



ccBay Pilot Architecture

Pilot Design

- Highly modular architecture utilizing standard building blocks.
- Focus on a simple and extensible lightweight design.
- Utilize existing libraries, utilities, modules and standards instead of existing systems.
 - We have looked at many, but there is no existing software that does all we need.
- Use Python as a full-featured development language.
 - Great language for development of pilot and beyond.
 - Widely included with Linux distributions.
 - Works well in the Windows environment.

Pilot Architecture

- Normalization agents to convert raw event and logging data to ccBay XML documents.
 - Each event is a file, more efficient approach needed beyond pilot.
 - For pilot, normalization agents
 - Linux (RH9 and Fedora C2) - kernel events (klogd), user events (slogd)
 - Windows (WIN2K and WIN2003 server) – application, system and security logs, installer (WIE)
 - Snort Events
 - NetFlow Events (Version 5)

Pilot Architecture

- Forwarding agents move XML documents between storage agents.
 - Transfer of XML files to storage agent using SCP for pilot.
 - Simple design for pilot, may not be production approach.

Pilot Architecture *(continued)*

- Storage agents to archive to files or populate database with XML data.
 - Storage agent to populate MySQL database for pilot.
- Transformation agents to anonymize, aggregate and remove events.
 - For pilot, basic anonymization by removing IP addresses.
- Applications to analyze event and logging data.
 - For pilot, basic forensic Web site and CLI tools.

Pilot Architecture *(continued)*

- Management System to manage and control the distributed system.
 - Diagnosis and ccBay event generation for the system itself.
 - For the pilot, management events will be generated for certain error conditions as well as startup/shutdown notifications.
 - Web, GUI and CLI applications for controlling and configuring the overall ccBay network of hosts.
 - Manually configure the ccBay hosts for the pilot.