XKEYSCORE
System Administration

December 2012

Introduction

Lesson Objectives

- Introduction to XKEYSCORE
- Purpose and Capabilities
- Data Flow
  - What is a Cluster?
  - XKEYSCORE Databases

Introduction

- XKEYSCORE performs filtering and selection to enable analysts to quickly find information they need based on what they already know.

- XKEYSCORE also performs SIGDEV functions such as target development to allow analysts to discover new sources of information.

Introduction

- XKEYSCORE processes data at field sites, where it is collected, and allows analysts from all over the world to query it.
- At field sites, the XKEYSCORE software can run in clusters of few or many servers, giving it the ability to scale in both processing power and storage.
- All processing is plugin or fingerprint based, which allows new capabilities to be quickly deployed to support operational needs.

Purpose and Capabilities

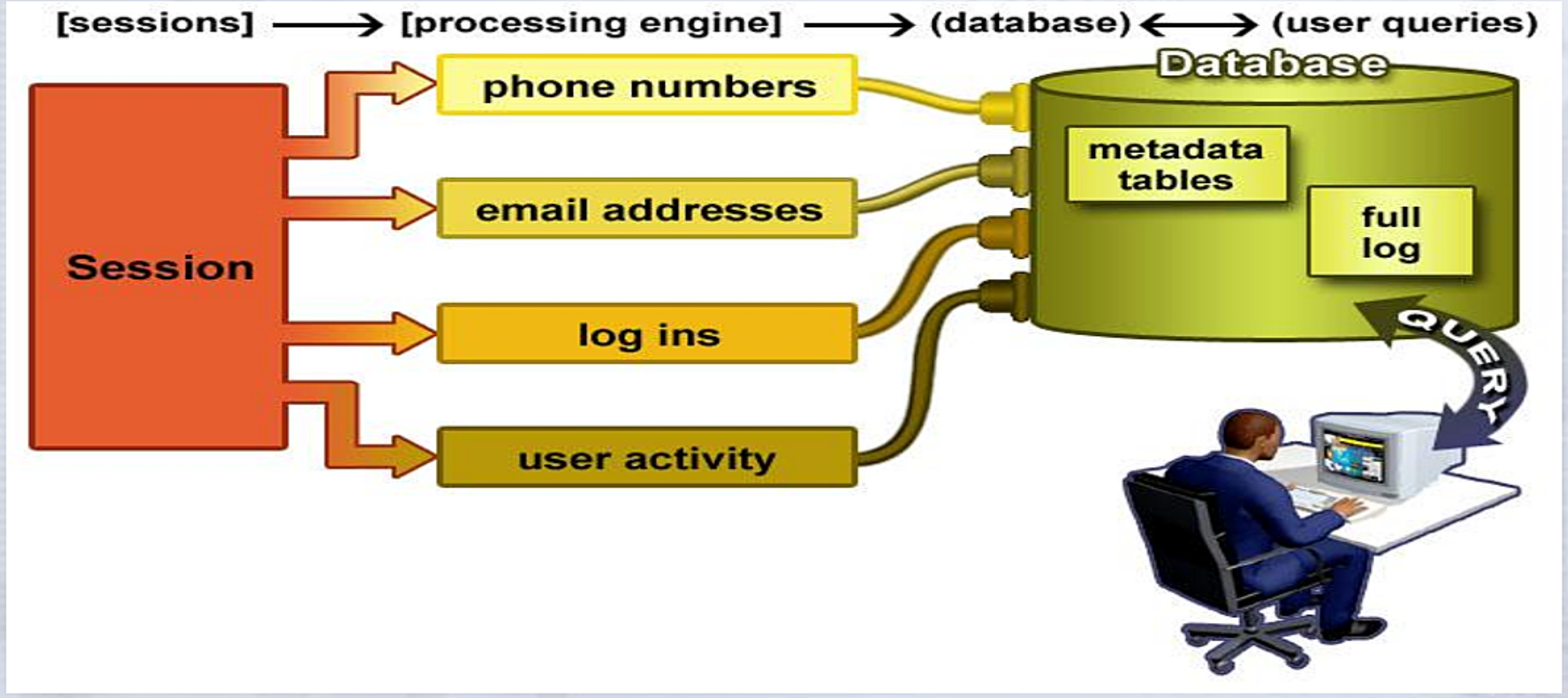
- XKEYSCORE is a Computer to Computer (C2C) exploitation system.
- It is a fully distributed processing and query system.
- XKEYSCORE can run on multiple servers.
- Plugin and fingerprint architecture allows new capabilities to be quickly deployed.

Hardware

- XKEYSCORE is typically installed with Red Hat AS5u8 operating system. The suggested disk set up is:
  - Set up separate partitions for / (root), */var,* /tmp, and */export/data*
- XKEYSCORE clusters can be composed of three different functionalities, which are:
  - One host acts as the web server/user interface, etc…
  - Another host normally runs as the real-time processing unit
  - Other host acts as the search or query system.
- Hybrid system can perform multiple roles on one server, which enables efficient registration.
  - process_data_parent
  - 1 query_proc

Data Flow (High-level)

- The backend is where the raw data for XKEYSCORE is processed; that is, we receive information from our sources (e.g. WEALTHYCLUSTER2), process it, and store it into a database.



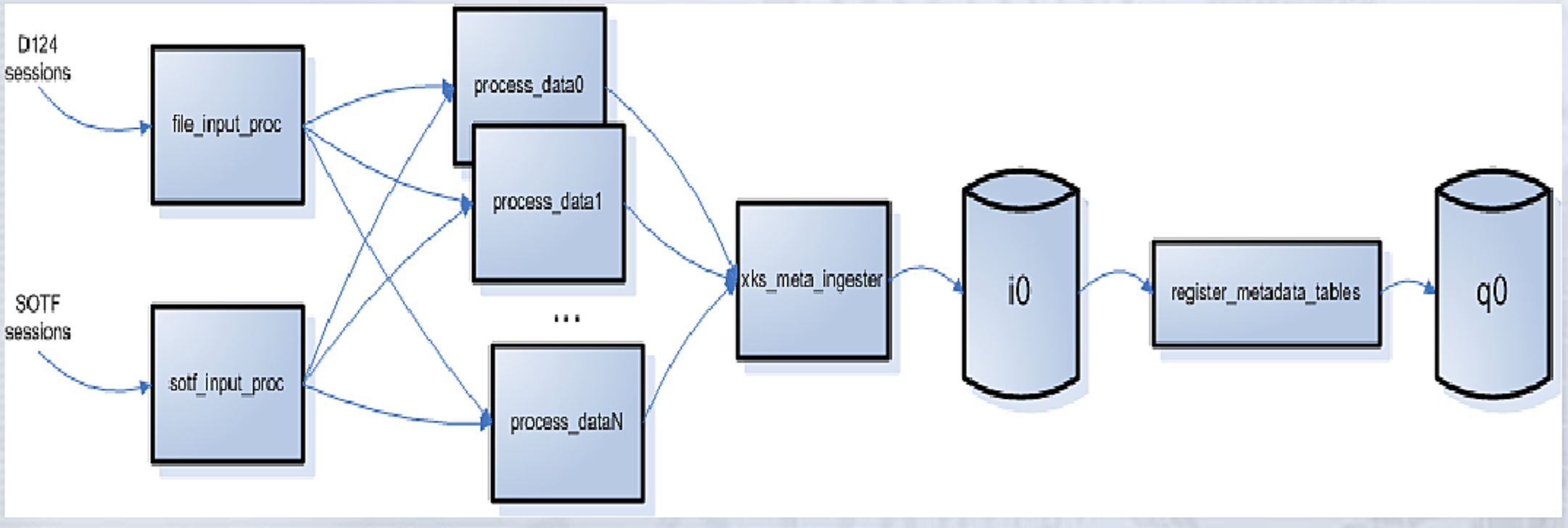
Data Flow - XKS Cluster

- A cluster is comprised of one master server and one or more slaves.

- All slaves in a cluster have their own copy of configurations (/opt/xkeyscore/config) files via the **xks rsync push_config** cronjob.

Data Flow - Databases

- There are two types of databases on an XKEYSCORE system: insert (i0) and query (q0)



NOTE: sotf_input_proc is now called, sotf_dist
process_dataN's are now called, process_data_parent

Data Flow - Databases

- file_input_proc and sotf_dist take in sessions from the front-end and load balances them across multiple process_data_parent's.
- process_data_parent is responsible for processing sessions and extracting metadata
- xks_meta_ingester takes the metadata from the process_data_parent's and writes it to the insert database, i0
- register_metadata_tables takes completed insert tables, indexes them, and moves them to the query database, q0

XKEYSCORE Architecture

Lesson Objectives

- Operating System Services
  - HTTPD
  - MYSQL
  - NFS
  - AUTOFS
- Mount Points
  - /xks_data
- Directory Structure

Operating System Services

- XKEYSCORE is typically installed on servers running Red Hat 5u8 operating system.

- This section discusses common operating system services used during XKEYSCORE operation.

HTTPD

- The **http** daemon is needed for the web-based GUI, viewing content, and is required on all servers.

- The master server is the web server and the slaves retrieve content through HTTPS.

MySQL Daemon

- The **mysql** daemon is a SQL-based database server for processing, querying, and is needed for the XKEYSCORE GUI.

- It is required on all servers for administration, processing, and querying metadata in databases.

NFS

- Mounting a directory uses the NFS service.
- NFS allows file systems that physically reside on one computer to be shared by other computers on the network.
- The machine with the hardware containing the directory must allow the hardware to be made available to other machines.
- Required on all computers for clustering.

NFS

- /etc/exports
  - /export/data/xkeyscore master(rw) slave(rw)
  - /opt/xkeyscore/config/loadserver *(rw)

AUTOFS

- Computers requiring shared access to the /export/data/xkeyscore directory must be told where to find the directory.
  - This is accomplished via automounting.
- The **autofs** daemon listens for computers trying to connect to the directories, or mounts, that it is responsible for.
- The mounts are dropped after a time out, but **autofs** remounts the drive when drives need to be accessed.

Automounting

- For a clustered XKEYSCORE, automounts must be set up on all of the computers in the cluster.
- *auto.master and auto.data* files in the */etc* directory must be edited or created.
- When finished, the mounted directories on the remote machines can be accessed.
- The *oper* account should have full read/write permissions on all shared drives.

Mount Points

- **auto.master** – designates mount points on the local computer and the directory to mount on the remote server.
  - Example:
    - /xks_data /etc/auto.data --timeout=60
- **auto.data** – enables all servers to see the /export/data/xkeyscore directory on other machines and locate databases, archived, data, and MAILORDER directory.
  - Example:
    - xks1 -rw,soft,intr,tcp xks1:/export/data/xkeyscore
    - xks2 –rw,soft,intr,tcp xks2:/export/data/xkeyscore

Directory Structure

- **/opt/xkeyscore/** – contains all of the XKEYSCORE software. Software includes the GUI, processing, scripts, and configurations.
  - **bashrc** – XKEYSCORE environment variables file.
  - **beacon/** - contains the beacon perl script (shm_beacon.pl) and a link to the beacon configuration file (shm_beacon.config).
  - **bin.shells/** and **bin.shells/sysadmin** - contains miscellaneous bash, python, and C shell scripts.
  - **build/** - contains libraries and plug-ins.
  - **install/** - contains installation scripts.

Directory Structure

- **/opt/xkeyscore/config/** – consists of sub-directories and each contain configuration files for building and running XKEYSCORE.
  - **crontab/** - contains the master and slave crontab file.
  - **dictionaries/** - contains the dictionary files for the filtering, selection, TRAFFICTHIEF, CADENCE, fist tables, and any other local dictionaries.
  - **misc/** - contains miscellaneous per-plug-in configuration files, (i.e. sotf_input_proc.xml).
  - **plugins/** – contains event handler configuration files for each of the plugins (default.xml).
  - **www/** - contains web configuration files and xscore.cfg.
  - **SERVICE/** - contains the config files for all the services needed by XKEYSCORE (httpd, php, mysqld, etc.)

Directory Structure

- **/opt/xkeyscore/www/** - contains the contents of the web front end.
  - **docs/** - contains documents viewable through the XKS GUI.
  - **html/** - contains web pages and scripts that are not on the secure server.
  - **secured/** - contains web pages and scripts that are on the secure server including:
    - **crons/** - location of cron job scripts
  - **src/** - contains source code for the XKS GUI.

Directory Structure

- **/export/data/xkeyscore/** - is used for both internal databases and metadata archive databases, input, output, and archiving of data.
  - **archives/** - (optional) destination for processed content
  - **inputs/** - (optional) used for file based input
  - **mysql/** - location of the MySQL database consisting of admin, insert, and query databases.
  - **outputs/** - (optional) contain the following sub-directories:
    - **mailorder/** - pickup point
    - **mailorder_working/** - file creation point before being moved to mailorder/

Directory Structure

- **/xks_data/** - logical mount point for all other XKEYSCORE (including itself) /export/data/xkeyscore.
  - **<hostname>/** - mount point for the *hostname's local directory /export/data/xkeyscore* (referenced by host name).
    - All servers must export their */export/data/xkeyscore* directory and mount this on the */<hostname>* directory for each hostname of each machine, including itself.

XKEYSCORE GUI

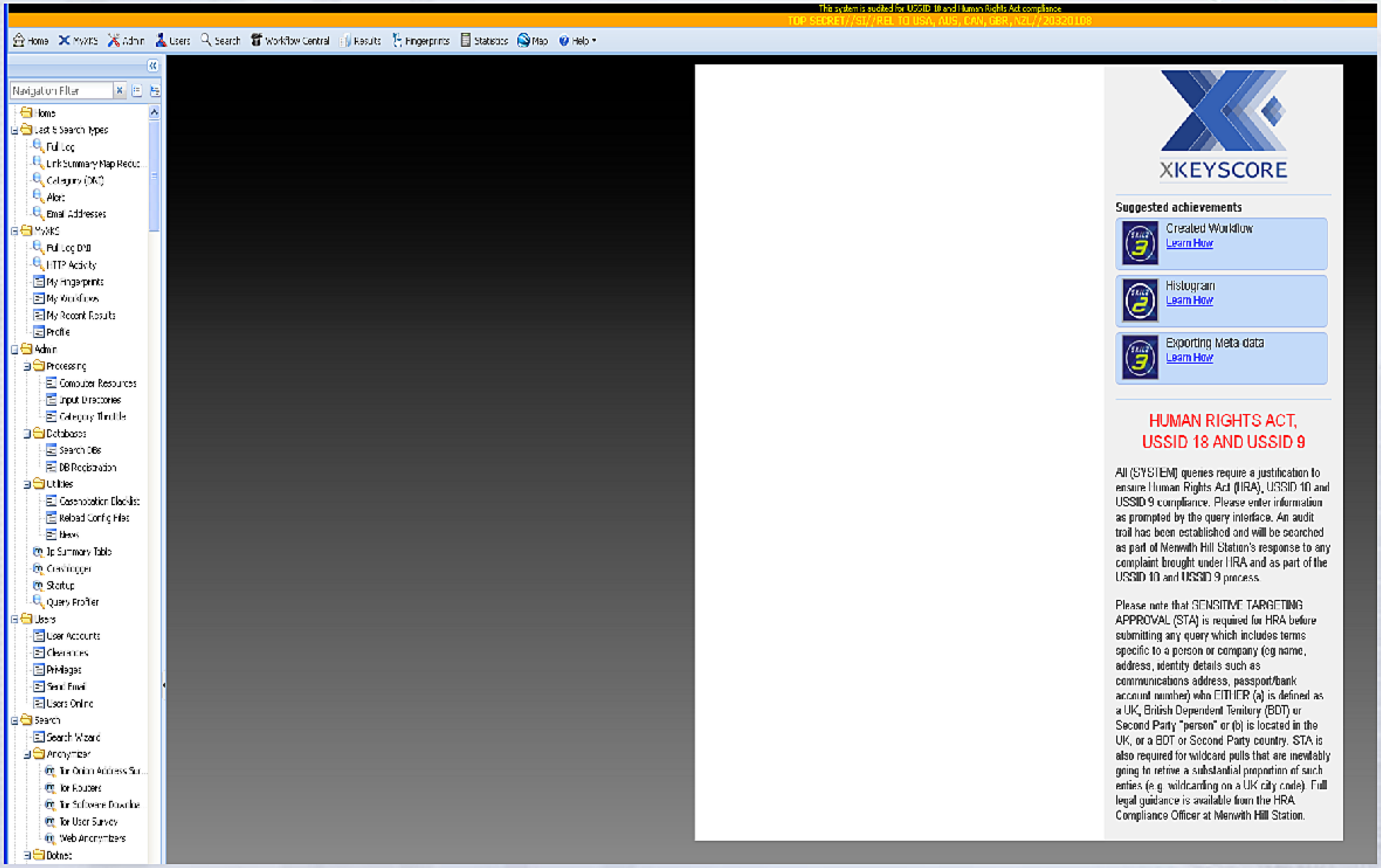
Lesson Objectives

- Accessing the GUI
  - Exiting a Session
- Main Menu Bar
- MyXKS
- Admin
  - Computer Resources Option
  - Start and Stop Processing
  - Run a Process Manually
- Users
- Search
- Workflow Central
- Results
- Fingerprints

Accessing the GUI

- In the address field of a web browser, type *https://<master hostname or IP address>.*
- PKI's or a UserID and password are required. After successfully launching a new session, the XKEYSCORE WELCOME window appears.
  - Note: Compatible web browsers for XKEYSCORE version 1.5 are:
    - Internet Explorer is not supported
    - Firefox/3.0.* and above

Accessing the GUI



Main Menu Options

- The main menu bar across the top of the window has menus that, when selected, each has additional options available in a drop down menu form.

Main Menu Options

OPTION	DESCRIPTION
Home	Returns to the main page.
MyXKS	Can edit user settings, disable/enable access to databases, edit a search form search setting, and restore default settings.
Admin	Computer resources, Input Directories, Category Throttle, Search DBs, and DB Registration settings.
Users	Contains User Accounts, Clearances, Privileges, Send Email, Users Online, My Auditees, My Audit Logs, and All Audit Logs.
Search	Provides different search query forms, such as email addresses, category, full log, and user activity.
Workflow Central	Request , modify, and view standing queries that will execute at a specified time or interval.
Results	Can search personal searches by date time, query type, query name, output table, and user.
Fingerprints	Fingerprint builder and viewer.
Map	Brings up Google Earth
Help	Help Documentation, XK Forum, Account Maintenance, and About XKEYSCORE

Admin Menu

SUB-MENU OPTION	DESCRIPTION
Computer Resources	Allows for process configuration and management.
Input Directories	Contains the configuration for file-based input directories.
Category Throttle	Edit CADENCE quota limits by category and/or fist table.
Search DBs	Configuration for query databases which are queried when a search is submitted.
DB Registration	Contains the mapping from insert databases to query database.
News	Add, modify, delete mandatory and home page News.

Computer Resources

- The **Processing->Computer Resources** option from the **ADMIN** menu allows control of the entire daemon-styled, or continuously running, processes for XKEYSCORE.
- Processes appears in a table following the convention:

<PROC_HOST><PROGRAM_NAME><PROGRAM_ARGUMENTS>
xkey01 process_data_parent

Computer Resources

Computer Resource Window Process Table

Home | MyXKS | Admin | Users | Search | Workflow Central | Results | Fingerprints | Statistics | Map | Help

Navigation Filter

- Processing
  - Computer Resources
  - Input Directories
  - Category Throttle
- Databases
  - Search DBs
  - DB Registration
- Utilities
  - Casenotation Blacklist
  - Reload Config Files
  - News
- Ip Summary Table
- Crashlogger
- Startup
- Query Profiler

Help

Color Key				
STOPPED	STOPPING	STARTING	RUNNING	WON'T START UP
STOPPED? APP LAUNCHER STOPPED	STOPPING? APP LAUNCHER STOPPED	STARTING? APP LAUNCHER STOPPED	RUNNING? APP LAUNCHER STOPPED	WON'T START UP? APP LAUNCHER STOPPED

Computer Resources

Help | Add | Actions | App Launcher is Running

Actions	Proc Host	Program Name	Program Arguments	Program PID	Commanded Status	Status	Datetime Started	Datetime Stopped
STOP	tlxksvr01	GUId		7607	RUN	RUN	2012-12-03 16:44:49.0	2012-12-03 16:44:49.0
STOP	tlxksvr01	query_proc		30965	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	check_mailorder_site.php		31019	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	xks_meta_ingester		31051	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	clickstreamService.sh		31053	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	query_dispatch		1311	RUN	RUN	2012-12-03 21:08:09.0	2012-12-03 21:08:07.0
STOP	tlxksvr01	file_input_proc		31104	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	xks_system_monitor		13986	RUN	RUN	2012-11-27 21:01:13.0	2012-11-27 21:01:13.0
STOP	tlxksvr01	sotftod124server		31108	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	tomcat.sh		31124	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	cadence_tasking_proc	--myfdi XYD --pddg IE --digraph X5	31136	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	xks_server_stats		31138	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	mailorder_proc	--copydir /export/data/xkeyscore/out...	31140	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:32:47.0
STOP	tlxksvr01	register_metadata_tables	--loglevel error	31143	RUN	RUN	2012-11-27 17:33:57.0	2012-11-27 17:33:50.0

Computer Resources

- The **xks_app_launcher** process runs on all servers from the inittab.
- It tells the computer which program to run by looking at its tasking host.
- */opt/xkeyscore/config/www/xscore.cfg*
  - The config file specifying the location of the tasking database.
- Processes can be stopped, started, edited, or deleted from the Computer Resources window.

Computer Resources

- **Add a new process** – click **Add**
- **Edit a process** – click **Stop** in the **ACTION** column, then click **Edit**.
- **Delete process** – click **Stop** in the **ACTION** column, then click **Delete**.
- **Stop the App Launcher** – disables the xks_app_launcher on every host.

Resources – Color Conventions

- Visual cues in the form of colors are used to help identify activities performed by XKEYSCORE and serve as status indicators for monitoring purposes.
  - **Red** – indicates processes have been stopped
  - **Green** – indicates processes are running
  - **Yellow** – indicates processes are starting
  - **Orange** – indicates processes are being stopped
  - **White** – indicates processes won't start
- Visual cues are also available in the COMMANDED STATUS and STATUS columns of the table.

Resources–Start/Stop Processing

- It may be necessary to stop or start processes for troubleshooting or for a graceful server restart.
- **Individual processes and programs** – Click **Stop** in the **ACTION** column. To start it, click **Run**.
- To stop all individual programs, select **ACTIONS->Start/Stop Resources**. Enter the program name in **PROGRAMS** field, then click **OK**.
- Can use '**xks proc**' actions and commands to do the same function

Resources–Start/Stop Processing

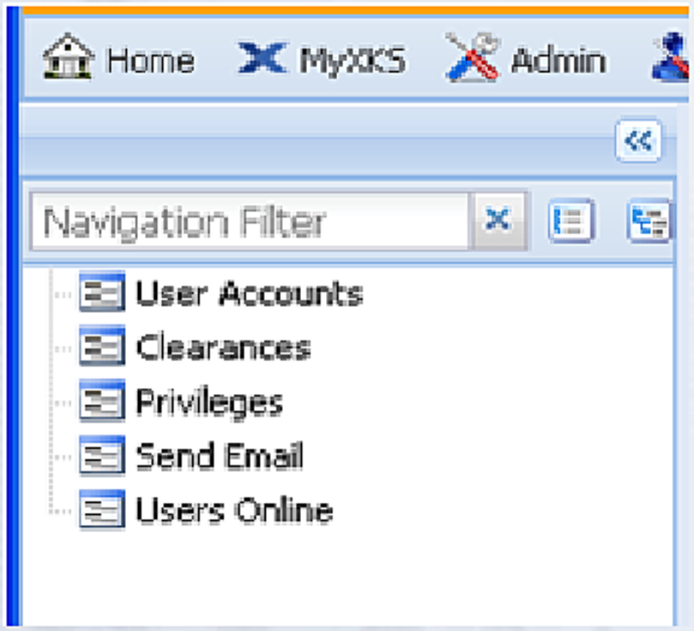
- **All Processing** - select **START/STOP Resources** from the **ACTIONS** drop-down menu, leave the **PROGRAMS** and **ON HOSTS** fields to their defaults, click **OK**.
- **Specifying programs or hosts** – select **STOP** or **START**, enter a wildcard expression such as * or **!** in the **PROGRAMS** or **HOSTS** field, and click **OK**.
  - Example: `process*`
- Alternatively, in a terminal window can run:
  - xks proc stop process*

Run a Process Manually

- It may be necessary to run a process manually for troubleshooting purposes. To run a process manually:
  1. Launch the GUI and log on as oper or admin.
  2. Click **ADMIN > Processing > Computer Resources**
  3. Click **Stop** in the **ACTION** column for the process.
  4. Open a terminal window and **ssh** to the host running the process, as the user `oper`.
  5. Type `ps -ef | grep <process name>` to verify that the process is stopped.
  6. Type `<program name><program argument>`
- Example:

```
query_proc <program arguments, if any>
   --loglevel debug
```

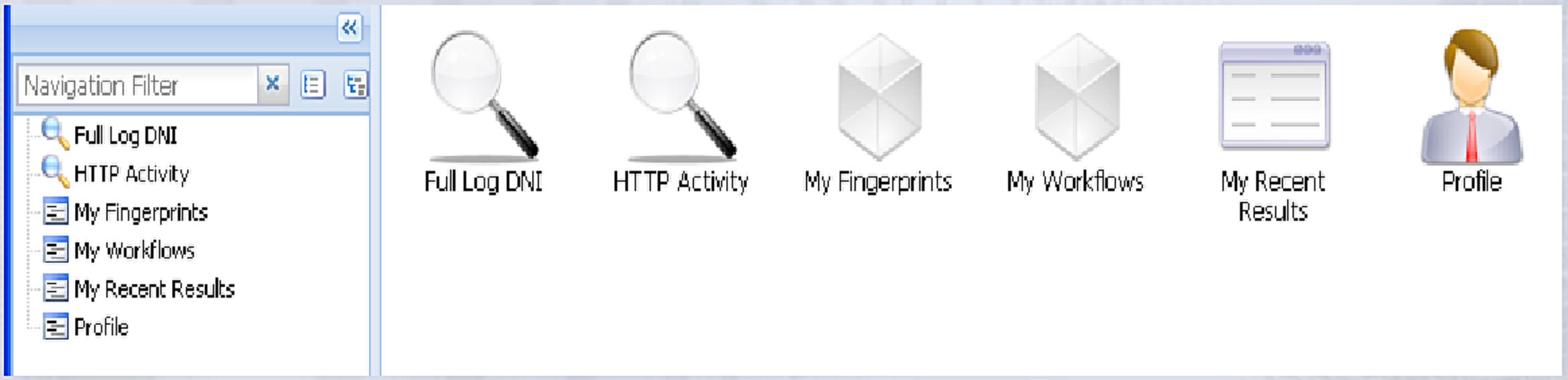
Users Menu



- This menu is only accessible to users with system administration privileges.
- An SA can add/modify user accounts, add groups, clearance levels, privileges, and email users from this menu.

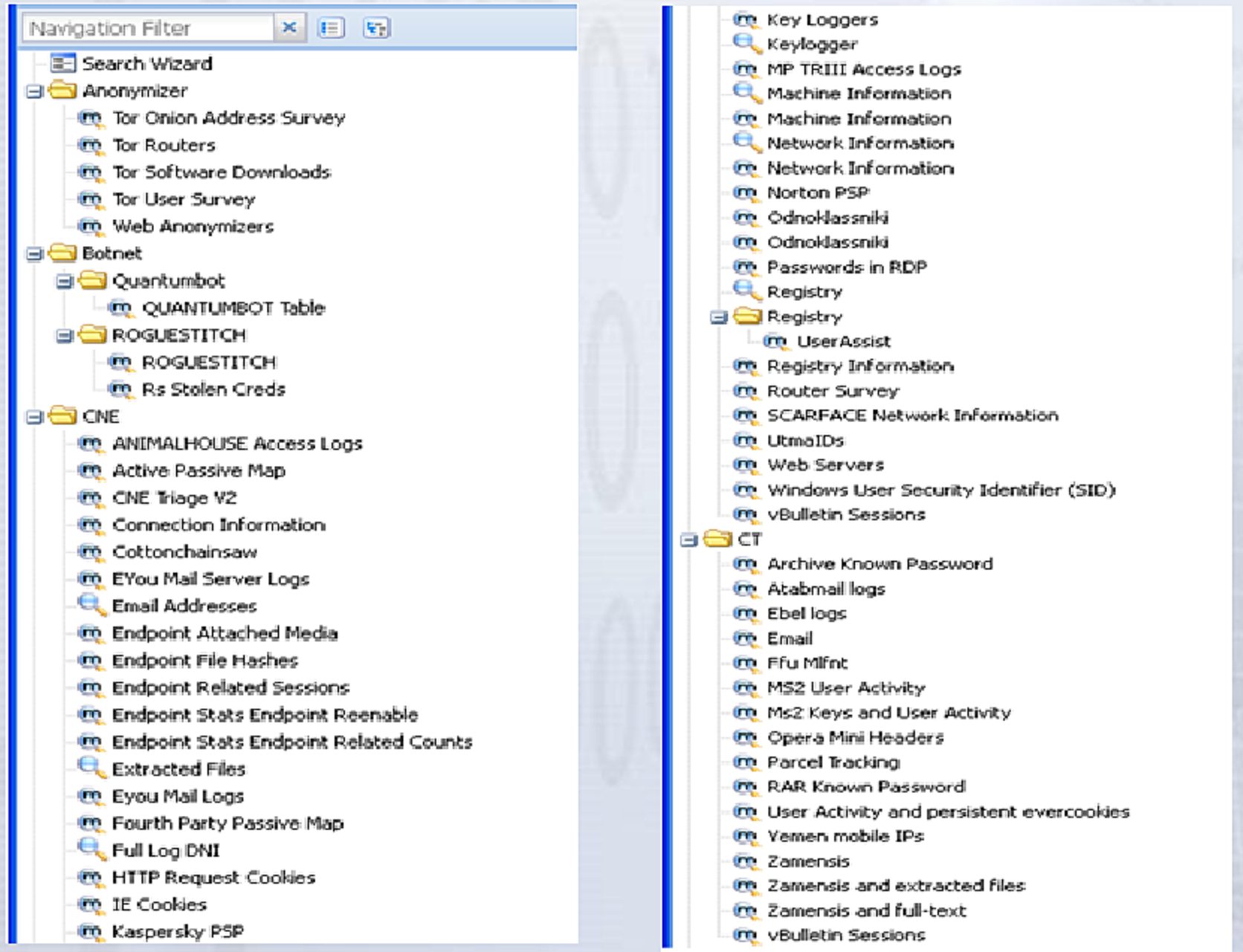
MyXKS

- From the main menu bar, click **MyXKS** to view your profile, accesses, privileges, auditors, settings, fingerprints, workflows, and recent results.
- Right click on any search form name to add a shortcut for that search form.



Search Menu

- From the main menu bar, click **SEARCH**. Menu options display in the vertical pane on the left.



Search Menu

- When choosing a plugin type from the menu options, the only data searched is the data that was identified as a hit when the plugin was processed.

SUB-MENU OPTION	DESCRIPTION
Category DNI	Searches dictionary category hits.
Full Log DNI	Searches all sessions received by XKEYSCORE.
User Activity	Enables a user to search by a user's activity. Example: a user can find a hotmail user's msnMailToken

Search Menu

- All searches are conducted on database tables where the results of the XKEYSCORE engine are stored.

- Each row of a database table contains values from an individual session that was identified as a hit by XKEYSCORE when that plugin or microplugin processed the session.

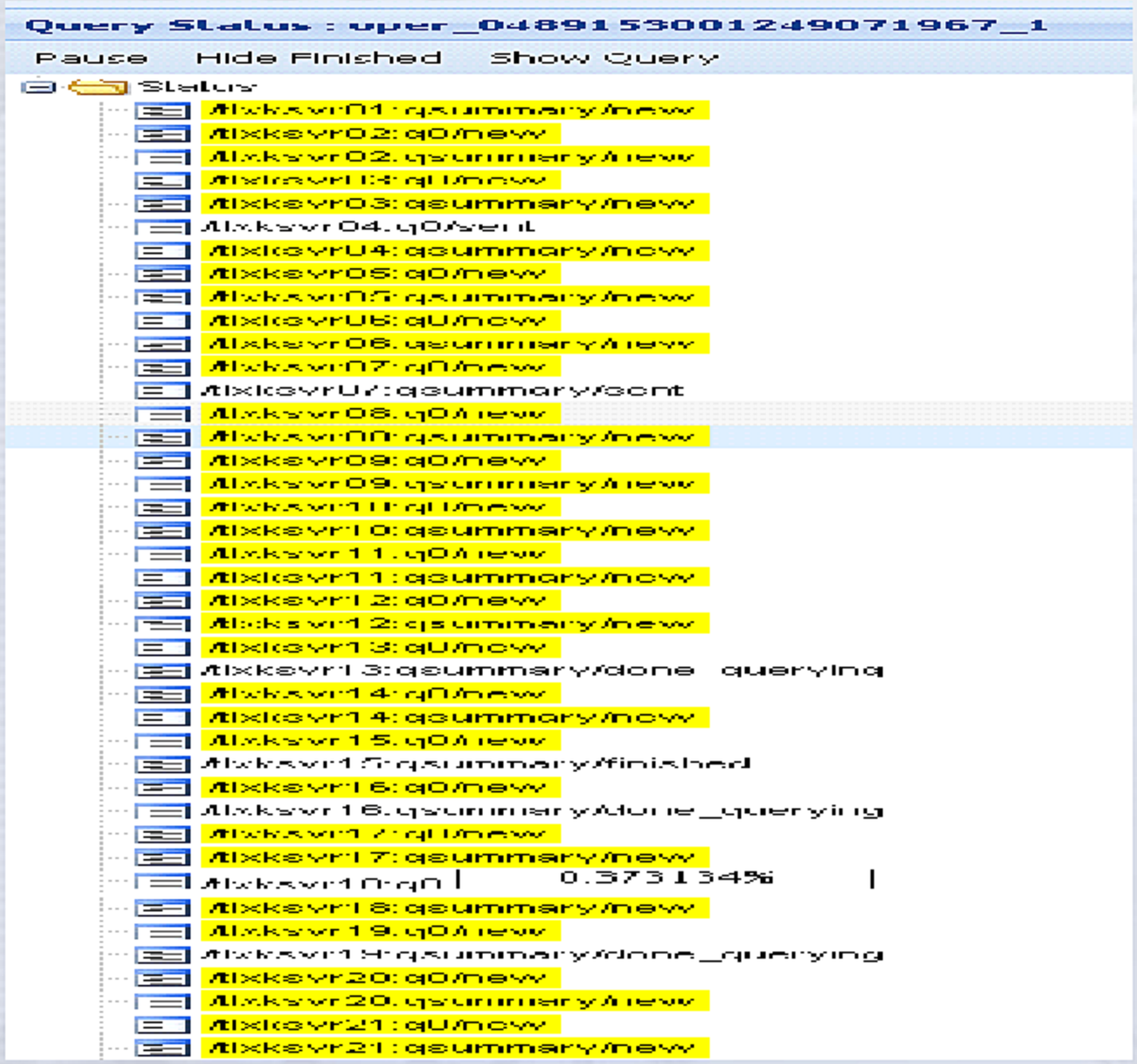
- Each search type query is related to a plugin or microplugin, which performs the metadata extraction.

Search Menu - Details

- Search details can be accessed from the Search status window by clicking **Details.**
- **CURRENT SEARCH DETAILS** window displays and allows the user to watch a query run through the appropriate databases.
- **RESULTS** link in the main menu bar can be used to display a list of all previous search results.
- Queries operate in parallel on each host.

Search Menu – Details Window

Details Window



Results

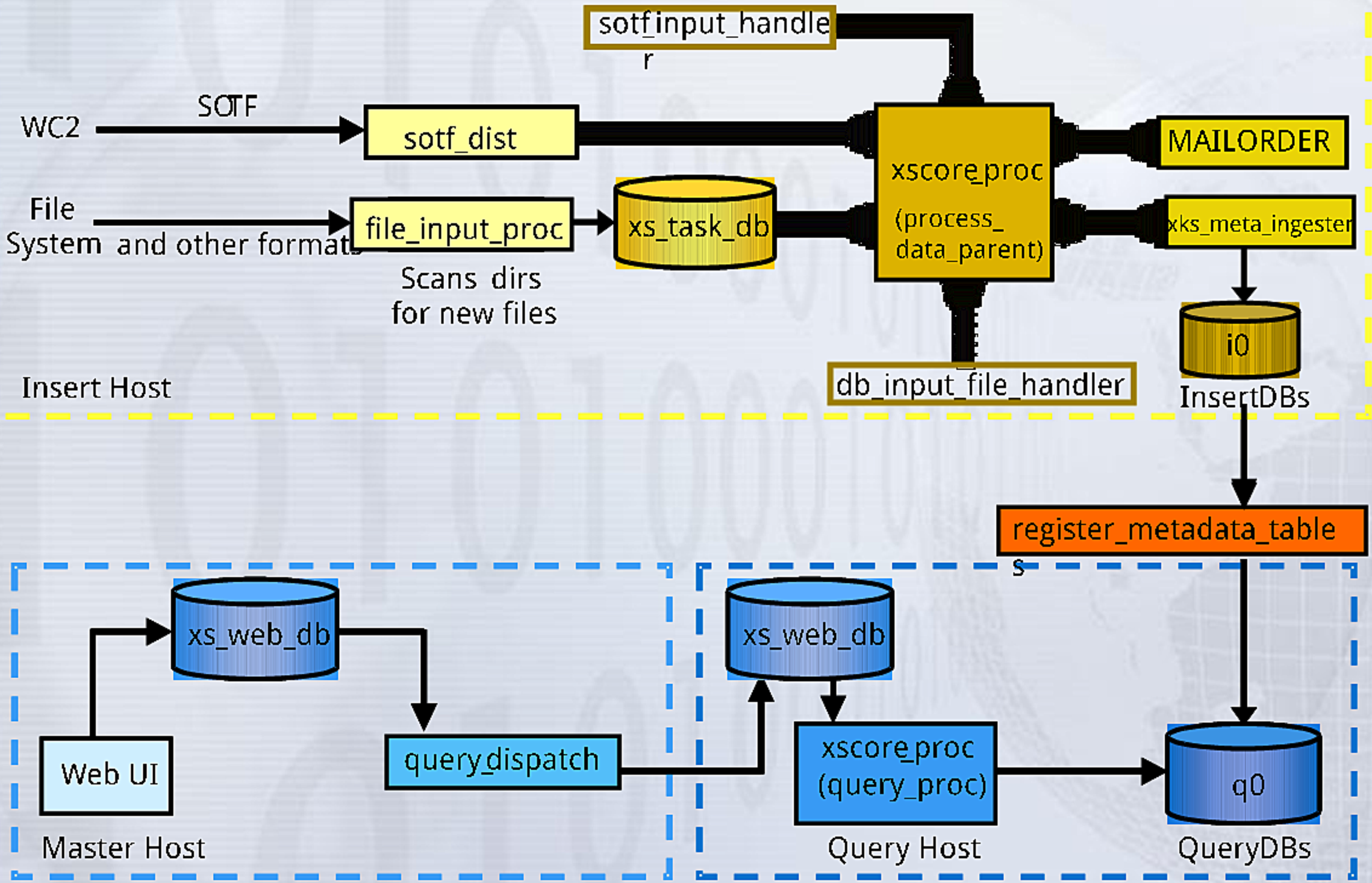
- From the main menu bar, click **RESULTS** to retrieve the results of previous queries.
- By changing the start and stop dates, queries performed between those dates can be viewed.
- If the query name is known, it can be entered in the **QUERY_NAME** field.
- If the USERID is known, it can be entered.
- When complete, a window displays with the matching queries.

XKEYSCORE PROCESSES

Lesson Objectives

- XKEYSCORE Process Data Flow
- Processing Programs
- Query Processes
- Other Processes
- Cronjobs
  - crontab

Back-End Process Data Flow



Processing Programs

- Processing programs are the main processes that extract metadata from the traffic and then database the information in insert databases.

PROGRAM	DESCRIPTION
file_input_proc	Scans for new input files. (before processing moves the file to the .tmp directory of the input directory specified)
sotf_dist	Listens for incoming SOTF sessions
process_data_parent	Processes all new files discovered by file_input_proc or sotf_dist; optionally archives content and databases metadata. Parent process loads all dictionaries and starts up, then forks child processes which do the actual processing.

Processing – process_data_parent

- This process replaces process_data0 through process_dataX
- The “parent” process starts up and loads all the dictionaries, and then “forks” child processes which actually do the processing
- Parent acts similar to the xks_app_launcher, managing restarts for the children when they die
- When dictionaries are modified, parent reloads them and restarts the children
- “xks proc” will show an “X/Y” number next to process_data_parent
  - This is the number of children currently running, over the number that should be running (based on the xks.config num_data_processors setting)
  - pdp will show up yellow anytime X != Y and green when everything is running normally
  - This means when you first (re)start pdp, it will show yellow while it is loading the dictionaries, because none of the actual child process_data’s are running yet
- “xks proc” will report extra or missing process_dataX with a PID of 0
  - Can’t tell what PID missing process_data is suppose to have, because its managed by the parent now

Query Processes

- Query processes are processes that search and submit all necessary tables for the analysts queries.

PROGRAM	DESCRIPTION
query_dispatch	Submits search jobs to search databases and propagates the status of the search and results back to the web server
query_proc	Searches through all the necessary tables for the analysts queries.

Other Process

- Other process which is run from the Application Launcher.

- **mailorder_proc** – polls the /export/data/xkeyscore/outputs/mailorder_working directory by default. Then renames and moves mailorder files to /export/data/xkeyscore/outputs/mailorder.

Other Process

- **xks_meta_ingester** – streams metadata over socket. This process improves database performance. Instead of each xscore_proc writing to the database independently, they stream their metadata over socket to the meta_ingester, which combines it by plugin and writes to the database.
  - Reduces the number of connections to MySQL and gives better control over table size.

Other Process

- **register_metadata_tables** – moves tables from processing database of XKEYSCORE system to query database.
  - Works against the uber_index table
  - uber_index->table_name, base_table_name, join_table
  - Base table – contains common information amongt tables (full_log_xxx_xxxxx table)
  - Extension table – extends the base table
  - Registration process takes place in two phases:
    - Register all base tables
    - Register all extension tables that have had its base table registered

Other Processes

- **signal_acquisition_loopback** – process that feeds modified packets back into the system.
  - Front-end for packet recursion or any other process that feeds modified packets back into the system
    - Reinjects back to front-end – xfip
    - Process is completel independent

Other Processes

- **mpmr_server** – this is the map-reduce server for microplugins, which runs the "Reducer" portion of GENESIS v5 microplugins.
- Runs outside the normal processing flow, and will not affect the rest of the system.
- It has a telnet port (5850) just like an xscore_proc.

Other Processes

- **correlation_server_0** – in-memory map-reduce server for correlation engine.
- Each machine has one correlation_server, and every process_data_parent connects to every correlation_server
  - xscore_proc – 8GB by default
  - uses port 4321

Other Processes

- **xks_comms_server** – a more efficient way to communicate with hosts within and outside an XKS cluster (not currently implement)
  - Automatically handles configuration for talking between slaves, master and overlord at site
  - Configuration is needed to connect to the “peer” on the path towards, other sites
  - Comms configuration lives in $XSCORE_DIR/config/comms/comms.config
  - Supports a “quality of service” which “fairly” distributes available bandwidth to the services that are using comms

Other Processes

- **xks_comms_server**
  - Allow and Peer rules have a “network” parameter which the comms systems uses to determine an “inside” and an “outside” in proxies.
  - Comms system will only accept connections from address ranges it has been specifically configured to allow.
  - Every between 2 comms servers connection should have:
    - “bandwidth_rule” on each side, name doesn’t matter but both rules should usually have same bandwidth cap
    - “allow” rule on one side with a reciprocal “peer” rules on the other side

Other Processes

- **xks_comms_server**
  - Example: If we have a site named "US-123" connecting to xks-central over a 1Mbps link, US-123's config would be:

bandwidth[world] = 1Mbps

peer[00] = address=xks-central.corp.nsa.ic.gov, port=2412, bandwidth=world, network=external

And xks-central would have:

bandwidth[us123] = 1Mbps

allow[00] = address=xkey-master.us123, bandwidth=us123, network=internal

GUI Processes

- Other process which is run from the Application Launcher.
- **GUId** – rescans content against fingerprints when a user clicks to view the content of a session.
- **tomcat.sh** – web server used to host XKS GUI
- **sotftod124server** – downloads sessions
  - Gets called from the GUId process
  - Works with any downloaded traffic that is SOTF

Statistic Processes

- Other process which is run from the Application Launcher.

- **xks_server_stats** – sends to xks_system_monitor on Master and generates stats about the server itself.
  - CPU usage, memory usage, disk space, disk I/O, network traffic, etc.
  - Stats are fed to xks_system_monitor and the system monitor does magic with them.

Statistic Processes

- **xks_system_monitor** – collects stats messages from all over the system (front-end and back-end and the server itself) and summarizes them for forwarding. Optionally it can database stats locally.

Cronjobs

- XKEYSCORE uses a number of cron jobs to perform tasks.

CRONJOB	DESCRIPTION
age_off_new.php	Ages off metadata and content when the disk is near capacity, or when thresholds have been met.
xks update_dictionaries	Pulls updates from various sources.
xks rsync push_config	Copies the /opt/xkeyscore/config directory to the slaves.
rwc_post_to_pub.py	Once an hour kicks off an update request

CRONTAB

- **Crontab** is the program used to install, uninstall or list the tables used to drive the cron daemon.

- The crontab consists of
  - age_off_new.php
  - xks update_dictionaries
  - xks rsync push_config
  - rwc_post_to_pub.py

CRONTAB

- **age_off_new.php**
  - Options:
    - -debug : extra debug statements in the output
    - -info : extra info statements in the output
    - -task_db : explicitly state that the machine is a task host
    - -web_db : explicitly state that the machine is a web host
    - -nosleep : use if you want to run now
  - This process ages off tables and archived data based on the settings in the **xks.config** file and the percentage of disk space used.

CRONTAB

- **xks update_dictionaries**
  - This process pulls the necessarily files from various sources to update the dictionary.
  - Configure /opt/xkeyscore/config/xks.config
    - #[dictionaries]

```
dictionary[0] = type=royale, \
src=sftp://tssi_fvey:tssi_fvey@
xks-control/home/tssi_fvey/xks_dict_update.tar.gz, \
dest=update/xks_dict_update.tar.gz, \
action[0]="cd
$XSCORE_DIR/config/dictionaries/update;$XSCORE_DIR/config/
dictionaries/update/dup_install.pl > /dev/null 2>&1"
dictionary[1] = type=cadence
```

CRONTAB

- **xks rsync push_config**
  - Transfers Master configurations to its slaves.
  - Excludes dot files, “httpd/logs”, loadserver/packages”, “httpd/log”
  - force: option to xks to force push_config when not on the master

CRONTAB

- **rwc_post_to_pub.py**
  - The automatic starProc process is as follows:
    - Hour 1: master asks whoever (say xks-control) for an update, gets the rpm, installs it, there is much rejoicing. The slaves asks the master for the rpm at the same time the master asks xks-control, but obviously the master doesn't have it, so nothing happens.
    - Hour 2: everyone asks for an update again, this time the master has the rpm, the slaves download it and install and there is much rejoicing.
    - The rpm is installed and process_data_parent's are restarted as soon as the rpm is downloaded on a given machine.

DeepDive

DeepDive

- What is a DeepDive?
- Why DeepDive?
- What does a DeepDive look like?
- Front-End Processes
- xFIP
- Promoter

DeepDive XKEYSCORE

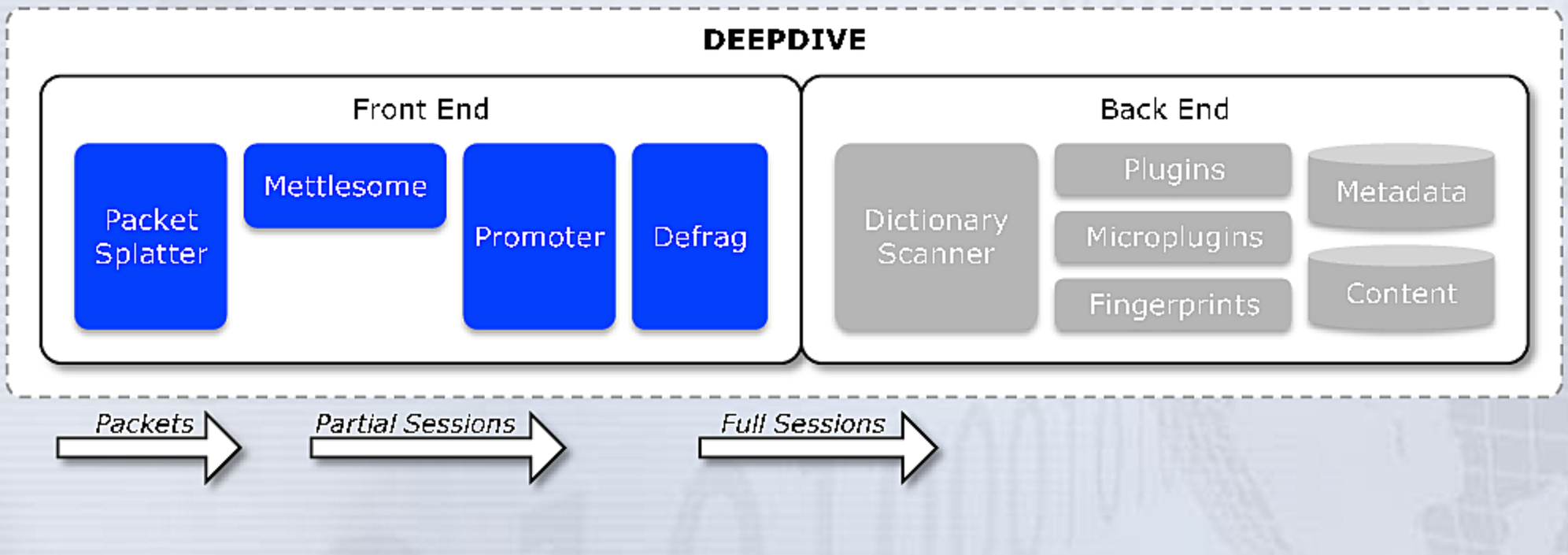
- XKEYSCORE packet processing solution
  - XKEYSCORE's software handles all packet processing
  - No upfront filtering prior to XKEYSCORE
  - XKEYSCORE "promoter" tries to promote richest/most interesting traffic
    - All Strong Selectors
    - Full take ASDF (User Activity metadata)
    - Subset of GENESIS signatures
      - List managed by XKEYSCORE team in concert with collection managers and site engineers
  - 20% - 30% of site traffic is fully processed and can be found via XKEYSCORE search
    - Typically does not include unknown or uninteresting protocols

Why DeepDive?

- Access to most relevant DNI data supporting SigDev and collection missions. Enables new mission capabilities (e.g., Correlation)
- Session promotion can be synchronized and managed based on Genesis signatures, traditional tasking selectors and available resources
  - Provides better scaling
  - Drop unwanted data. Keep the rest and make decisions later and more accurately
- Better control of the processing space
  - Instantiate new mission capabilities and dataflows quickly
  - Troubleshooting and monitoring made easier
- Need access to “raw” packets to support new mission (e.g., Cyber, Bulk Crypt)
  - Sessions can be displayed as Packet Bundles like Wireshark

DeepDive

- What does a DEEPDIVE look like?
  - XKEYSCORE full-take session processor (Back End)
  - High speed packet ingest: an end-to-end solution
  - Intelligent filtering to vary the proportion of traffic retained



XKEYSCORE Front-End Processes

What it's called	What it does	What it means
Packet Splatter	Ingests packets (from files, from the network, from a capture card) in a variety of formats.	If it's a packet stream, it can probably be fed into a DEEPDIVE.
xFip	Fast reassembly of TCP/IPv4, UDP/IPv4 streams*, and TCP/IPv6 and UDP/IPv6 streams*.	DEEPDIVE sessionizes **everything** before making a keep/drop decision.
METTLESOME	Reassembly of streams from less common protocol stacks.	
Promoter	Rule-based filtering of reassembled sessions, based on keyword, country code or appid/fingerprint.	DEEPDIVE intelligently chooses the most useful traffic for retention.
Defrag	Fully rebuilds sessions**	Enough content available to do full decoding/document descent at the Back End

xFIP

- Packet bundles
  - Preserves original packets and packet order
  - Preserves information that is lost during sessionization
  - Original pcap available in the XKS Viewer
- Packet API
  - Microplugins can iterate over raw packets
  - Microplugins can use information that is lost during sessionization
    - E.g. timestamps, flags, checksums
- Packet fingerprints
  - Fired based on observations xFip has made
    - E.g. large sequence gaps, TTL variation

Promoter

- Filters sessions prior to back end processing
  - keywords, regex, country code, appids*
  - SIGDEV: promotion rather than strong selection
- Set the focus of the back end
  - traffic types of interest
  - regions of interest
  - legal/policy constraints
- Set the width of the access aperture
  - promote 20% of 20 signals?
  - promote 100% of 4 signals?
- Set the length of data retention
  - promote 20% and keep for 3 days?
  - promote 30% and keep for 2 days?

```
allow appid chat.*
allow country_code PK
block country_code US-US
```

xks Script

xks Script

- Usage
  - options
- General Commands
- Services
- Actions
- Options

xks Options

- Usage: xks [options] <command>
  - Try 'xks help <name>' to get help on a specific service or action
  - General commands:
    - services - list available services
    - actions - list available actions
    - dependencies [invert] - show service dependencies
    - help [items] - print help on services or actions
  - Services (specify one or more service names or 'all'):
    - start <services> - start the specified services
    - stop <services> - stop the specified services
    - restart <services> - restart the specified services
    - status <services> - print the status of the specified services
    - setup <services> - setup/configure/fix the current xks install

xks Actions

- Actions:
  - accounts_report - sends an email containing accounts usage to the specified users
  - add_admin - sets up a local Linux user to administer XKS
  - change_db_password - changes the XKS database user's password and updates all references to it
  - cluster - cluster actions
  - compile_genesis - compiles GENESIS signatures
  - disk_check - get raid and disk status
  - ext4_format - format $XSCORE_DATA_DIR partition and convert to ext4 filesystem
  - ext4_upgrade contents - convert to ext4 filesystem while preserving of $XSCORE_DATA_DIR (no formatting)
  - fetch - fetch a remote file
  - force_register - force metadata table registration
  - info - show cluster information
  - install_slave - install a slave machine in this cluster
  - local_tagging - checks and/or loads tagging file

xks Actions

- Actions:
  - monitor - view XKS monitoring messages via activemq
  - mpmr_register - force mpmr table registration
  - mysqls - run a mysql script
  - onall - run a command on all machines in this cluster
  - powertower - configure or run a powertower command
  - proc - control XKS processes on this cluster
  - query - display query status or submit a query
  - query_dispatch - command line interface to the XKS query_dispatcher(s).
  - rac - access remote admin ports
  - reload_dictionaries - force running processes to reload dictionaries
  - rsync - push configs or files to slaves
  - search_fields - populates user settings with search fields
  - show_config - show values from xksconfig for specified keys
  - switch - query or rebalance data switch
  - sync_accounts - synchronize user accounts (except for classifications)
  - tail - view realtime xks logs
  - tasking_dump - print out the contents of the xkTasking and xksTasking_voip databases.

xks Actions

- Actions:
  - top - display system performance
  - update_dictionaries - update all XKS dictionaries
  - update_gui_help - update the 'help' pull downs in GUI
  - users - display the users currently logged into the GUI
  - version - show XKS version information
  - watchdog - check and (re) start essential XKS processes.
  - workflow - manually submit a workflow

xks Options

- Options:
  - -verbose : print extra information to the screen
  - -debug : used for debugging script problems

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **first** – initialization service that runs before all others
    - **virus_scanner** – sets up virus scanner, assuming tarballs are present.
    - **ftpd** – enables ftp on the master if mailorder is enabled
    - **distcc** – sets up distributed compiler service
    - **slash_proc** – setup optimal /proc parameters
    - **myricom** – handles installation and configuration 10GigE network cards
    - **home** – sets up the home directory for the xks user account
    - **gcc** – check there is a working compiler on the system
    - **upgrade** – updates configuration files when upgrading to a new version of xks
    - **bashrc** – sets up bash environment variables
    - **beacon** – sets up xks monitoring beacon based on xks.config
    - **tt** – checks connectivity to TRAFFICTHIEF server

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **sendmail** – configures sendmail for use with xks
    - **role_files** – this service installs role-specific files
    - **issue** – sets up the DoD mandatory login warnings
    - **royale_with_cheese** – setups automatic updates
    - **ntpd** – configure ntp based on xks.config
    - **link_summary** – sets up xks link summary GUI
    - **nfsd** – sets up xks-specific nfs mounts
    - **server_certs** – sets up server certificates for SSL applications
    - **openoffice** – installs and configures OpenOffice for use in the xks GUI
    - **init_d –** sets up the xks init.d services
    - **resolver** – sets up resolver config
    - **php** – sets up PHP related stuff. Except php.ini
    - **httpd** – sets up xks-specific httpd configuration

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **www** – sets up GUI configuration files
    - **voip** – sets up voip processing
    - **crond** – ensures xks can use cron and sets up xks cron jobs
    - **sshd** – configures the secure shell service for use with xks
    - **license** – checks for a valid license file and if one isn't found prints a message
    - **syslog** – configures the syslog service for use with xks
      - all xks processes log to /var/log/xks.log
    - **dictionaries** – checks status of any configured dictionaries
    - **cluster_check** – checks network connectivity across the cluster
    - **autofs** – start, stop, restart automounts
    - **loadserver** – start, stop, and setup loadserver
    - **directories** – sets up directories used for xks
    - **auditd** – no help available

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **ldap** – no help available
    - **mysqld** – sets up the mysql server for use with xks
    - **disks** – checks status of disk partition used by xks
    - **databases** – maintains database scheme consistency
    - **local_tasking** – reapplies local tasking if necessary
    - **workflows** – sets up xks default workflows
    - **category_throttle** – overrides default category throttle settings based on overrides specified in xks.config
    - **enrichment_tomcat** – sets up enrichment tomcat java application server
    - **plugin_setup** – populate plugin database tables from xml files, appy default plugin config specified in xks.config, apply overrides from xks.config, regenerate plugin config files from database
    - **crdb** – no help available
    - **tomcat** – sets up tomcat java application server
    - **clickstream** – sets up clickstream service

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **file_input** – sets up directories and database entries needed for file-based input
    - **age_off_db** – synchronizes the database (xs_task_db.age_off) with xks.config's settings for content and metadata. The values in the database will be unconditionally overwritten with those found in xks.config
    - **db_connectivity** – verifies connectivity to critical databases
    - **pdf** – sets up xpdf language packs
    - **ul_age_off** – sets the maximum data retention time to a little over an hour in UL mode.
    - **mDNSResponder** – sets up mDNSResponder for use with SOTF input
    - **app_launcher** – controls the xks app launcher, which is responsible for monitoring xks processes and starting/stopping them as commanded from the GUI
    - **processes_setup** – configures xks processes based on specifications in xks.config
    - **comms** – sets up the XKS communications system configuration
    - **advanced** – performs advanced configuration

xks – General Commands

- Type: xks help services
  - This will list all available services:
    - **endace** – handles all the installation and configuration for Endace Dag packet capture cards
    - **last** – cleanup service that runs after all others

xks - Services

- start
  - xks start mysql
- stop
  - xks stop httpd
- restart
  - xks restart nfs
- status
  - xks status autofs
- setup
  - xks setup plugins

xks - Actions

- **xks onall 'ps –ef | grep xscore | grep –v grep'**
- **xks force_register**

```
[oper@tlxksvr01 run]$ xks force_register
Forced update on xks_meta_ingester
```

- **xks rsync push_config -force**
  - Usage: xks rsync <options> [push_config|push_compiled|push_slaves|push] <src> <dest>
- **xks update_dictionaries**
  - Usage: xks update_dictionaries [test|check|print|force|help]
- **xks version**

```
[oper@tlxksvr01 run]$ xks version
1.5.9-65
```

- **xks info**

```
[oper@tlxksvr01 run]$ xks info
Site            : Timberline - SV
SIGAD           : USF-790
PDDG            : IE
XKS version     : 1.5.9-65
Master          : tlxksvr01
Num slaves      : 13
Input           : file, sotf
Config          : managed_srtd
```

xks - Actions

- **xks query servers**

```
[oper@tlxksvr01 run]$ xks query servers

tlxksvr02:q0
     3a   14s   98n   54w   2012-12-05 16:07:22   (top)

tlxksvr03:q0
   230a    0s    0n    0w   2012-12-05 17:55:02   (top)

tlxksvr04:q0
   230a    0s    0n    0w   2012-12-05 17:55:02   (top)

tlxksvr05:q0
   225a    0s    4n    0w   2012-12-05 17:55:02   (top)

tlxksvr07:q0
   230a    0s    0n    0w   2012-12-05 17:55:02   (top)

tlxksvr08:q0
   225a    0s    5n    0w   2012-12-05 17:55:02   (top)

tlxksvr09:q0
     0a    0s 173n    2w   2012-12-05 17:55:02   (top)

tlxksvr10:q0
   230a    0s    0n    0w   2012-12-05 17:55:02   (top)

tlxksvr11:q0
   230a    0s    0n    0w   2012-12-05 17:55:02   (top)

tlxksvr12:q0
   225a    0s    4n    0w   2012-12-05 17:55:02   (top)

a=awaiting dispatch, s=sent, n=new, w=working
timestamp shows earliest submitted but unfinished query
current time: 2012-12-05 18:02:09
```

xks Example

- **xks proc**

```
[oper@tlxksvr01 run]$ xks proc
GUI   GUId                          qp    query_proc
cli   clickstreamService.sh         rmt   register_metadata_tables
cms   check_mailorder_site.php      s2d   sotftod124server
cs00  correlation_server_0          sab   signal_acquisition_base
ctp   cadence_tasking_proc          sal   signal_acquisition_loopback
enr   enrichment-tomcat.sh          sd    sotf_dist
file  file_input_proc               sst   strong_selector_targeting
mp    mailorder_proc                tom   tomcat.sh
mpmr  mpmr_server                   xcs   xks_comms_server
pd#   process_data#                 xmi   xks_meta_ingester
pdp   process_data_parent           xsm   xks_system_monitor
qd    query_dispatch                xss   xks_server_stats
Run 'xks proc full' to show full listing

tlxksvr01 GUI cli cms      ctp enr file mp            qd qp rmt s2d sab:4/4           sst tom xcs xmi xsm xss
tlxksvr02             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr03             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr04             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr05             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr06             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr07             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr08             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr09             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr10             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr11             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr12             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr13             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
tlxksvr14             cs00              mpmr pdp:4/4    qp rmt     sab:4/4 sal:6/6 sd        xcs xmi     xss
Process consistency check OK on all hosts
```

xks Example

- **xks proc full**

```
[oper@tlxksvr01 run]$ xks proc full
app launcher status: RUN (pid 30705)

  id  hostname   program                  arguments                        commanded  actual      pid
  14  tlxksvr01  cadence_tasking_proc     --myfdi XYD --pddg IE --dig...   RUN        RUN       31136
   4  tlxksvr01  check_mailorder_site.php                                  RUN        RUN       31019
 723  tlxksvr01  clickstreamService.sh                                     RUN        RUN       31053
 654  tlxksvr01  enrichment-tomcat.sh                                      RUN        RUN       31200
   9  tlxksvr01  file_input_proc                                           RUN        RUN       31104
   1  tlxksvr01  GUId                                                      RUN        RUN        9335
 548  tlxksvr01  mailorder_proc          --copydir /export/data/xkey...   RUN        RUN       31140
   8  tlxksvr01  query_dispatch                                            RUN        RUN       17549
   3  tlxksvr01  query_proc                                                RUN        RUN       30965
 193  tlxksvr01  register_metadata_tables --loglevel error                 RUN        RUN       31143
 709  tlxksvr01  signal_acquisition_base  -f generic_packet_to_bundle...   RUN        RUN: 4/4  31236
  12  tlxksvr01  sotftod124server                                          RUN        RUN       31108
 461  tlxksvr01  strong_selector_targeting                                 RUN        RUN       31145
  13  tlxksvr01  tomcat.sh                                                 RUN        RUN       31124
 653  tlxksvr01  xks_comms_server                                          RUN        RUN       31148
   5  tlxksvr01  xks_meta_ingester                                         RUN        RUN       31051
 462  tlxksvr01  xks_server_stats                                          RUN        RUN       31138
  11  tlxksvr01  xks_system_monitor                                        RUN        RUN       13986
 724  tlxksvr02  correlation_server_0     --loglevel debug                 RUN        RUN       13946
  31  tlxksvr02  mpmr_server                                               RUN        RUN        7150
 730  tlxksvr02  process_data_parent      --max-mem 20                     RUN        RUN: 4/4  22661
  30  tlxksvr02  query_proc                                                RUN        RUN       14754
 187  tlxksvr02  register_metadata_tables --loglevel error                 RUN        RUN       14994
 710  tlxksvr02  signal_acquisition_base  -f generic_packet_to_bundle...   RUN        RUN: 4/4  14996
  41  tlxksvr02  signal_acquisition_loopback -f packet_aux.config -i loo... RUN        RUN: 6/6  14990
 259  tlxksvr02  sotf_dist                                                 RUN        RUN       14845
 679  tlxksvr02  xks_comms_server                                          RUN        RUN       14992
  38  tlxksvr02  xks_meta_ingester                                         RUN        RUN       14970
 473  tlxksvr02  xks_server_stats                                          RUN        RUN       14988
```

xks Example

- **xks query**

```
[oper@tlxksvr01 run]$ xks query
id          user        type                search start     search stop      duration  status
66250201                http_parser         00:00 12/2/12    00:00 12/6/12    00:00:10  ongoing
66250183                full_log            00:00 12/3/12    00:00 12/6/12    00:00:17  ongoing
66250155                full_log            00:00 12/4/12    00:00 12/6/12    00:00:40  ongoing
66250127                geo_info            00:00 11/30/12   00:00 12/6/12    00:01:16  ongoing
66250052                email_addresses     00:00 11/21/12   00:00 12/6/12    00:03:36  ongoing
66249873                full_log            22:00 12/3/12    21:59 12/4/12    00:11:31  ongoing
66249660                full_log            00:00 12/2/12    00:00 12/6/12    00:18:57  ongoing
66244233                category            00:00 11/5/12    00:00 12/6/12    00:42:17  ongoing
66244135                full_log            00:00 11/28/12   00:00 12/6/12    00:44:30  ongoing
66244009                http_parser         00:00 11/5/12    00:00 12/6/12    00:48:49  ongoing
66243967                http_parser         00:00 11/5/12    00:00 12/6/12    00:49:34  ongoing
66243855                document_metadata   00:00 11/21/12   00:00 12/6/12    00:50:48  ongoing
66243785                correlation         00:00 11/5/12    00:00 12/6/12    00:52:46  ongoing
66243463                correlation         00:00 11/21/12   00:00 12/6/12    00:56:13  ongoing
66243071                email_addresses     00:00 11/1/12    00:00 12/6/12    01:08:24  ongoing
66242973                user_activity_exif  00:00 11/21/12   00:00 12/6/12    01:12:03  ongoing
66242413                http_parser         00:00 11/28/12   00:00 12/6/12    01:26:32  ongoing
66242315                full_log            00:00 12/4/12    00:00 12/6/12    01:30:52  ongoing

There are 18 queries in progress
```

xks Example

- **xks query detail**

```
[oper@tlxksvr01 run]$ xks query detail id=66251065

Query Summary
  Userid:
  Type: email_addresses
  Searching from 00:00 11/30/12 to 00:00 12/6/12
  Duration: 00:01:29
  Priority: 5
  Cancel: N(o)
  Max Results: 10000
  Max Time: 6000

Query SQL
  Name: smatev3_4
  Classification: S,IS/SI,NSA NOFORN,IS,HCS,S/SI,MUSCULAR,REL_USA,NSA_NOFORN,SI,C,R
  Where: WHERE datetime >= '2012-11-30 00:00:00' AND datetime <= '2012-12-06 00:00:00' AND   email = '          '   AND   domain = 'hotmail.com'


Query Status
        host                            database          status
        tlxksvr01                       q0                finished
        tlxksvr02                       q0                ongoing
        tlxksvr03                       q0                finished
        tlxksvr04                       q0                finished
        tlxksvr05                       q0                finished
        tlxksvr06                       q0                finished
        tlxksvr07                       q0                finished
        tlxksvr08                       q0                finished
        tlxksvr09                       q0                finished
        tlxksvr11                       q0                finished
        tlxksvr12                       q0                finished
        tlxksvr13                       q0                finished
        tlxksvr14                       q0                finished
        tlxksvr10                       q0                finished
```

Monitoring

Lesson Objectives

- Executables
  - mysqls
  - onall
  - xks onall
  - xks monitor
  - sotf_stat
  - xks top
- Web Status
- Additional Monitoring

Executables

- System monitoring can be performed from the command line using the following executable commands:
  - mysqls
  - onall
  - xks onall
  - sotf_stat
  - xks top

mysqls

- The **mysqls** bash shell script can be used to execute MySQL statements from in the */opt/xkeyscore/bin.shells/sysadmin/mysqls* directory. The most commonly used options in **mysqls** are:
  - **status** – displays file-based input statistics.
  - **speed** – displays the total file based input processing rate (Mbps)
  - **speed1** – displays file-based input processing rate (Mbps) per input source.
  - **speed2** – displays file-based input processing rate (Mbps) per xkeyscore processing server.
  - **count** – displays the count of input files in the new, working, error, and done states.

mysqls

- **mysqls status**

```
[oper@tlxksvr01 run]$ mysqls status
status  count(*)        sum(filesize)   priority
bitrate_Mbps
NULL
```

xks onall

- **xks onall ‘xks mysql status’**

```
[oper@tlxksvr01 run]$ xks onall 'xks mysql status'
Do you want to execute "xks mysql status" onall? [y|n] y


~~~~ tlxksvr01 ~~~~

status mysqld
        mysqld is running



~~~~ tlxksvr02 ~~~~

status mysqld
        mysqld is running



~~~~ tlxksvr03 ~~~~

status mysqld
        mysqld is running



~~~~ tlxksvr04 ~~~~

status mysqld
        mysqld is running



~~~~ tlxksvr05 ~~~~

status mysqld
        mysqld is running
```

xks monitor

- This script will monitor your front-end processes.
- Type: **xks montor** or **xks monitor h** to receive the help menu

```
XKEYSCORE-Menu
Command             Name                Description
c                   config              Configure this utility
d                   dataflow_all        FrontEnd Dataflow Menu
b                   dataflow_be         BackEnd Dataflow Menu
m,h                 menu                View this menu
a                   packet_splatter     Packet Acquisition [Front End]
p                   process_data        Process Data [Back End]
q                   quit                Quit/Exit
s                   servers             Server Stats (CPU,IO,etc.)
t                   sotf_input          SOTF Input [Back End] 'xks top' replacement
f                   xfip                Sessionization [Front End]
```

xks monitor

- Type: **xks montor f** to receive xfip stats

```
XKEYSCORE-Sessionization [Server View Press '4']
Casenotation          Rate(Mbps)      Loading%     TCPQuality%      Rate(Pkt)     Count(Pkt)        Punt%   Fragments%
7DH115181980000             0.00          0.00            0.00           0.00              0         0.00         0.00
7DH115209040000             0.00          0.00            0.00           0.00              0         0.00         0.00
Total: 0.00          [KEY: STM1, STM4, STM16, STM64]
```

sotf_stat

- The **sotf_stat** command is used to display the SOTF (streaming object transfer format) input statistics for an entire cluster.
- The statistics include total number of **process_data's** running on the cluster, session input rate (sessions/sec), total bytes input (Mbps), and total bytes output to process_data(s) (Mbps).

sotf_stat

- To execute the **sotf_stat** script:
  - Log on to the server and open a terminal window.
  - Type *sotf_stat* because the command is in the path
  - Type *s* to toggle the summary statistics view from total statistics to individual host statistics.
  - Type *q* to quit the program

sotf_stat

- The **sotf_stat** script lists the hostname, number of process_data's currently running, Mbps, number of sessions, and number of bytes.

```
XKEYSCORE SOTF Statistics
Hostname          #In #OA/#OC   Mbps In       Sess In      Bytes In   Sess Q  MaxBlk
mhxkssvr02           7    4/4       18.66    410920147 481811297497 6        0
mhxkssvr03           3    4/4       16.15    410121822 478354986500 4        0
mhxkssvr04           3    4/4       16.65    410444622 478132027699 2        0
mhxkssvr05           3    4/4       15.79    409831857 475993930392 0        0

PRC:  15/ 15  Rate: 64.52 Mbps  Sessions: 1641358767  Bytes: 191432892127 72
```

xks top

- The **xks top** script lists the hostname, Mbps sotf rate, number of process_data's running, the % of CPU, and % of IO wait.

hostname	sotf	#procs	cpu%	iowait%
mhxkssvr01	-0.00	0	0.53	0.02
mhxkssvr02	21.08	4	12.88	7.94
mhxkssvr03	13.55	4	13.35	7.28
mhxkssvr04	14.97	4	14.50	8.63
mhxkssvr05	14.13	4	17.14	8.01
TOTAL	63.74	16	11.68	6.38

Additional Monitoring

- **xks tail**

```
[oper@mhxkssvr02 ~]$ xks tail
Dec  5 18:27:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:27:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:27:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:27:59 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:27:59 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:27:59 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:28:08 mhxkssvr02 sotf_dist[13986] : <sotf_dist_t> NOTICE: cu
Dec  5 18:28:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:28:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
Dec  5 18:28:29 mhxkssvr02 register_metadata_tables[13877] : <register
st (automatic?) repair failed
```

Troubleshooting

Lesson Objectives

- Common Troubleshooting techniques
  - Full Disk
  - Sotf Problems
  - MySQL
  - Processing Problems
  - Outputs
  - Query Problems
  - Directory Permissions

Disk Full

- **/var/log/xks.log (xks tail)** – Relevant error messages can be viewed in this file. This directory may fill the disk, some known reasons are:
  - process_data has lost its connection with the sotf_dist and is continuously trying to reconnect to sotf_dist.
  - nfs error may have occurred and a detailed message can be found in the file /var/log/messages.
  - Corrupt tables in the insert database.
  - Check to make sure the age_off_new.php cronjob aged off old metadata and content.

Disk Full *continued…*

- **/export/data/xkeyscore/inputs**
  - If there are too many files in the directory:
    - file_input_proc may be running improperly or not at all. Verify that file_input_proc is running from the command line type:
      - ps –ef | grep file_ | grep –v grep
      - xks proc
    - The file_input_proc may need to be restarted.
- No new files in the directory:
  - The directory may not be cross-mounted properly, if automounting is used.

Disk Full *continued…*

- **/export/data/xkeyscore/mysql/i0 or i1**
  - If */export/data/xkeyscore/mysql/i0* or *i1* are filling and *q0* and/or *q1* maintains its size, **register_metadata_tables** may not be working properly.
    - Restart process and watch the databases to see if it is transferring files or run the process by hand to troubleshoot further.
  - If */export/data/xkeyscore/mysql/q0* or *q1* is filling, the **age_off_new.php** script may be running improperly or not at all.
    - First run the command: `ps -ef | grep age_`
      - If script isn't running, try running it by hand.
      - If script is running, then stop script and try running it by hand to see if there are any errors.

SOTF Problems

- **Can an sotf_input_proc run with a file_based file_input_proc?**
  - Yes. Both input types can run on XKEYSCORE given that each are independently configured correctly.
- **Can file-based input be disabled so that only sotf_input is processed?**
  - If moving from file-based input to sotf_input, and no additional file-based input is expected, the plug-in for file-based input, *db_input_file_handler*, should be disabled.
  - From the TERMINAL WINDOW:
    - Stop all the processes : xks stop all
    - Change /opt/xkeyscore/config/xks.config to set file_input to 'no'.
    - Setup the config : xks setup plugins, xks setup processes
    - Rsync change to slaves : xks rsync push_config
    - Restart process_data's : xks proc restart pdp

SOTF Problems *continued…*

- **Is XKEYSCORE receiving input?**
  - To verify whether XKEYSCORE is receiving input, run the **sotf_stat** command to get the current input statistics.
  - If no connection is visible, from the command line:

1. Type `telnet localhost 5042`
2. Output statistics for the specified sotf_dist
3. If running, type `ps -ef | grep sotf_dist`
4. Determine if sotf_dist's are listening on the specified port:
   Type `telnet localhost 5040`
   If command is refused, the sotf_dist is not listening on the port. Continue with step 5.
5. Type `netstat -a | grep 5040`
   If a connection is established for this port then most likely the sotf_dist is listening on this port.

SOTF Problems *continued…*

- netstat will tell if…
  - sotf_dist is listening for connections
  - If connections have been made to the sotf_dist
  - If we are “backing up”- i.e., if sotf_dist is running but has no process_data’s connected to it, it won’t be able to send data anywhere, so eventually its network receive queue will get large.
    - Ideally, the receive queue should always be 0.

SOTF Problems *continued…*

- **Is the process_data_parent running?**
  - At least one process_data must be running and synchronized with the sotf_dist for it to receive input.
    - If problems continue, run the sotf_dist in a terminal to further troubleshoot and identify error messages.

MySQL – D124 file troubleshooting

- **Symptom: A lot of errors or too many errors display when performing the command 'mysqls status':**
  1. First try, `mysqls cleanup`, in a terminal window.
  2. Type `mysqls status`
  3. Type `mysql xs task_db;` to log into MySQL database and use the xs_task_db database.
  4. Execute the following command: `delete from tar_files where status="error";`
  5. Exit out of the MySQL database
  6. Type `mysqls status`
     There should no longer be any error files.

Processing Problems

- The heart of the XKEYSCORE processing engine is the xscore_proc with related plugins.
- Input to the xscore_proc is either file-based and from an file_input_proc, or streaming from an sotf_input_proc.
- After processing, the written metadata to the insert databases can be sent to a follow on system for additional processing.

Processing Problems *continued…*

- **How many process_data's should be running on a host?**
  - From the XKEYSCORE GUI:
    - Click **ADMIN > Processing > Computer Resources**
    - Determine how many *process_data's* are configured to be running on the specified host.

Processing Problems *continued…*

- **How many xscore_proc's are actually running on a host?**
  - Log onto the XKEYSCORE server and open a terminal window.
  - Type `ps -ef | grep xscore | grep -v managed_`

```
[oper@lxks-vr02 ~]$ ps -ef | grep xscore | grep -v managed_
oper     22661 11539 15 Dec03 ?        07:38:24 xscore_proc --parent --base_port 5550 --external_appid_compile --db i0 --renice_level -5 --command db_force_next --max-mem 20 --loglevel error --restart_childre
n --childpid 31396 --childpid 29482 --childpid 29619 --childpid 29710
oper     29482 22661 26 19:08 ?        00:06:46 xscore_proc child-1  --port     5551 --external_appid_compile --db i0 --renice_level -5 --command db_force_next --max-mem 20 --loglevel error

oper     29619 22661 26 19:08 ?        00:06:31 xscore_proc child-2  --port     5552 --external_appid_compile --db i0 --renice_level -5 --command db_force_next --max-mem 20 --loglevel error

oper     29710 22661 26 19:09 ?        00:06:24 xscore_proc child-3  --port     5553 --external_appid_compile --db i0 --renice_level -5 --command db_force_next --max-mem 20 --loglevel error

oper     31396 22661 41 19:28 ?        00:02:13 xscore_proc child-0  --port     5550 --external_appid_compile --db i0 --renice_level -5 --command db_force_next --max-mem 20 --loglevel error
```

Processing Problems *continued…*

- **xks_app_launcher is running, but not starting processes specified in the Computer Resources window?**
  - This may indicate that the xks_app_launcher is defunct. Use the **kill** command to kill the app_launcher and its related sub-processes:
    - Type `pkill -f app_`
      - If a PID is not being specified, use the **pkill** command. The **–f** option kills all of the sub-processes.
    - Type `ps` to look for the new xks_app_launcher process.

Processing Problems *continued…*

- If, after performing the procedures, the xks_app_launcher is still not starting applications:
  - In a terminal window, manually run the problem process to see if there are any error messages.
  - The xks_app_launcher on any host is dependent on the access of the xs_task_db.proc_resources database table on the master. Verify that the specified host can access the master's database and */opt* directory.
  - On the slave system type `mysql xs_task_db -h <masterhostname>`
    - performs a remote MySQL server login

Processing Problems *continued…*

- To test the xscore_proc, type:

  `telnet <process host> <port number>`

  Optional commands to assist trouble shooting are:

- **sbr** – prints the processing rate for the single xscore-proc.
- **sh** – displays dictionary hit statistics.
- **ss** – displays statistics on the internal plug-in processing rates.
- **help** – there are many commands and can be described in the help menu.

Processing Problems *continued…*

- If the **process_data_parent** continues to deny access through the command port, and input still has not started processing, check the input source.
- Run the process in a terminal window with the argument --**loglevel debug**, to view debug messages.
- The command port also provides processing rates and statistics for troubleshooting performance issues, outages, and general administration issues.

Outputs - *Mailorder*

- **/export/data/xkeyscore/outputs/mailorder**
- If there are no new files in the MAILORDER directory, MAILORDER may not be working properly. Possible causes are that:
  - Files are being written to the wrong directory or it is not configured properly
  - Permissions on the MAILORDER directory will not allow MAILORDER to move files

Query Dispatch

- **Query dispatch** is the process that submits search jobs to search databases and propagates the status of the search and the results of the search back to the web server.
- After submitting a new query, Search Status window displays a summary listing query name, date and time submitted, number of databases complete, and number of results.

Query Problems

- The query never moves to the finished state.
  - If a database outage or a comms outage occurs, results will not be reported from the single system. However, results from all other databases will return properly with the query results, but they will not appear in this state.

Query Problems

- **Query job status is stuck in awaiting_disbatch.**
  - If a status appears stuck in this state, the *query_dispatch* may not be running on the web server. To determine whether it is running:
    - Type ps -ef | grep query_
  - If the process is not running, restart it from the XKEYSCORE GUI or troubleshoot the xks_app_launcher.

Query Problems

- Another cause of this scenario is that a query database may have hung up the query dispatch process. Check the progress of queries on the query database hosts by viewing the table *sdb_query_jobs* in the query database, which tracks the status of queries:
  - Type `mysql q0`
  - Type `select status,count(*) from sdb_query_jobs where group by status;`
- The select statement displays the current state of the queries on the query host. If many more queries appear in the **new** state when compared to other query databases, begin troubleshooting the problem *query_proc* on the specified query database.

Query Processing

- The query is in the sent state, but never appears in new.
  - After the **query_dispatch** process disbatches the query, the status is moved to **sent.** A query moves to the **new** state when the query has been placed in the query processing queue on the query_host.
  - If a query does not move to the **new** state in a reasonable amount of time, the connectivity of the database should be tested.

Query Processing

- To check the progress of queries on the query database hosts, view the table *sdb_query_jobs* in the query database, which tracks the status of queries:
  - Type `mysql q0`
  - Type `select status,count(*) from sdb_query_jobs where group by status;`
- The **select** statement displays the current state of the queries on the query host. If many more queries appear in the **new** state when compared to other query databases, begin troubleshooting the problem *query_proc* on the specified query database.

Query Processing

- **The query appears in the new state, but never finishes.**
  - query is in the **new** state, has been received by the query host and placed in a queue waiting to be processed.
  - Queries can become backlogged with a large number of queries waiting in the **new** state, though the *query_proc* is processing the queries properly. It is hard to predict the time to work off a query backlog, but using the following **select** statement the status of queries for the current day can be checked for processing trends.

Query Processing

- **To display the number queries in each state for the current day:**
  - Type

```
select status,
count(*),datetime_submitted,(UNIX_TIMESTAMP(now())-
UNIX_TIMESTAMP(datetime_submitted))/3600 from
sdb_query_jobs where (datetime_submitted>(now() - INTERVAL
'1' DAY)) group by status;
```

- **To display the number of queries processed per hour for the current day:**
  - Type

```
select status,count(*)/24 AS queries_per_hour from
sdb_query_jobs where cancel!="C" and
(datetime_submitted>(now() - INTERVAL '1' DAY)) AS Backlog
group by cancel;
```

- If processing properly, queries can take hours, if not days, to complete based on the backlog and the processing trends.

Retrieving Metadata and Content

- **Queries complete but there are no results.**
  - If queries complete, but no results are visible, verify that the date range of the query coincides with the collection date of the data. If using test data, test the query system by putting the start date range at a year or two older to assure it is not old test data.
  - Verify that query metadata is in the query database by checking the contents of the */export/data/xkeyscore/mysql/{query_db}/* directory.

Retrieving Metadata and Content

- **Queries complete and metadata returns, but there is no content.**
  - The metadata in the XKEYSCORE viewer displays the host and directory path of the content file. Verify the content file exists using the **ls -l** command. Trace a dataflow issue if the file does not exists. If the content file exists, confirm the httpd daemon is started on all slave systems. To confirm the httpd daemon:
  - 1. Type `su - oper`
  - 2. Type `xks status httpd`
  - 3. If the daemon is not on, type `xks start httpd`

Query Results

- To troubleshoot problems with metadata or content from a query, it will be necessary to retrieve the actual content, since recreating the problem is very difficult. This can be accomplished from the XKEYSCORE GUI. Click **RESULTS** and begin a search of the questionable queries.

Questions?