(TS//SCI) Communications System
How To Guide

The communications system (Comms) is the primary means that XKEYSCORE systems use to communicate among servers in the same cluster as well as with the outside world. The hub for the communications system is *xks_comms_server.* Nearly all communications in XKEYSCORE operate via the Comms system. Separate "service" processes within the system perform specialized tasks such as query execution, statistics gathering, and dictionary updates.

Comms makes it easy to configure and to write new services. More important, however, Comms is much more scalable than is the sql-based infrastructure. For example, a single "supermaster" communications server should be able to handle throughput of all communications for an entire site, including those with multiple deep dives.

The Comms system is identified in the *xks proc* file as “xks_comms_server” or “xcs.” It can be found in XKEYSCORE versions 1.5.8 and higher.

Firewall Holes

It is important to note that the Comms system needs a port visible to its "next hop" towards xks-central / xks-control. By default, the Comms system uses port 2412 to communicate with "peers." A *peer* is a communications server residing on a different cluster, proxy, or viewer. Connection paths among *peers* and xks-central can go in any direction (central->proxy->site, central<-proxy->site, site->proxy<-central, etc.), but an installation can only listen on one port. So if you chose port 2412 for *central*, then anyone's connection TO central will need to connect on 2412.

> ***Note:*** The system does support a mode where the Comms system can "share" a port with Apache HTTPS (443, which is typically already open).

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

Set-Up

(U//FOUO) Setup and configuration of the XKS Comms system take just a few steps:

Step 1: Log on as the user `oper`.

Step 2: At the command prompt, type:

```
cd $XSCORE_DIR/config/comms
```

and press `Enter` to get to the Comms directory.

Step 3: From the Comms directory, type `vi comms.config` and then press `Enter`. The Comms.config file will open.

The Comms system uses *Comms.config* to handle all configurations for "talking" among the site's overlord, masters and the slaves. The default values for most configurations (see page 5, Communications Configuration Table) should NOT be altered unless absolutely necessary. For every connection between two peers, there are only a few rules that MUST be configured manually:

- a reciprocal *`peer`* rule on the other end of the connection.
- an *`allow`* rule on one end of the connection.
- a `bandwidth_rule` on each end of the connection. The name of the rule on each end does not matter, and bandwidth caps can be different in each direction if the connection speed is different.

These rules are described in Steps 4 through 6.

Step 4: Change the *`peer`* configuration.

Exomple:

```
peer[01]=address=xks-central.corp.nsa.ic.gov, port=2412,
bandwidth=to_central, network=external
```

A *peer* is a communications server residing on a different cluster, proxy, or viewer. *Peer* connections are initiated when the *xks_comms_server* process starts. In an XKS cluster, *peer* connections are made only by the cluster master or cluster overlord. There are four settings:

a. `address=value`: This is the IP address or hostname of the *peer*.

b. `port={Default: 2412}`: This is the port that is open for connections on the given hostname/IP address.

c. `bandwidth=bandwidth_rule:` This is the maximum amount of bandwidth the Comms system will use for all communications between the local server and this *peer*. You can configure this with a rule name that is defined in Setup 6, or you might choose to enter a specific value (in bits) instead.

> ***Note***: To assist in selecting a good bandwidth setting, a bandwidth tester is provided in: *$XSCORE_DIR/bin.shells/comms_bandwidth_test.py*

d. `network={internal/external}:` The Comms system uses this parameter to establish an "inside" and an "outside" when configuring proxies and making routing decisions. For example, a site connecting to a particular proxy would have a *peer* rule of `external` TO the proxy, and the proxy would have an *`allow`* rule of `internal` FROM the site (Step 5). Similarly, anything flowing FROM the proxy toward xks-central would have a *peer* rule of `external`. If not specified, the default is “external”. Less common is an `internal` *peer* rule such as a country level proxy. For example, in a USA -> GBR proxy, everything on the USA side is internal. Everything else is external.

Step 5: Change the *`allow`* configuration.

Example:

```
allow[01]= subnet=192.168.1.0/24,
bandwidth={some bandwidth name},network=internal
```

The Comms system will only accept connections from address ranges it has been specifically configured to *allow*. If a proxy/viewer is going to receive incoming connections from a *peer*, then an *`allow`* rule for each connection must be entered in *comms.config*. There are three settings.

a. `subnet=value:` This can be a hostname, IP address, or a subnet in CIDR notation. The default setting is XXX.

b. `bandwidth={default: bandwidth_rule}:` As with the *peer* configuration, this is the maximum amount of bandwidth the Comms system will use for all communications between the local server and the *peer*. You can configure this with a rule name that is defined in Setup 6, or you might choose to enter a specific value in bits.

> ***Note***: To assist in selecting a good bandwidth setting, a bandwidth tester is provided in: *$XSCORE_DIR/bin.shells/comms_bandwidth_test.py*

c. `network={default: internal}`: The Comms system uses this parameter to establish an "inside" and an "outside" when configuring proxies and making routing decisions. For example, a site connecting to a particular proxy would have a *peer* rule of `external` TO the proxy, and the proxy would have an *`allow`* rule of `internal` FROM the site. Refer to Step 4d for more information about the `network` rule.

Step 6: Change the *bondwidth_rule* configuration.

Example:

```
bandwidth_rule[to_central] = 1500k
```

The Comms system fairly balances bandwidth among all its services. This prevents one service from using up all the bandwidth and effectively "blocking" another service. To ensure that all services will work, it is *very* important to tell the Comms system how much bandwidth is available. Identifying a value that accurately reflects the amount of bandwidth available on the connection is paramount in making the system work efficiently and correctly.

> ***Note:*** Each physical connection should have its own bandwidth rule. Any "virtual" connections that use that physical connection also should share the same rule name, which also causes them to share the same pool.

Each bandwidth rule has two components:

- *rule name* - The unique name assigned to the *bondwidth_rule.* This is the name that might be referred to in the bandwidth part of the *peer* and *allow* configurations.
- *unit size/measure* – The bandwidth size. Bandwidths are measured in Bps (*b*), Kbps (*k*), Mbps (*m*), Gbps (*g*). A unit (i.e., *g, m, b, or k*) is required for all bandwidth values.

> ***Note:*** the unit size is always in [UNIT] bits per second, never bytes.

In the example, `to_central` is the rule name and `1500k` is the maximum bandwidth (in kiloBITS, *k*) that can be allocated to all services on the entire connection. Typically, if the physical connection between two nodes is symmetrical, then a *peer* rule and its corresponding *allow* rule will share the same numeric bandwidth limit.

If you do not want the nodes to share the named pool (e.g., `to_central`), then you can always use different names. Alternatively, you do not have to use names at all. Instead, you might put the bandwidth size limit directly in the `bandwidth=` part of the *peer/client* definition.

> ***Note:*** On rare occasions, the connection between two nodes is asymmetrical, where the bandwidth for "upload" is different from the bandwidth for "download." On these occasions, each side of the connection will have a different numerical bandwidth.

Typically a site has one limited pipe it uses to talk to the world, so there should be one bandwidth rule (perhaps named "world") and all *peer* and *allow* rules should use that rule. If multiple *peer* and *allow* rules use the same bandwidth rule, they also will share the same cap.

Step 7: Change any other configurations as presented in Communications Configurations Table only as advised by your Network Administrator.

Step 8: Type `:wq!` and press `Enter` to save and exit *comms.config*.

Communications Configuration Table

`peer port = {default:2412}`	Sets the port for the communications server to listen for connections from other communications servers (*peers*).
`client port = {default:2411}`	Sets the port for the communications server to listen for connections from other clients.
`is_xks_site = {default:true}`	Instructs the communications server to behave like an XKEYSCORE site and to pull some configurations from xks.config.

Proxy Viewer Instance Names

Similar to DNS in an IP network, proxies and viewers need to specify a "name" that they will be known as to the rest of the world. This can be any name, but it is best to come up with a name that “makes sense” (e.g., xks-central or nsa-central). Proxies/Viewers can have more than 1 name, but it is best to pick just one or two to avoid confusion later.

`instance_name[] = PLACEHOLDER`	Sets a human readable name for a proxy/viewer. If this rule is left empty, it will default to sigad for an XKS site and to the hostname for all other installations. At load time, these names will be replaced automatically with the correct name. ***Important:*** Instance names are shared with all *peers* that are directly connected to this server. Therefore, DO NOT directly connect *peers* to which the CLASSIFICATION and INSTANCE NAMES of THIS server are not releasable.

Proxy Viewer Classification

There are two types of proxies:

- "transparent" proxies which are basically just a hop towards a site and don't do filtering.
- country-level proxies which do filtering and sanitization.

Every proxy and server has a classification. A Comms server’s "clearances" reflect the country that owns the server and the classification of messages that are *allowed* to pass through the node.

When *is_xks_site* is `true` and all classification and marking fields (below) are empty, their values will be taken from xks.config. If *is_xks_site* is `false`, there is no default and this value must be supplied in the comm.config file.

`clearances =`	Sets the clearances tag list of the current server and is presented in the form TS,SI,CTZN_USA. Note that the `CTZN_` tag indicates the country that owns the server. The `CTZN_` tag is what gets checked against the REL tags on a piece of data. ***Note:*** There can be many owners for a piece of data, but only one CTZN tag for a piece of hardware.

Proxy Viewer Classification (cont.)	
`internal_clearances = {default: clearance tag value}`	Sets the clearance tags of the "internal" network on proxies. It is not necessary to set internal values on Non-proxy servers.
`external_clearances = {default: clearance tag value}`	Sets the clearance tags of the "external" network on proxies. It is not necessary to set external values on non-proxy servers.
`default_marking =`	Sets the default classification of data produced by the server. The value is formatted as an XKS classification tag. For example: `TS,SI,REL_FVEY,OWNER_USA`
`routing classification = {default: default_marking}`	Sets classification of routing messages produced by this server.
MAILORDER	
MAILORDER is also configured as a *peer* but with no address specification and a `direction=` of `in` or `out`. Additional options and their defaults:	
`directory={default:}`	Depending on the direction of the flow of data, this is the location where files are read from or written to. If not specified, this uses the *comms_mailorder_input_directory* entry in xks.config file. If not there, then it defaults to: */export/data/xkeyscore/inputs/comms_mailorder* If not specified for output, then this uses the *mailorder_output_directory* entry in xks.config. If not there, then it defaults to: */export/data/xkeyscore/outputs/mailorder_working*
`destination={default:}`	Specifies the destination of Comms messages that this flow receives. This is required for output.
`timeout={default:600}`	Specifies the maximum amount of time (in seconds) the output file is kept open. This is required for output.
`maxfilesize={default:100000000}`	Specifies the maximum size of the output file. The file will be closed once the file size is met or exceeded. This is required for output.
`pddg={default:}`	Specifies the collection source of Comms messages. If not specified, this uses the pddg entry in xks.config. If not found there, then it defaults to "XX." This is required for output.

MAILORDER (cont.)	
comms_dest_trigraph= {default:}	Specifies the destination the Comms message will be routed to. If not specified, this uses the *comms_dest_trigroph* entry in *xks.config*, If not found there, then it defaults to XKJ. This is required for output.
mailorder priority= {default:}	Sets the priority of all Comms messages being transported using MAILORDER. The highest priority is 1 on a scale of 1 to 5. If not specified, this uses the mailorder priority entry in *xks.config*. If not found there, then it defaults to 2. This is required for output.
source_digraph= {default:}	Sets the source location of the Comms messages. If not specified, this uses the source_digraph entry in *xks.config*. If not found there, then it defaults to XX. This is required for output.
Routing Mode	
Routing mode=proxy	This is used for "security" proxies. No routing is possible through this node. Communications through this node are “NATed” in both directions.
Routing mode=endpoint	This is the default for XKS site installations. Routing is possible among the node's internal interfaces and among the node's internal and external interfaces. It is NOT possible among the node's external connections.
Routing_mode=midpoint	Routing is among any/all *peer*s connected to this node.
Communications Proxy	
Connections to the communications server that do not follow the communications protocol can be proxied to a different IP:Port instead of being closed. This is useful if you want to use the communications server on a port usually used by another service, such as HTTP or HTTPS, and want to forward normal connections to the new port used by that service. Note that this only works for protocols in which the client sends the first message. This works for HTTP and HTTPS but not SSH or MySQL.	
unknown_proxy_ip = 127.0.0.1	Defines the proxy IP port.
unknown_proxy_port = 444	Defines the proxy port.

Key Terms

Cluster: A single Master and 0 to *n* Slaves. A system may have front-end and/or back-end clusters. Front-end clusters perform raw packet collection and back-end clusters perform protocol processing.

Master: A single machine that runs the XKEYSCORE software and distributes the configuration to all Slaves in its cluster. At a site with multiple systems and an Overlord, the Master receives its configuration from its Overlord.

Overlord: A single machine that runs the XKEYSCORE software and controls the clusters in a complex system. It passes configuration files to the individual Masters.

Site: A single SIGINT Activity Designator (SIGAD). A site may contain 1 to *n* systems.

Slave: A single machine running the XKEYSCORE software that receives its configuration from its cluster Master.

System: One to *n* clusters and 0 or 1 overlord.