available hardware, network egress, and target-side constraints.
- **Protocol fidelity:** HTTP/2 multiplexing and connection behaviors are preserved where supported by the target; when multiplexing is unavailable the emulator falls back to appropriate HTTP/1.x behaviors.

Deployment guidance

- Place attack servers on networks with stable paths to intended target infrastructures to produce representative traffic patterns; avoid oversubscription of egress links.
- Use geographic distribution to emulate realistic adversary footprints and validate geo-based mitigation controls (CDN/WAF behaviors).
- Plan capacity by sizing both network egress and server count: for example, to reach 1,000,000 req/s at the validated per-server baseline, provision ~100 validated attack servers (plus overhead for protocol and TLS sessions). Actual numbers depend on request size, TLS handshake rate and target response characteristics.