CSI Security

Nokia Siemens Networks
Security

Certificate Authority for Network Elements

Description of OAM Counters and Alarms

Version 2.0

Table of Content

1. Introduction

1.1 Executive Summary

Nokia Siemens Networks provides a cost-effective and easy-to-use Public Key Infrastructure (PKI) solution for issuing and managing digital certificates. The management of certificates is automated and compliant to standards.

The described Public Key Infrastructure is based on the INSTA DefSec Oy Certifier® software product. INSTA DefSec is part of the INSTA group (http://www.insta.fi/en) and has signed official and exclusive partnership with NSN, for joint development of a TelCo environment capable PKI.

To reduce OPEX of the solution, one of the first steps taken jointly is the Integration of INSTA certifier into common TelCo Operator Network Management. The current solution release provides such and the details needed for the integration are described in this document.

1.2 Scope of this document

NSN PKI solution is based on SUN/Oracle Netra X4270 hardware, running RedHat Enterprise Linux 6.1 ES and INSTA Certifier 5.0 on top. Together with a customized HA architecture, this Platform is called HAPF2.0.

All information of this document is based on those prerequisites and will be maintained by NSN CSI Security.

2. OAM architecture

In TelCo environments, SNMP is very commonly used for the following purposes:

- Fault Management (FM): Sending of spontaneous event notifications (traps) by the Managed Element (i.e. the PKI). In an NMS, the notifications are mapped into stateful Alarms or are used to control the status of active Alarms on the managed Object.

- Performance Management (PM): Providing of statistical data by the Managed Element in shape of so-called PM counters. Counters are usually polled by the NMS and stored there over a period.

Since the mapping of notification or analysis of counters is completely done by the NMS, other interpretations of each information but are possible. It is for example common, that an NMS polls the Managed Elements in high frequency (e.g. 1/min) and triggers Alarm conditions in the NMS, if a PM counter goes out of defined thresholds.

In order to manage the complete INSTA Certifier solution, it is necessary to supervise both its components, OS and application:

- RedHat Operating System is supervised using the rpm for net-snmp 5.5 of the RedHat package repository (el6). This includes sending of spontaneous notifications as well as ability to poll information via SNMP.

- INSTA Certifier Application is supervised by an own snmp agent. This agent registers to the net-snmp master agent via standard agentx configuration directive. It adds support for the INSTA-PKI-MGMT-MIB and according SMI to the existing net-snmp stack and functions. The INSTA-PKI-MGMT-MIB defines PM counters as well as notification formats.

2.1 net-snmp configuration for NSN INSTA Certifier

In order to supervise OS-managed resources, net-snmp offers a set of directives to be defined in `/etc/snmp/snmpd.conf`. Many directives can be used in different ways to supervise e.g. interfaces, CPU load or process table. Each can be checked with the same configuration directive, but by targeting different things. Also, an event like “Needed process found not running” or “filesystem full” can be defined with or without cancel notifications, with or without defining a corrective action etc …

Besides listing the plain notifications to be expected, it is therefore needed to explain in this chapter, what host resources are monitored and how the system reacts once an event is fired.

2.1.1 Process Table supervision

Process table supervision is realized via internal polling of the hrSWRun Table of host resources MIB. This allows using a "`proc`" directive in `snmpd.conf`. Each time a "`proc`" is found not running, a corresponding (internal, not towards the trap-receiver!) event is fired and `prErrorFlag` of the corresponding process is set. An additional `setEvent` triggers to set also the `prErrFix` flag. This in turn runs the command defined by the '`procfix'` directive and restarts the process in question, in case it was not halted clean.

In order to not only restart a process, but also to inform about the automated process restart, an external notification is sent, the trap is of notification type `mteTriggerFired`.

```
proc crond
procfix crond /usr/sbin/wrap_procfix /etc/init.d/crond
monitor -r ${PT} -e cronDead   -i -o prNames.1 -o prErrMessage.1 "cronProcCheck"  \
prErrorFlag.1 != 0
setEvent          cronDead    prErrFix.1 = 1
notificationEvent cronDead    mteTriggerFired
```

There is one such entry needed for each process to supervise. In this example, entries are indexed and `crond` has index 1. Meaning, another process has such a block defined after with index 2 and the next one would be using index 3. Using wildcards has proven to be tricky on many operating systems throughout `net-snmp` history and seems not recommendable.

All notifications send for failed processes use the same OID, the information which process died is found inside the notification PDU var bindings and according to the index in `prNames`.

2.1.1.1 Notification format

If defined as above, the following notification var-binds are received to identify the event:

1.3.6.1.2.88.2.0.1	Indicates the mteTriggerFired event, hence that any of the defined "monitor" directives has fired
1.3.6.1.4.1.2021.2.1.2.**X**	Is the prNames field from UCD-MIB, describing that the process instance with index X has died (Hence X should be 1 if it is the first proc directive that fires, x=2 for the second etc)
1.3.6.1.4.1.2021.2.1.101.**X**	Is the prErrMessage to the corresponding process entry.

There will be no automatic notification send to indicate a reverse event (no cancel-trap). That means, every time a process has been found dead, regardless if it was cleanly halted or crashed - a notification is fired and ventually resulting Alarms in a 3rd party NMS must be canceled manually.

2.1.2 Filesystem Consumption supervision

Filesystem consumption is controlled with disk directives in the `snmpd.conf`. Notifications are sent if a certain percentage or absolute value of Mbytes is used on the disk. When the situation is cleared again, i.e. filesystem space freed up so that current usage goes under the defined threshold, an automatic cancel event can be sent, but not necessarily.

Depending on the possibilities in the 3rd party management system, disk supervision can be done in 2 ways, either with or without auto-canceling of Alarms. Default configuration in HAPF2.0 is, to trigger a notification when < 30% diskspace are free and to send a cancel notification upon clearing.

2.1.2.1 Without cancel-event notification

A "`disk`" directive for each filesystem to be supervised can directly define a percentage threshold to monitor. Once the threshold is overcome, a simple monitor directive on the `dskErrorFlag` MIB object will fire an external notification without further configurations needed.

```
disk / 30%
monitor -r 30 -o dskPath -o dskErrorMsg triggername  dskErrorFlag !=0
```

In above setup, one notification will be sent if the filesystem "/" has less than 30% free space. Since no cancel event ever has to be sent brings certain advantage in regards to stability: The "status" of the filesystem does not need to be tracked further after the notification is sent. Different than from process supervision, Disk supervision works "wildcarded" with a single monitor directive being sufficient for all disk filesystems. (i.e. you can list several "`disk`" directives but only need one monitor directive without object indexes.

2.1.2.2 Notification format

If defined as above, the following notification var-binds are received to identify the disk related events:

1.3.6.1.2.88.2.0.1	Indicates the mteTriggerFired event, hence that any of the defined "monitor" directives has fired
1.3.6.1.2.88.2.1.1.**X**	Is the mteHotTriggerfield from DISMAN-MIB, describing the name of the instance with index X has fired the notification.
1.3.6.1.2.88.2.1.2.**X**	Is the mteHotTargetName, it is empty if setup like in our example.
1.3.6.1.2.88.2.1.3.**X**	Is the mteHotContextName, it is empty if setup like in our example.
1.3.6.1.2.88.2.1.4.**X**	Is the mteHotOID, a link to the object (i.e. filesystem) that triggered the notification. In our case this is the OID for the UCD-MIB dskErrorFlag.**X**
1.3.6.1.2.88.2.1.5.**X**	Is the mteHotValue, a link to the value (i.e. status) of the object that triggered the notification. In our case this is the value for the UCD-MIB dskErrorFlag.**X**

additional (Those fields are added by the "-o" options in the monitor directive as additional var-bindings in the PDU) :

1.3.6.1.4.1.2021.9.1.2.**X**	Is the dskPath field from UCD-MIB, describing that the filesystem instance with index X is described (Hence X should be 1 if it is the first disk directive that fires, x=2 for the second etc)
1.3.6.1.4.1.2021.9.1.101.X	Is the dskErrMessage to the corresponding disk entry.

2.1.2.3 With cancel-event notification

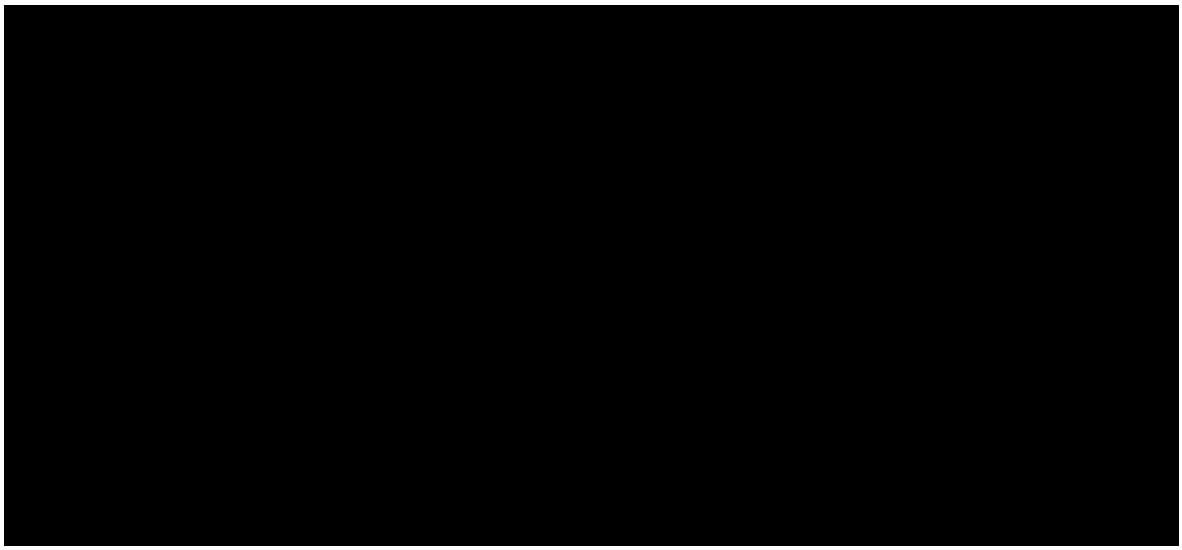
The "`disk`" directives still have to indicate each filesystem to be supervised with a percentage threshold to monitor. Once the threshold is overcome, a simple monitor directive on the `dskErrorFlag` MIB object will fire an external notification without further configurations needed.

```
disk / 30%
monitor -r 30 -e fsFull -o dskPath -o dskErrorMsg triggername dskErrorFlag !=0
monitor -r 30 -e fsCLear -o dskPath triggername dskErrorFlag ==0
notificationEvent fsFull mteTriggerFired
notificationEvent fsClear mteTriggerFired
```

In above setup, one notification will be sent immediately at startup of the agent for each filesystem supervised with a "`disk`" directive. If the filesystem is full or not full, an "`fsFull`" triggered – or "`fsClear`" triggered notification is sent, respectively. Then, no further notifications are sent until the status changes. This way, a 3rd party management system can track the status of filesystems and automatically receive clearance or alert notifications.

2.1.2.4 Notification Format

If defined as above, the following notification var-binds are received to identify the event



2.1.3 Interface supervision

Same as with disk supervision, a monitor directive can be used to send external notifications about interface failure states. If the operational status of any interface becomes different from regular, an external notification is sent. A cancel is sent once the interface comes back and the operational status goes to normal again. There is no additional directive needed, simply the monitors and a notification event to fire up/down traps.

```
notificationEvent  linkUpTrap    linkUp   ifIndex ifAdminStatus ifOperStatus
notificationEvent  linkDownTrap  linkDown ifIndex ifAdminStatus ifOperStatus
monitor -s -r 60 -e linkUpTrap   "Generate linkUp" ifOperStatus != 2
monitor -s -r 60 -e linkDownTrap "Generate linkDown" ifOperStatus == 2
```

2.1.3.1 Notification Format

If defined as above, the following notification var-binds are received to identify the event

1.3.6.1.6.3.1.1.4.1.**X**	Indicates trapOID, which is in this case 1.3.6.1.6.3.1.1.5.4 (IF-MIB linkUp)
1.3.6.1.2.1.2.2.1.1.**X**	indicates the index of the interface in the IF-MIB tree of that host
1.3.6.1.2.1.2.2.1.2.**X**	indicates the description of the interface with Index X (e.g. eth0 or sit)
1.3.6.1.2.1.2.2.1.7.**X**	Indicates the Interface administrative status
1.3.6.1.2.1.2.2.1.8.**X**	Indicates the Interface operational status
1.3.6.1.6.3.1.1.4.3.0	Is an SNMPv2-MIB OID that carries as values an indication of the operating system and snmp agent that is sending the notification

3. Notification Reference (FM)

OS supervision via MIB-II and related repositories is relatively complicated because the MIB structures are quite generic and not as straightforward as in most modern private enterprise MIBs.

Therefore, the INSTA notifications are much easier to handle because each notification OID can directly map into an Alarm or cancel event. While the plain net-snmp based alarms to notify about OS/HW status are based on generic event triggers of the DISMAN-EVENT MIB, which produces various types of notifications, where sometimes PDU text must be evaluated by an NMS, and sometimes notification OIDs can be directly mapped to an Alarm. (For example: If 4 processes are supervised to be alive and any one of them dies, the notification event will always come with the same OID and only the PDU text while unveil, WHICH particular process of the 4 died)

3.1 Recommended net-snmp notifications

Per default, the system is delivered with an `/etc/snmp/snmpd.conf` setup of net-snmp directives as described below. For a better understanding of that please refer to the net-snmp wiki, discussion groups or the net-snmp manpages, e.g. under

http://www.net-snmp.org

resource	varbind text / OID	
auditd	.1.3.6.1.2.1.88.2.0.1 / *"no auditd running"*	A notification is sent if auditd is not running
rsyslogd	.1.3.6.1.2.1.88.2.0.1 / *"no rsyslogd running"*	A notifictaion is sent if rsyslogd is not running
sshd	.1.3.6.1.2.1.88.2.0.1 / *"no sshd running"*	a notification is sent after the last sshd dissapeared (does not differ daemon and sesisons in privilege separated mode)
ntpd	.1.3.6.1.2.1.88.2.0.1 / *"no ntpd running"*	a notiofication is sent if no ntpd is running
slapd	.1.3.6.1.2.1.88.2.0.1 / *"no slapd running"*	a notification is sent if slapd openldap server is not running (FE only)
/	.1.3.6.1.2.1.88.2.0.1 / *"name of filesytem"*	mte trigger fired for filesystem supervision
	.1.3.6.1.2.1.88.2.0.3 / *"name of filesytem"*	mte trigger falling for filesystem supervision
/backup		
/usr/local/certifier		
/usr/local/certifsub		
all LAN interfaces	.1.3.6.1.2.1.88.2.0.1 / "ifup-ifdown notifications"	one varbind per interface included in the trap

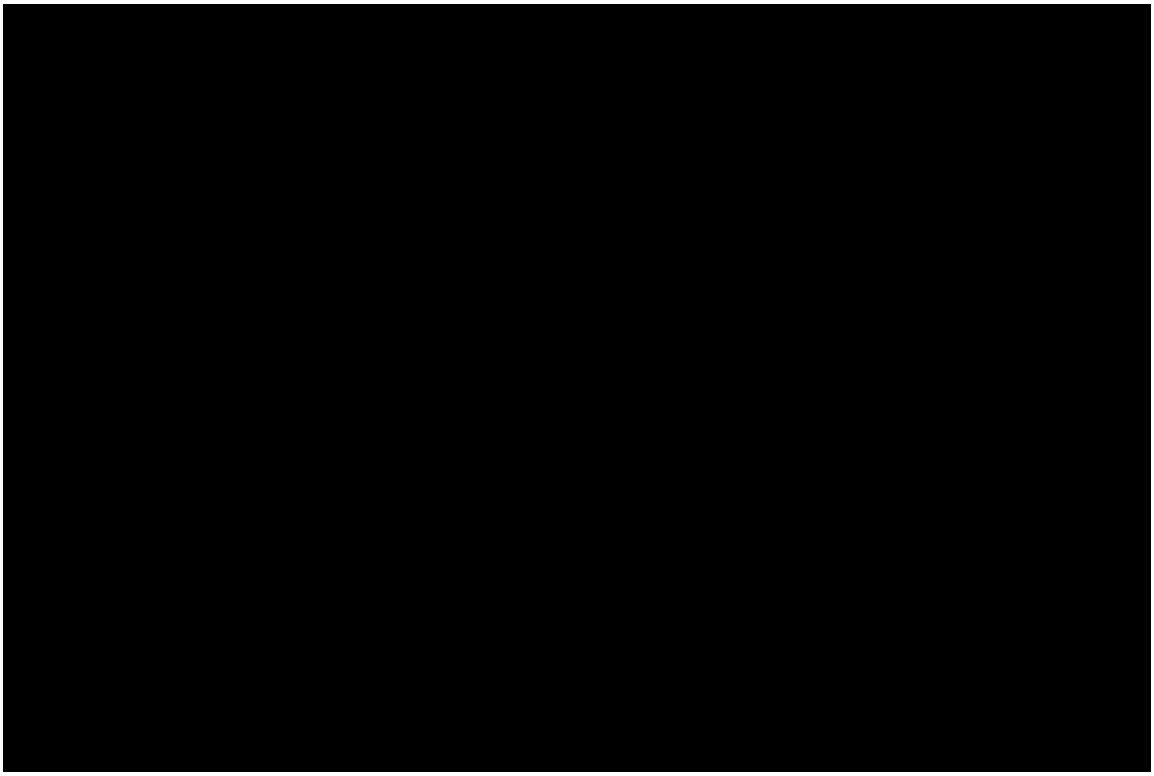
OIDs in the table only identify that an `mteTriggerEvent` has created the notifications!

Please refer to the descriptions in Chapter 2.1 to understand required OID/notification to Alarm mappings in a 3rd party NMS.

Mind, that NSN NetAct supports the PKI „plug and play“, meaning, there is no adaptation or mapping needed. Whereas a 3rd party system like e.g. NetCool or EMC SMARTs will normally not have built in support for the INSTA MIB! Instead, all var-binds need to be mapped manually!

3.2 Available Certifier application subagent notifications

The following events are monitored by the Certifier SNMP subagent and notifications in case an event is fired are sent to the NMS.



3.2.1 Trap conversion to SNMPv2c

HAPF2.0 comes with a built in mechanism to convert the certifier SNMPv3 notifications into the simpler structure of SNMPv2c trap notification PDUs. This feature allows integration of certifier FM also to an NMS without SNMPv3 capability.

Trap conversion is described in detail in the HAPF2.0 Architecture description document and does not require any manual changing of the remaining SNMP configuration. When the platform is installed, conversion can either be enabled and will then always take place for all SNMPv3 notifications sent from Insta Certifier. Or it will be disabled, and the SNMPv3 notifications from certifier are sent directly from the application.

4. Counter Reference (PM)

NSN INSTA Certifier provides support for all MIBs that are supported by net-snmp 5.3 snmpd, plus support for the INSTA-PKIMGMT-MIB via agent registration of the INSTA subagent.

The rich support for MIBs by net-snmp under RedHat allows a variety of OS and HW centric information to be polled by an NMS. It is unfeasible to assume, that all available information would be evaluated by a single operator. Therefore, we describe in the following a subset of the available counters, which are recommended. This subset can be changed, Operators could freely define which MIBs they want to query.

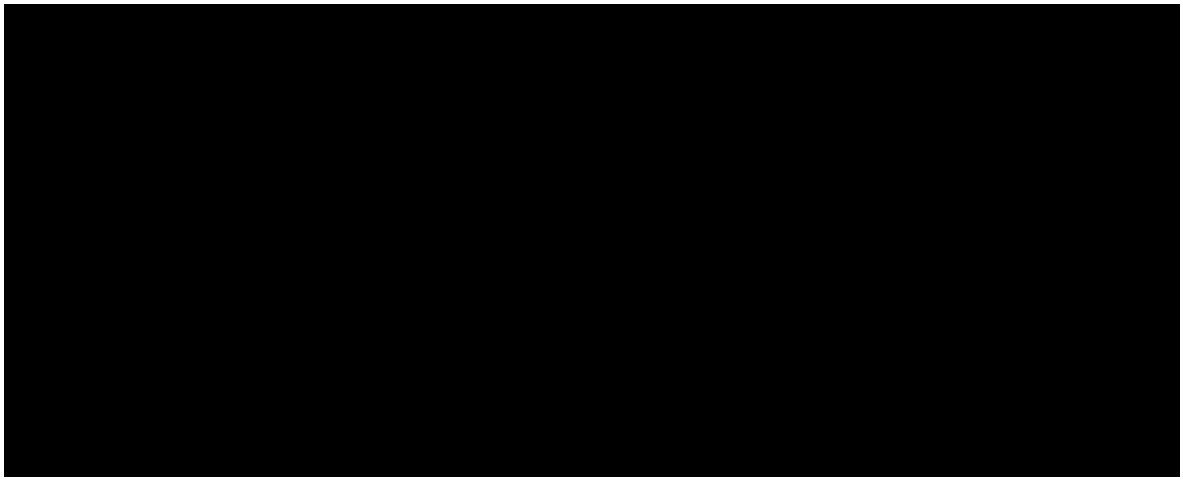
A complete list of all MIBs supported by net-snmp can be found under

http://www.net-snmp.org/docs/README.agent-mibs.html

4.1 Recommended net-snmp counters

As a subset of all SMI supported by net-snmp snmpd under RedHat Linux, NSN recommends to supervise the underlying OS and Hardware with at least the following counters to be polled for evaluation in PM context. Those are in addition to the available Certifier application counters, which should all be polled as well.

Mind the following typographic convention: OIDs that end with a dot "." are wildcarded (meaning there is one Object instance indexed per Object, as one Index for each CPU-core, ethernet interface etc in the Netra server). While OIDs that end without a trailing "." are standalone and already *instanced*. (meaning they do not have an index trailing and are by net-snmp slang "not *wildcarded*")



4.2 Available Certifier application subagent counters

Following counters are available as PKI Objects in the INSTA-PKI-MGMT-MIB. Mind the following typographic convention: OIDs that end with a dot "." are wildcarded Objects (meaning there is one Object instance indexed per CA in the certifier). While OIDs that end without a trailing "." are standalone and already *instanced*. (meaning they do not have a CA as index trailing and are by net-snmp slang "not *wildcarded*")

Object Name	OID	Object counts number of …
ocspCounterTotalRequests	.1.3.6.1.4.1.36878.1.1.1.19	received OCSP requests (total)
ocspCounterGoodRequests	.1.3.6.1.4.1.36878.1.1.1.20	GOOD state OCSP responses (total)
ocspCounterBadRequests	.1.3.6.1.4.1.36878.1.1.1.21	BAD state OCSP responses (total)
caCertificatesEnrolled	.1.3.6.1.4.1.36878.1.1.1.18.1.3.	Certs enrolled under CA
caCertificatesInitial	.1.3.6.1.4.1.36878.1.1.1.18.1.4.	Certs enrolled after EE initialisation
caCertificatesUpdated	.1.3.6.1.4.1.36878.1.1.1.18.1.5.	Certs updated/renewed by CA
caCertificatesExpired	.1.3.6.1.4.1.36878.1.1.1.18.1.6.	Certs that are expired under CA
caCertificatesRevoked	.1.3.6.1.4.1.36878.1.1.1.18.1.7.	Certs revoked by CA
caCertificatesOnHold	.1.3.6.1.4.1.36878.1.1.1.18.1.8.	Certs placed onHold under CA
caRequestsReceived	.1.3.6.1.4.1.36878.1.1.1.18.1.9.	Certification Requests received
caRequestsInitial	.1.3.6.1.4.1.36878.1.1.1.18.1.10.	Initial Certification Requests received
caRequestsUpdate	.1.3.6.1.4.1.36878.1.1.1.18.1.11.	Cert update requests received by CA
caRequestsRejected	.1.3.6.1.4.1.36878.1.1.1.18.1.12.	Cert Requests rejected under CA
caRequestsPending	.1.3.6.1.4.1.36878.1.1.1.18.1.13.	Cert Requests waiting approval
caCrlSize	.1.3.6.1.4.1.36878.1.1.1.18.1.14.	Certs on the CRL of CA

All Certifier System management Information starts at OID 1.3.6.1.4.1.36878.1.1 which is iso.org.dod.internet.private.enterprise.insta.instaMgmt.pkiMgmt.

5. Abbreviations and References

Abbreviation	Description
NMS	Network Management System
PKI	Public Key Infrastructure
FM	Fault Management
PM	Performance Management
CM	Configuration management
IM	Inventory management
MIB	Management Information Base
SMI	System management Information
SNMP	Simple Network management Protocol
OS	Operating System
HW	Hardware
SW	Software
CMP	Certificate Management Protocol
SCEP	Secure Certificate Enrollment Protocol
OCSP	Online Certificate Status Protocol
CA	Certification Authority

Ref	Description
/1	NSN Insta PKI Installation Guideline for HAPF2.0
/2	LTE Transport Security Sizing & Ordering Guideline
/3	Radio Transport Security Solution - Technical Solution Description
/4	PKI Configuration Quick-Start Guide
/5	HAPF FM PM Reference
/6	HAPF20 Admin Notes (Collection of independent Leaflets)