Polarity – Supporting Closed Network Deployments

What is Polarity?

Knowledge and data is spread across disparate systems. Polarity fuses them together in one unified view.

Polarity provides a new way for IT and Security Professionals to capture and deliver critical intelligence to the right team members only when it is relevant to what they are working on. Polarity drives analysts to make better and faster decisions, increasing productivity, and reducing the risk of incidents going undetected. Polarity works by analyzing the content actively using search automation or passively supplied by users on-demand and notifying the user about material or directly related intelligence, helping to ensure that analysts never miss critical intelligence when they need it most.

Polarity's Components

Polarity follows a typical client-server model where the Polarity Clients (Windows, macOS, LInux) run on the analyst's workstation and connect to a Polarity Server running CentOS 7 or RedHat Enterprise Linux 7 or higher. The server is self-contained, which allows Polarity's full capabilities to run in closed and airgapped environments.

Detailed components of the Polarity architecture are included within the latest version of the Polarity Administrative Guide. Here is a summary of the Polarity Platform components:

Polarity-Server:

Polarity-Server is the backbone of the Polarity Platform. It hosts the Polarity REST API, stores Polarity Annotation data and Integration code. Polarity administrators can interact with the server via the administration portal, or the command line to install integrations, perform maintenance, or troubleshoot issues. Core components include:

- **PostgreSQL Database**: Stores and enables "recall" from Polarity entities and annotations.
- **REST API Server**: Runs and manages Polarity Integrations, Channels, Authentication, and Annotating.
- **Nginx**: Proxies requests to and from the Polarity REST API.
- **Cache(s):** Enables session management and improves integration experience.

Polarity-Client:

Polarity-Client is the application that runs on the users' desktop and allows for automated, on-demand and focused searches and actions. It is the application where you can control the settings for what Polarity recognizes on your screen as well as explore what data is in Polarity-Server and turn on and off Integrations. Polarity-Client is how you also control the Overlay-Window for Heads Up Display (HUD) searches and determine which entities and annotations you are overlaying. Core components include:

- **Configuration Window**: Where most user options are configured, and users can explore data within Polarity.

- **Overlay Window**: Where users receive information about entities recognized on the user's screen.

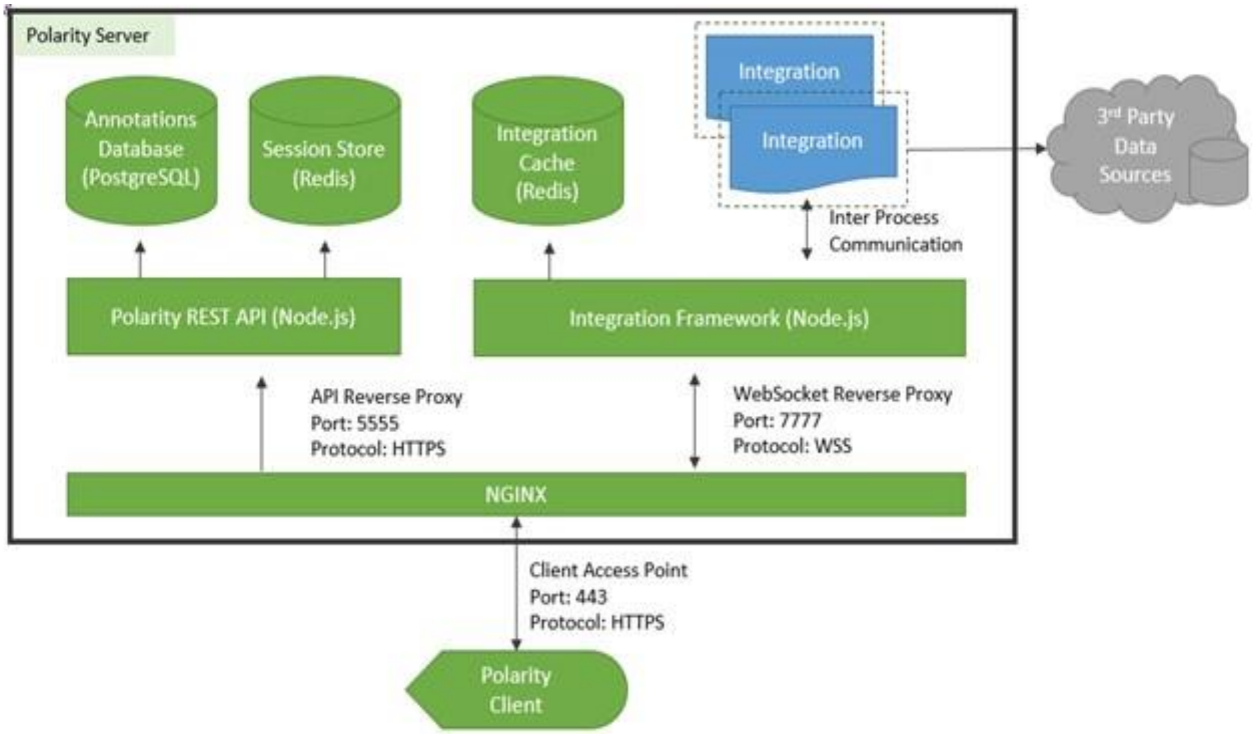
- **Highlight**: Enables screen drawing and awareness via in-line overlays.

- **Focus:** Enables the tactical designation of screen areas for data extraction with Polarity's proprietary Optical Character Recognition (OCR) technology.

- **Apply-Annotation Window**: Allows the user to quickly apply annotations to entities.

Common Polarity Deployment Model:

Most commonly, Polarity-Client(s) and Polarity-Server components are not co-located to a single host. The most common deployment is depicted below:



The common Polarity deployment assumes that users of the Polarity-Client are connected to a network that has access to the centralized Polarity-Server as well as access to integrated internal and external information resources.

Limitations of Common Architecture on Closed Networks:

Analysts perform investigations within environments that are purposefully isolated from otherwise interconnected systems for several reasons:

- Absence of Connectivity
- System Integrity
- Attribution
- Policy / Mandate

Where analysis is being performed under these conditions, and analyst computers/local Polarity-Clients cannot interact with the centralized Polarity-Server, "local" Polarity-Servers can enable analyst teams outside of the Common Polarity Deployment.

Polarity's Closed Network Deployments

In scenarios where a Polarity-Client(s) cannot connect to a centralized Polarity-Server, there are at least two secondary models that can be pursued. These models are ideal for enabling single analysts conducting analysis on an isolated system, or for enabling multiple analysts on a closed network.

Single Analyst (Isolated) Model:

In a single analyst model, it is assumed that one analyst is conducting analysis on a single physical host machine. It is also assumed that their workflow, although temporarily less dependent on participation of peer analysts, would benefit from either the collective annotations of the team, or information from integrated data sources.

Initial Installation

- Establish a local Virtual Machine capability: Install software that allows for the hosting of a virtualized server on the analyst machine.
  - Deploy the Polarity OVA: Instruct the virtualization utility to unpack and install the local Polarity-Server.
  - Allocate Resources: Assign resources (e.g. CPU, RAM) to the Polarity-Server.
  - Start Polarity Server: Select "Polarity-Server" (or alias created upon deployment) from the Virtual Machine Library and "run".
- Configure Polarity Server: (See Administrative Guide)
  - Install Desired Integrations: Retrieve integration packages from the Polarity integration repository.
- Install Polarity Client: The Polarity-Client should be installed directly on the host OS, or on a separate guest OS running on the same host.
  - Configure the Polarity Client to leverage the local Polarity-Server as its Primary Data Source and authenticate to Polarity. (See Polarity User Guide)
  - Configure Integrations: Configure Polarity Integrations to interact with designated data sources. For example, configure URLs, API keys, credentials, etc.
- Polarity Annotation Import: If desired, the user of Polarity in this Isolated mode may desire to enable overlays of historical team annotations. Should this be desired, the steps in *Polarity Annotation Importing and Exporting* of this guideline should be followed.

Multi-Analyst Model:

In this model, Polarity is either deployed on a physical server hosted within a closed network, or as a VM hosted on a computer (laptop/server) complimenting the suite of technology (the "kit") enabling the analyst team.

When supporting multiple analysts, interconnectivity between (1) team member computers running PolarityClient and (2) the Polarity-Server is paramount to achieve the objectives of the team.

The steps to enable a team with Polarity in a Multi-Analyst model are not all that different from a more traditional deployment, or otherwise a Common Polarity deployment. However, some non-standard conditions will need to be accounted for.

Primarily, the team will need to account for operations in absence of interconnectivity to the Central PolarityServer as well as an Internet connection. Polarity does not require the Internet in order to support operations. However, in the absence of a connection to the centralized Polarity-Server:

- Polarity annotations will be committed to a non-centralized instance of Polarity-Server, limiting system redundancy to only those copies of the Polarity Database deployed and replicated within the "Kit."
- Integrated data sources will need to be available within the closed network environment.
- The complete annotations of the team, operating outside of the closed network will not be available to the airgapped deployment until the contents of the Local Polarity-Server and the centralized Polarity-Server can be merged. (*See Polarity Annotation Importing and Exporting – page 6*).

Initial installation and configuration of Polarity should be performed via traditional deployment methods, however, the team responsible for initial installation of Polarity within closed network environments that are using the RPM-based installation process can contact Polarity for offline installation instructions at support@polarity.io.

Polarity has and will continue to support customers deploying to closed networks.

Performance Considerations

The following table highlights the recommended resources that should be allocated to a virtual guest functioning as the Polarity-Server.

Polarity User(s)	Integrations	Cores (vCPU)	RAM
1	5	1	2 GB
2-10	10	2	2 GB
11-20	15	3	3 GB

At minimum, 4 logical Cores must be allocated to the host/guest machine running Polarity-Client.

Special Security Considerations

When attribution, or the analysts'/computers' affiliation with an organization is of chief concern, the following steps should be considered to preserve anonymity.

- **Integration Considerations:**
  - ○ Ensure the following do not disclose true user identity or business entity.
    - Configured User names
    - Configured API Keys
    - Configured URLs/Host Names ○ Ensure interconnected data sources are not attributable to the organization
    - No logos are displayed on web applications ○ If all traffic is not already routed through an anonymous proxy, a proxy can be configured on a per integration basis (e.g. proxy Virus Total lookups)

- **Client Installation**:
  - ○ Ensure that any quick links configured within Polarity-Client instance(s) do not disclose internal information (e.g. URLs, IPs), true user identity or business entity
  - ○ Ensure that email addresses are not configured

- **Server Installation:** Upon server installation, do not create user accounts and/or groups that are attributable to a true user or business entity ○ Disable SMTP notifications

- **Cryptographic Considerations:**
  - ○ Do not create an SSL Certificate with attributable information
- Certificate Authority
- Certificate Name (Self-Signed Cert) ○ Ensure integrations are configured to **only** perform look-ups via encrypted channels

- **License key:** Ensure that the license file allocated to the non-standard deployment of Polarity is not attributable to the business entity.

- **Operational Security:** The following should be observed when leveraging Polarity under circumstances where avoidance of attribution is paramount.
  - ○ When investigating targeted entities, do not leverage integrations that create a record of its query
  - ○ Do not create attributable channels / entities / annotations / comments ○ Do not send Polarity error reports, and report issues out of band
  - ○ Interconnected data-sources / servers should not be registered to the business entity or the true name of a business entity employee
  - ○ Payments for interconnected servers / data sources are not made from credit cards bearing the company name

Polarity Annotation Importing and Exporting

When working with Polarity in a closed network deployment it will often be desirable to move annotations collected from the closed network to a centralized Polarity-Server. Polarity supports this movement of annotations through an export and import process.

A Polarity Server admin on the closed network deployment can export specific channels from the server to a compressed file. The file will contain annotations and their associated entities. This file can then be moved via “sneakernet” (i.e., via physical media such a USB flash drive), from the closed network deployment to the centralized server.

Once the file has been loaded on the centralized server, the import process can be run to import the exported data into a specified channel on the centralized server as well as under a specified user account on the centralized sever.

This process can be reversed to bring annotations from the centralized server to the closed network deployment thereby enabling analysts to tap into the organization’s historical knowledge even while working on a closed network.

About Polarity

Polarity fuses knowledge and data together into one unified view, enabling information delivery, automating knowledge transfer across teams, and allowing leaders to understand which of their data sources deliver value. Polarity up-levels teams in security operations at Fortune 100 companies, incident response for health care organizations, threat intelligence at financial services companies, hunt operations for the US federal government, and more. Learn more at www.polarity.io