(S//REL) Royale with Cheese Updater
How To Guide

(S//REL) Setup and configuration of the Royale with Cheese Updater takes just a few steps. Follow the steps below to quickly "get up and running." Refer to the Read Me for greater detail about each option.

(U//FOUO) Log on as the user `oper`, and follow these steps to configure the Royale with Cheese Updater:

1. (S//REL) At the command line from within any directory, type `viconfig` and then press `Enter`. The *xks.config* file will open.
2. (U//FOUO) Next to the *rwc_publisher* option, type the IP address of the Publisher from which the current Subscriber (Master or Proxy) will receive downloads.

   Note: (U//FOUO) Alternatively, you may use a host name instead of an IP address if you are sure the current machine can resolve the name of its Publisher.
3. (S//REL) Type `:wq!` and then press `Enter` to exit *xks.config*.
4. (S//REL) At the command line, type `xks setup rwc` (`xks setup royale_with_cheese` in version 1.5.8) and then press `Enter`. This sets up directories and variables that are used internally.
5. (S//REL) At the command line, type `rwc` and then press `Enter` to get to the rwc directory.
6. (U//FOUO) From the rwc directory, type `exe/rwc_post_to_pub.py` and then press `Enter` to run the *rwc_post_to_pub.py* script. This sends the post from the Master or Proxy to its Publisher.

(U//FOUO) Log on as the user `oper`, and follow these steps to customize Publisher/Subscriber configuration. A complete list of Publisher/Subscriber options can be found at the end of this document.

1. (S//REL) At the command line from within any directory, type `viconfig` and then press `Enter`. The *xks.config* file will open.
   a. (S//REL) Confirm that a Publisher IP address or host has been entered beside the *rwc_publisher* option.
   b. (S//REL) Next to the *rwc_mox_chunk* option, type the maximum allowable "chunk" size of a Publisher file (in bytes) that is manageable for download by your Subscriber node. For example, to change the maximum chunk from the default size to 1MB, you can change *rwc_mox_chunk=5000000 to rwc_mox_chunk=1000000.* More details regarding *rwc_* max_chunk are on page 6.
2. (S//REL) Type `:wq!` and then press `Enter` to exit *xks.config*.
3. (S//REL) At the command line, type `cfg` and then press `Enter` to get to the config directory.

Classified By: [redacted]
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20370901

4. (S//REL) From the config directory, type `vi xks.rwc.config` and then press `Enter` to open the *xks.rwc.config* file. Editing this file is optional. If no edits are made to this file, then the configuration automatically will get and publish every module that is available.

5. (S//REL) Change Publisher options within *xks.rwc.config*:

   a. (U//FOUO) Customize any pub_xx option to indicate whether the current machine will or will not publish a given package, xx, to its Subscribers. For example, if *pub_metodoto_forworder=true* and you do not want the machine to publish any metadata forwarder packages, change the option to read:

   ```
   pub_metadata_forwarder = false
   ```

   b. (U//FOUO) Customize *odd/remove* options:

      i. (U//FOUO) To add a file, type:

      ```
      add_file[0] = src="rel/path/to/file",
      dest="rel/path/to/dest",
      classification="CLASSIFICATION"
      ```

      where *src* is the path to the source file including the file name), *dest* is the destination directory of the file, and *CLASSIFICATION* is the classification of the file being published.

      ii. (U//FOUO) To set a file to be removed from the Subscriber's machine, type:

      ```
      rm_file[0] = rel/path/to/file
      ```

   (U//FOUO) See pages 7 and 8 for all Publisher options.

6. (U//FOUO) Change Subscriber options within *xks.rwc.config*:

   a. (U//FOUO) Customize any sub_xx option to indicate whether the current machine will or will not accept a given package, xx, from its Publisher. For example, if *sub_metadata_forwarder=true* and you do not want the machine to subscribe to any metadata forwarder packages, change the option to read:

   ```
   sub_metadata_forwarder = false
   ```

   b. (U//FOUO) Customize *sub_new_modules* to indicate whether a site can or cannot automatically receive all new modules during a given update. The options are true or false:

   ```
   subscribe_new_modules=true
   subscribe_new_modules=false
   ```

   c. (U//FOUO) Use the `notify_email{}` option to enter the e-mail address(es) of anybody who should receive an e-mail message when new modules are available but that have not been automatically downloaded. Separate each address with a comma. For example, type:

   ```
   notify_email{address1,address2,address3}
   ```

   where `addressx` is an e-mail address.

d. (U//FOUO) Set *ignore* options to indicate to the Publisher that this specific Subscriber does not want updates for the specified files/directories. For example, type:

```
ignore_file[0] = rel/path/to/file
```

e. (S//REL) Enter options that execute bash scripts that are run only after installation of the update is complete. For example, type:

```
command[0] = source /opt/xkeyscore/bashrc && xks proc restart pd
```

(U//FOUO) See pages 8 and 9 for all Subscriber options.

7. (S//REL) Type `:wq!` and then press `Enter` to exit *xks.rwc.config*.

(U//FOUO) On an hourly basis, Masters and Proxies will run the post_to_pub script that will automatically download the latest packages from their Publisher. After they have received an update, they will run the push_config script to push new/updated content to *oil* slave nodes in their cluster. This ensures all nodes on the cluster have the same configuration. Additionally, packages containing the appropriate starproc rpm will subsequently be chunked and pushed for reassembly and installation on the slave nodes.

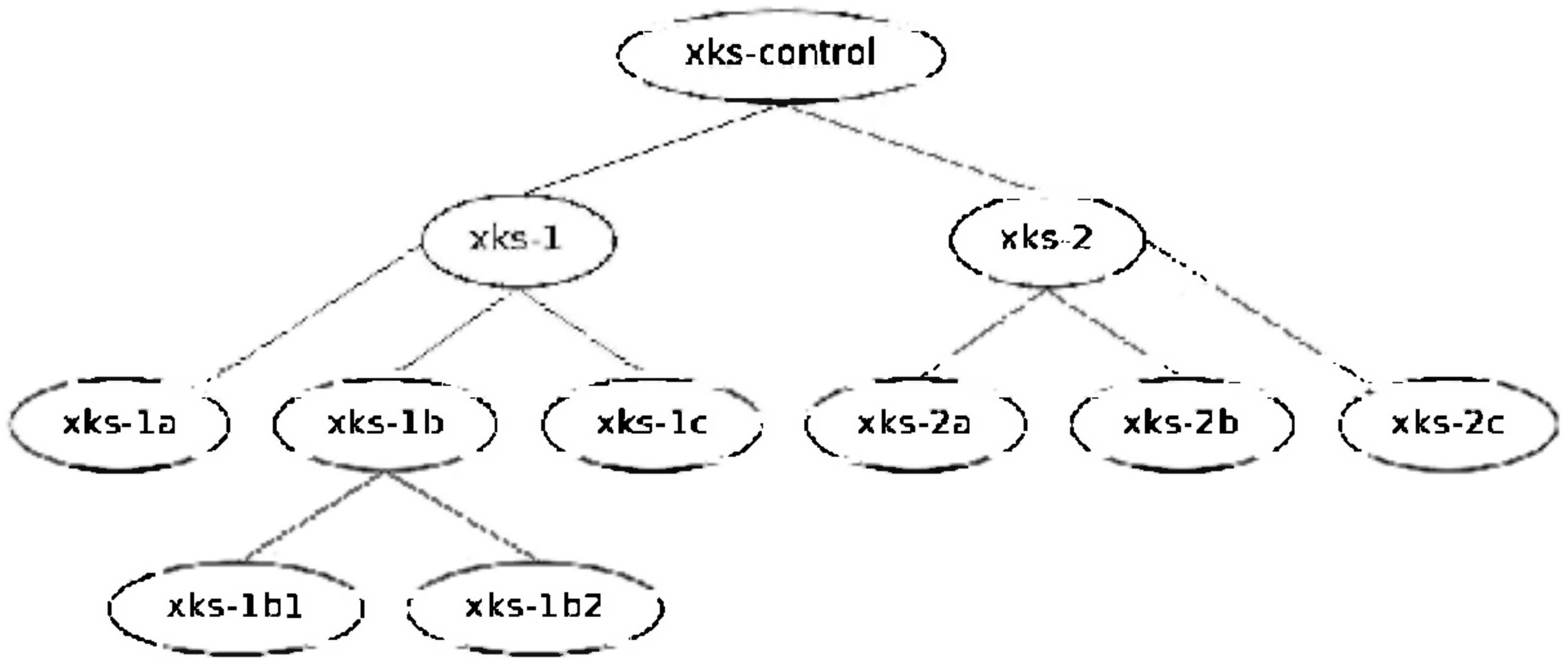
Note**:** (U//FOUO) The complete list of packages can be found on pages 12 and 13.

Royale With Cheese Updater
Read Me

(U//FOUO) Overview

(U//FOUO) The Royale with Cheese (RwC) Updater is the mechanism by which XKEYSCORE (XKS) dictionaries are updated. It requires each node in a given network that is being updated to fulfill at least one of two roles: Publisher or Subscriber. A Subscriber refers to any machine that receives dictionary updates. A Publisher refers to any machine that sends updates to Subscribers.

(U//FOUO) In this sample set-up, xks-control acts only as a Publisher and all other nodes act as a Subscriber. Of the Subscriber nodes, xks-1, xks-2, and xks-1b act as both Subscribers and Publishers. They subscribe to the node that is higher in the hierarchy (e.g., xks-1 subscribes to xks-control), but they also publish to nodes that are lower in the hierarchy (e.g., xks-1 publishes to xks-1b).



(U//FOUO) A slave, by default, sets its Master as its Publisher; although each node can be configured to pull from an arbitrary publisher. To manually "link" a Subscriber to a specific Publisher, use the rwc_Publisher option in xks.config (p. 5).

(U//FOUO) Anatomy of an Update

(U//FOUO) Each dictionary update is initiated by the Subscriber (e.g., xks-1 or xks-1b) and proceeds as follows:

1. (U//FOUO) On an hourly basis, the rwc_post_to_pub script sends the Subscriber's current inventory to its Publisher.

2. (U//FOUO) Each file that the Publisher is configured to publish is compared against the same file in the Subscriber's inventory. There are four possible results:

    a. (U//FOUO) The Publisher does not have a file which the Subscriber listed. The Publisher will tell the Subscriber to delete its copy of the file.

    b. (U//FOUO) The Publisher has a file that the Subscriber is missing. The Publisher will send the Subscriber a copy of the file.

    c. (U//FOUO) The Publisher has the file, but the Publisher's version has a different hash than the Subscriber's. This should only occur when the Publisher has a newer version of a file, and the Publisher will send its version of the file to the Subscriber.

    d. (U//FOUO) The Publisher has the file and both hashes are the same. The Subscriber is up to date and the Publisher makes no change to the Subscriber's inventory.

3. (S//REL) The Publisher creates a binary file that contains all files that need to be downloaded to the Subscriber. The binary file is split into some number of *n*-byte chunks. The size of these chunks is determined by the rwc_max_chunk option in xks.config (p. 5).

4. (U//FOUO) Hashes are generated for each chunk. The Publisher then sends a list of the locations (urls) of the chunks (partials) and the hashes to the Subscriber.

5. (U//FOUO) Once the Subscriber receives the URL list, it downloads each partial file from the Publisher.

    Note: The download of a partial must fail 10 times before the update stops.

6. (U//FOUO) When each download is finished, the Subscriber checks the hash of the file it has with the hash sent by the Publisher. If any of the partial hashes do not meet the expected value, the update stops. If the download completes successfully, the Subscriber recombines the partial chunks into the original binary file.

    Note: (U//FOUO) When the post-to-pub script runs (Step 1), it first checks to make sure it is not already in the middle of a download. If, for some reason, the update could not complete the previous download, it will restart from the partial file at which it stopped. This functionality was implemented to help some slower sites that were having difficulty completing the download of a full dictionary.

7. (U//FOUO) The Subscriber installs the shipment and updates its inventory.

    Note: (S//REL) The current version of each file can be updated on a Publisher by running the rwc_publish_packages.py script that re-computes the hashes for each file for which the Publisher generates updates. By default, this happens every hour on the half hour.

(S//REL) The xks.config File

(U//FOUO) The xks.config.in file includes basic information for Publishers and Subscribers to communicate and update successfully. This includes:

- (U//FOUO) *rwc_max_chunk:* By default, update packages are sent to sites in 5MB chunks. Because connection speeds at some sites can be very slow, it is sometimes necessary to break the packages into smaller chunks. This option allows each Subscriber to customize the chunk size to suit its site's speed constraints. There is no upper- or lower limit to the chunk size; although sizes to one extreme or the other can present other download problems.

  (U//FOUO) Within the xks.config file, simply enter the maximum allowable size in bytes. For example, to reset the maximum chunk from the default 5000000 to 1.2 MB, you might type:

  ```
  rwc_max_chunk=1200000
  ```

- (U//FOUO) *rwc_publisher:* This is the host name or IP from which the dictionary updates are being pulled. This value must be set on Masters and on Control, but it is automatically set on the slaves when *xks setup rwc* (*xks setup royale_with_cheese* in version 1.5.8) is first run. The xks setup script sets up directories and variables that are used internally. To direct a Subscriber to a Publisher located at IP address xxx.xxx.xxx.xxx, type:

  ```
  rwc_publisher=xxx.xxx.xxx.xxx
  ```

  (U//FOUO) To direct a Subscriber to a Publisher located at a hostname, type:

  ```
  rwc_publisher=hostname.
  ```

  Note: (U//FOUO) It is highly recommended that you identify Proxies or Control by IP rather than by host name.

(S//REL) The xks.rwc.config.in File

(S//REL) The xks.config.in file includes the default */true///false/* values for all XKS dictionary configuration options. It is located in:

```
$XSCORE_DIR/config/
```

(S//REL) An editable copy of the file, xks.rwc.config, is located in the same directory. Values set in xks.rwc.config will override those in xks.rwc.config.in. If, however, you encounter problems with xks.rwc.config, then you can delete the file and start over with a copy of xks.rwc.config.in.

(S//REL) The xks.rwc.config File

(S//REL) The xks.rwc.config file includes the custom configuration for a Publisher's cluster. RwC will perform all of its publishing based on a combination of xks.rwc.config.in and xks.rwc.config. Any option that is not set in xks.rwc.config will automatically use the default configuration from the xks.rwc.config.in.

(S//REL) Whether a Publisher or a Subscriber, all machines have their Publisher and Subscriber options default to *true.* On Subscribers that appear at the bottom of the hierarchy (e.g., xks-1b2), all Publisher options are ignored simply because there are no machines lower in the hierarchy. Similar, Control has no Publishers. Its Subscriber options are ignored.

(U//FOUO) Publishers such as Masters and Proxies, however, can be both Subscribers to other Publishers AND Publishers to Subscribers that are below them in the hierarchy. For example, xks-1, xks-2 and xks-1b are each Publishers and Subscribers. Each uses both Publisher and Subscriber options.

(U//FOUO) *Publisher Options*

(U//FOUO) Publishers use xks.rwc.config to publish, add and remove files and directories. *Remove* functions are always run last after evaluating all other options:

(U//FOUO) *Publish*
(U//FOUO) *pub_xx:* Any option in the config file that begins with "pub" indicates that the current machine can act as a Publisher for the given package, *xx.* For example, pub_metadata_forwarder is an option that determines if you want your slave machines to receive metadata forwarder packages. By setting Publisher's value to *false*, pub_metadata_forwarder = false, no machine that subscribes to this option will receive the associated files.

(U//FOUO) *Remove*
(U//FOUO) The *remove* option flags any files or directories that should be removed from the Subscriber. For example, suppose the Publisher is set to *not publish* the forwarding whitelist but the whitelist is part of the default dictionaries which are automatically included as part of an upgrade. In this case, in spite of the fact that the whitelist is in neither the Publisher's nor the Subscriber's inventory, you will need to remove it.

(U//FOUO) It is important to note that the *remove* options have relative paths from $XSCORE_DIR. Remove options include:

rm_file [#]= filename: During the next update, this removes a particular file from all Subscribers. In this, example, the forwarding whitelist is deleted.

Example: rm_file[0] = config/dictionaries/appid_authorized/forwarding_whitelist.cfg *rm_file[#] = regex*: During the next update, this removes from all Subscribers any files that fit the regular expression.

Example: rm_file[1] = config/dictionaries/appid_authorized/.*\.appid

rm_dir[#] = directory: During the next update, on all Subscribers, this removes the directory (in this case, ip_lc_partials) and the entire directory tree within that directory .

Example: rm_dir[0] = config/dictionaries/ip_lc_partials

(U//FOUO) *Add*

(S//REL) The *add* option identifies new files and directories that should be added to all Subscribers. As with the *remove* options, the *add* options also have relative paths from $XSCORE_DIR.

IMPORTANT: (U//FOUO) A classification string must be appended to all *add* options and must match the format used in the xks.config (e.g., comma-delimited). See the classification string at the end of each example.

(U//FOUO) Each file that is added will be bundled into an add_local module. The add_local package automatically assumes the classification of the highest classified file it contains.

add_file[#] = filename: This adds a single file to the add_local module.

Example: add_file[0] = src="config/dictionaries/appid_authorized/forwarding1.txt", classification="TS,SI,REL_FVEY”

add_regex[#] = regex: This adds all files that fit the regular expression to the add_local module.

Example: add_regex[0] = src="config/misc/.*\.txt", classification="TS,SI,REL_FVEY"

add_dir [#]= directory: For the specified directory, this creates a module that includes the entire directory tree. The module assumes the name of the last folder along the relative path. In this case, the module is called my_exampie_directory/.

Example: add_dir[0] = src="config/dictionaries/my_example_directory", classification="S,SI,REL_FVEY"

Note: (U//FOUO) The add_dir option assumes that every file in the directory has the same classification as specified for the directory itself.

(U//FOUO) *Subscriber Options:*

(S//REL) Subscribers cannot add, remove nor publish modules up the chain in the hierarchy. However the *ignore* and *command* options can be used to tailor the Subscriber to accommodate its own unique scenario.

Sub_xx: Any option in xks_rwc.config that begins with "sub" indicates whether the current machine will or will not accept a given package, xx, from its Publisher. For example, sub_metadata_forwarder is an option that determines if you want the Subscriber to accept the forward_metadata package. If a particular Subscriber's proxy value is set to *false*, sub_metadata_forwarder = false, then that Subscriber will not receive the associated files. Even if the corresponding option on the Publisher is set to *true*, no modules will be delivered if the option on the Subscriber is set to *false*.

Important: (U//FOUO) Subscriber settings are unique to the cluster on which they reside.

Sub_New_Modules: If set to *true* (the default), then the site automatically receives all new modules during a given update. The xks.rwc.config file is updated with the appropriate Publisher/Subscriber options and the new modules are downloaded. If Sub_New_Modules is set to *false*, then it will only update the .ini file with the names of the new modules, but it will not start downloading the new modules.

notify_email{name@provider}: When Sub_New_Modules is set to *false*, it prevents the Publisher from adding new modules. The names of any modules that will not be added to the Subscriber will be sent via e-mail to the designated recipient(s). This alerts site administrators of the availability of new modules and affords them the opportunity to evaluate the modules in a safe environment before allowing them to be downloaded.

ignore_file[#]= filename or regex: This tells the Publisher that this specific Subscriber does not want updates for specific files, even if they are part of the latest update. If the Publisher has a different version of the file, it will not send the new file.

(U//FOUO) This setting ignores *all* changes to the specified files, including files that are slated for removal by the Publisher. In fact, under normal configuration, if a file is removed on the Publisher, then that file will also be deleted on the Subscriber. However, if the Subscriber is set to ignore changes to that file, then it will not be removed. Here, the Subscriber does not want updates to forwarding_whitelist.cfg, appid_authorized.appid or any .txt files in the misc directory.

Example: ignore_file[0] = config/dictionaries/appid_authorized/forwarding_whitelist.cfg
ignore_file[1] = config/dictionaries/appid_authorized/appid_authorized.appid
ignore_file[2] = config/misc/.*\.txt

ignore_dir[#]= directory: Similar to ignore_file[#], this option applies to all files in a given directory.

Examples: ignore_dir[0] = config/dictionaries/ip_lc_partials

Command [#]: All bash commands specified within this option are executed only *after* the update is complete. Each command executes a bash script and there are no limitations to what can be written in a bash script. Take great care to avoid potentially disastrous consequences.

Examples:

```
command[0] = source /opt/xkeyscore/bashrc && xks proc restart pd
command[1] = rm /home/oper/*
```

(U//FOUO) Kicking Off An Update

(U//FOUO) The latest version of the RwC scripts is downloaded when the rwc_post_to_pub script is run. (This is what kicks off an update request.) The post_to_pub script downloads a tar file containing the latest version of the RWC scripts and installs the latest scripts before running the rest of the update.

(U//FOUO) IP Geo Updates

(U//FOUO) IP geo updates are distributed daily to sites that receive updates. Updates are processed differently based on the protocol used by given node.

(U//FOUO) *IPv4 Protocol*

(U//FOUO) *Server Side (control)*

1. Control downloads the latest IP table once per day.
2. A script splits the resulting plaintext file into 23 partiais (pieces) of roughly equal size. Each partial has roughly the same number of entries and corresponds to an hour of the day.
3. If the package for IPv4 table updates (ip_lc_partials) on the Publisher machine is different than those on the Subscriber machine, it will package and send at most one partial per hour to every client receiving updates.

(U//FOUO) *Client Side (deployments)*

1. Once per day, the client checks daily to see if it has a full set of 23 partiais. If not, no action is taken.
2. The client concatenates all 23 partiais into a single file.
3. The tables are recompiled and placed in $XSCORE_DIR/config/dictionaries/ip_lc_trie.

(U//FOUO) *IPv6*

(U//FOUO) *Server Side (control)*

1. Control downloads the latest ipv6 table.

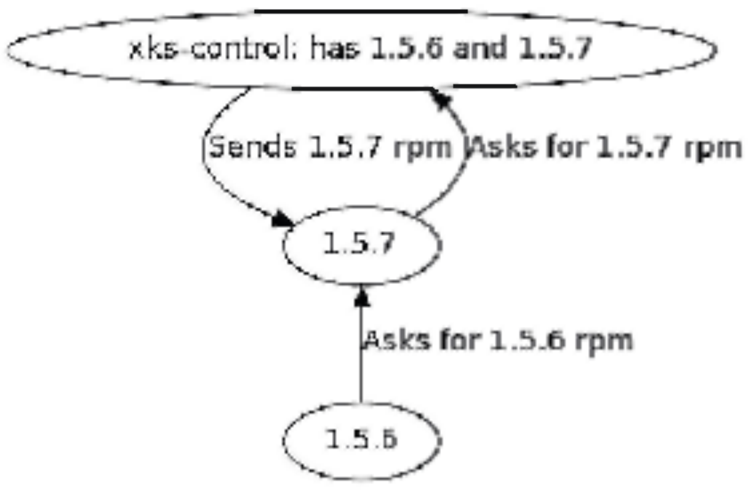
(U//FOUO) *Client Side (deployments)*

1. The client downloads the new IP file.
2. The Client compiles and installs the IP geo file.

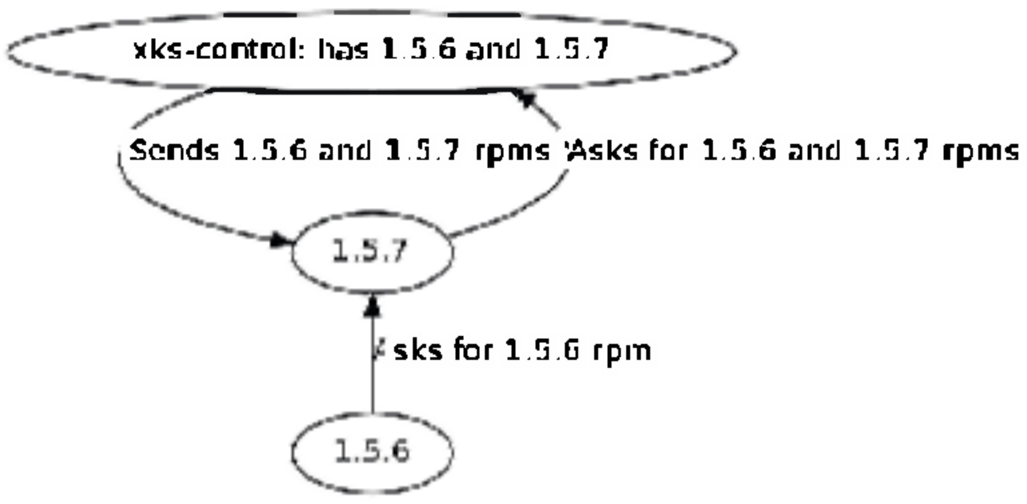
(U//FOUO) Starproc

(U//FOUO) Since starproc is compiled specifically for certain version/platform combinations of XKS, it is imperative that we send the correct version of the rpms to each machine. Sending the rpm is a simple feat when both the Publisher and Subscriber are of the same version and platform, but that's not always the case.

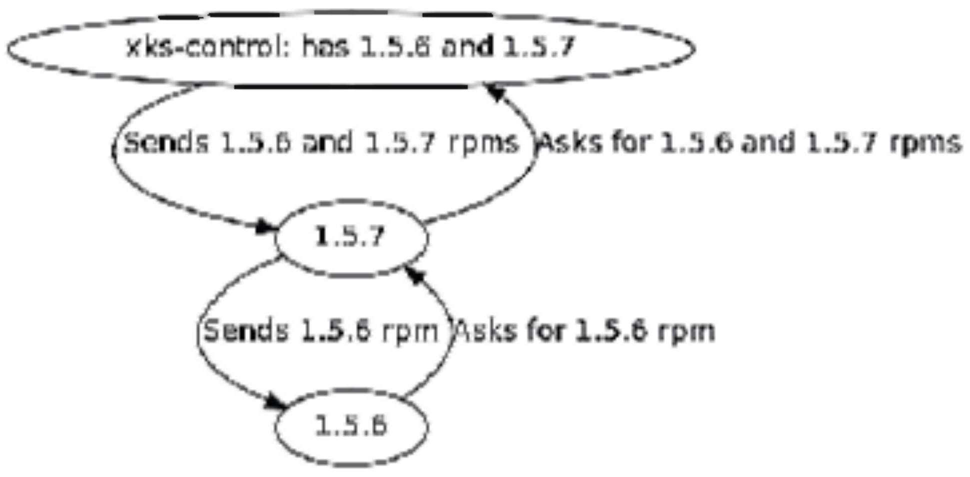
(U//FOUO) In the image below, there is a 1.5.6 machine that subscribes to a 1.5.7 machine. During the first update, the 1.5.7 machine asks control for a 1.5.7 rpm for itself. Control obliges. The 1.5.6 machine asks the 1.5.7 machine for an update, but the 1.5.7 machine doesn't have any 1.5.6 rpms. It sends nothing, but the 1.5.7 machine does add a listing for the correct version of the 1.5.6 rpm to its own inventory.



(U//FOUO) On the next update, the 1.5.6 machine asks for an update once more, but still will not receive it. Unlike the previous update, the 1.5.7 machine asks control for both a 1.5.7 rpm and a 1.5.6 rpm. Control sends both rpms only to the 1.5.7 Subscriber.


xks-control: has 1.5.6 and 1.5.7
Sends 1.5.6 and 1.5.7 rpms
Asks for 1.5.6 and 1.5.7 rpms
1.5.7
Asks for 1.5.6 rpm
1.5.6

(U//FOUO) On the third update, again the 1.5.6 machine asks for the 1.5.6 rpm. This time, the 1.5.7 machine *does* have a copy of the 1.5.6 rpm and sends the rpm down to its 1.5.6 Subscriber.


xks-control: has 1.5.6 and 1.5.7
Sends 1.5.6 and 1.5.7 rpms
Asks for 1.5.6 and 1.5.7 rpms
1.5.7
Sends 1.5.6 rpm
Asks for 1.5.6 rpm
1.5.6

(U//FOUO) When the rpm is delivered to a machine, the Updater also invokes a special install procedure. The Updater will not only install the new version of the rpm, but it will also use a saferestart on that machine's process data.

(S//REL) RwC Packages

(U//FOUO) The following table identifies all Publisher/Subscriber packages available for download as of 9/10/2012. As new packages become available, they may be added using the *odd_file, odd_regex*, and *odd_dir* options.

(S//REL)

Package	Description
alert_tt	The latest *alert_tt* file used for Traffic Thief alerts.
appid_authorized	The contents of the *config/dictionaries/appid_authorized* directory. These files enable content forwarding for the specified appids.
appids	The appids that are 3p/nocomint.
appids_extra	The appids that are 3p/comint.
appids_2p	The appids that are classified S//REL.
appids_2p_extra	The appids that are classified S//SI/REL.
appids_2p_ts	The appids that are classified TS//REL.
appids_2p_ts_extra	The appids that are classified TS//SI/REL.
appid_tables	All appid tables that are classified as TS but not SI.
appid_tables_extra	All appid_tables that are classified TS//SI.
cadence	The contents of the *config/dictionaries/cadence* directory, which defines the permutation rules for cadence dictionaries.
correlation	The contents of the *config/correlation* directory, which contains various configuration files and rules for the correlation engine.
fallen_oracle	The VoIP normalization rules from FALLENORACLE.
fileids	The contents of the *config/dictionaries/fileids directory*. These update the fileid definitions for probable file types.
filter	The contents of the *config/dictionaries/filter/* directory. Files in this directory define terms that will defeat sessions before processing occurs.
geo_info	Anything in the *config/misc* directory that begins with "geo_info_poi." By latitude and longitude, these files list various points of interest, such as cities, airports and embassies.
inquiry	The inquiry hotlist file for strong selectors.
ip_filter	The *config/misc/ip_lookup_lookup_file* file. (This is not a typo.)

(S//REL)

(S//REL)

Package	Description
ip_lc_partials	The 23 partial ipv4 geo table partials created from the NKB IP table.
ipv6	The *ipv6 geo* info table from the Network Knowledge Base.
metadata_forwarder	The default xml files used by the *metadata_forwarder* event processor. These files are in *config/dictionaries/metadata_forwarder/* directory.
misc	Anything in *config/misc* which does not fit into other packages. These files include, but are not limited to, login and phone number extractors, the *generic_wireshark.xml* and the Wireshark parser script.
monitoring_rules	The *monitoring_rules.xml* in the *config/misc* directory.
obfuscation_scanner	The contents of *config/dictionaries/obfuscation_scanner*. These are xml files used by the obfuscation scanner event plugin.
permute	The contents of *config/dictionaries/permute*. These directories contain python scripts which create permutations of selectors based on the realm of the selector (e.g. *youtube.py* creates permutations that will match YouTube traffic).
promotion_rules	The contents of *config/dictionaries/promote_3*. These files include the latest rule files used by the promoter. If a site uses customized rule files, they should probably unsubscribe to this module.
snort	The contents of *config/dictionaries/snort*. These files contain the latest detection rules for Snort that are checked into the XKEYSCORE repository.
stats	The contents of the *config/stats* directory. This directory contains the *remote_jobs* file, which is used to pull statistics from systems that send them home.
vbulletin	The vbulletin. Cf file in the *config/webproc/* directory. This lists selectors for message boards and users.
virus_scanner	The latest virus scanner definitions. The virus scanner files must exist for a user to download sessions from within your gui.
voip	The contents of the *config/dictionaries/voip* directory. The directory contains voip python scripts, xkdb tasking files and the default rules file.
voip_extra	The voip files contained in *config/misc*. The directory includes the *voip_setup* and *sip_generic_parser* xml files.
starproc-2p-xscore_version-platform	The starproc rpm for the given combination of the major version of XKEYSCORE (1.5.7, 1.5.8, etc.) and its platform. (As of 9/10/2012, AS5-64 is the only one for which we currently build rpms.)

(S//REL)