ARA SYSTEM

COMMAND DISTRIBUTION SOFTWARE (PAL-CDS) 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.

Command Distribution Software (PAL-CDS)

The Command Distribution Software (CDS) is the orchestration engine that translates attack plans into distributed control actions across the botnet and emulation infrastructure. It guarantees commands are delivered reliably, safely and efficiently to the right groups of controlled hosts, adapting distribution strategies to target characteristics, geography and available resources. The CDS is built for repeatable, auditable operations in training and assessment environments where precise control and accountability are mandatory.

Core functionality

- **Intelligent command routing**

Command distribution feature: Based on analysis of attack targets and scenarios, the system automatically routing commands to botnet regions, ensuring appropriate use of resources using basic information.

- **Target-aware dispatching**

Dispatch decisions take into account application and server fingerprinting, suspicious origin IP addresses behind reverse proxies/WAFs (if any), and the country/region where the target infrastructure resides. This enables more effective vector selection and localized resource allocation.

- **Automated parameter suggestion**

For each campaign, the CDS can propose tuned attack parameters (target URL/IP, ports, methods/payload families, and initial throughput settings) derived from the target profile and past campaign performance, while allowing operators to edit suggestions before execution.

- **Geo-distributed resource allocation**

The service can split command delivery and throttle resource utilization by geographic location (country, region, ASN) so attack traffic mirrors realistic distribution and avoids overwhelming single network chokepoints.

- **Command types supported**

Suite of operational commands including start/stop/pause, protocol-specific payloads, and controlled administrative actions such as remote file-deletion commands for authorized exercise scenarios.

- **Automated execution & scheduling**

Combine the distribution engine with the campaign scheduler to execute staged or chained command sequences, including phased escalation and rollback steps, with dependency awareness across connector pools.

- **Safety controls & approval workflows**

Multi-step approval flows, dry-run simulation mode, scoped command policies (to limit allowable target types, vectors and time windows), and an emergency global kill-switch are enforced to prevent unintended collateral effects.

- **Reliability & delivery assurance**

Uses reliable queuing, acknowledgements and retries for command delivery; supports transactional semantics for grouped operations to maintain campaign consistency.

- **Audit & traceability**

Every distributed command and resulting node acknowledgement is logged with timestamps, originator identity, and execution outcome for full traceability and post-exercise analysis.

- **Governance & policy engine**

Configurable policies determine permitted commands per role, per environment, and per campaign type. Rate-

limiting and geographic caps are enforced at distribution time.

Table 1: Licensing & deployment terms

Item	Specification
Software license	Perpetual or year-base
Installation license	Unlimited/limited installations/hosts under the license terms
Update & maintenance	Maintenance service for updates, security patches and support

Operational notes (for procurement & operators)

- The CDS must be co-located or have a low-latency path to Connector Management and Command Distribution worker pools for timely command propagation.
- Configure geo-allocation rules before running; the distribution engine includes pre-checks to validate target scope against policy.
- Destructive commands (e.g., file deletion) are available only within explicitly permitted exercises and require elevated approvals and explicit operator confirmations.
- The system provides simulation (dry-run) capability so distribution behavior and resource allocation can be validated without enacting live effects.