


# SNMPv3

## Secure, Safe and Still Simple

Dr. Michael Schwartzkopff


# Why SNMPv3?

- Several attemps for a SNMPv2 solved different problems.
- SNMPv3 offers a general solution that emphases on security and modularity.
- Only SNMPv3 offers:
  - Encryption,
  - Authentication *and*
  - Authorization.
- SNMPv3 the only valid IETF standard!


# A SNMPv3 Entity

- SNMPv3 does not speak of agents or managers, but knows *entities*.

- The definition of SNMPv3 ist *modular*. So v1 and v2c integrate nicely.
- This modularity allows for future extentions.
  - i.e. new encryption algorithm can be implmented.
- The work is done by so called *applications*. According to the role of the entity different appications work.


# Modular Architecture

SNMP Entity

SNMP Applications

- Command Generator
- Notification Originator
- Proxy Forwarder
- Command Responder
- Notification Receiver

SNMP Engine

- Dispatcher
- Message Processing Subsystem
- Security Subsystem
- Access Control Subsystem


# The Machine

- The *dispatcher* receives and and sends sends the SNMP messages. It passes the data on to the *processing*.

- The modular *message processing* creates or processes mesages. Modules for v1, v2 and v3 exist.
- The *security subsystem* authenticates and/or encryptes messages. A COMMUNITY (v1, v2c) and a *user based security model* (USM, v3) exist.
- Das *acces control subsystem* authorizes access to parts of the MIBs (*view based access control model*, VACM).


# The Applications

- The *command generator* creates `get`, `getnext`, `getbulk` and `set` requests. It also processes the answers. This module works in „managers“.

- The *command resonder* creates the answers to the requests. This module works in „agents“.
- The *notification originator* and *receiver* create or process notifications.
- There is pletny room for additional applicationens.


# The entities

- In v3 *entities* talk to each other.

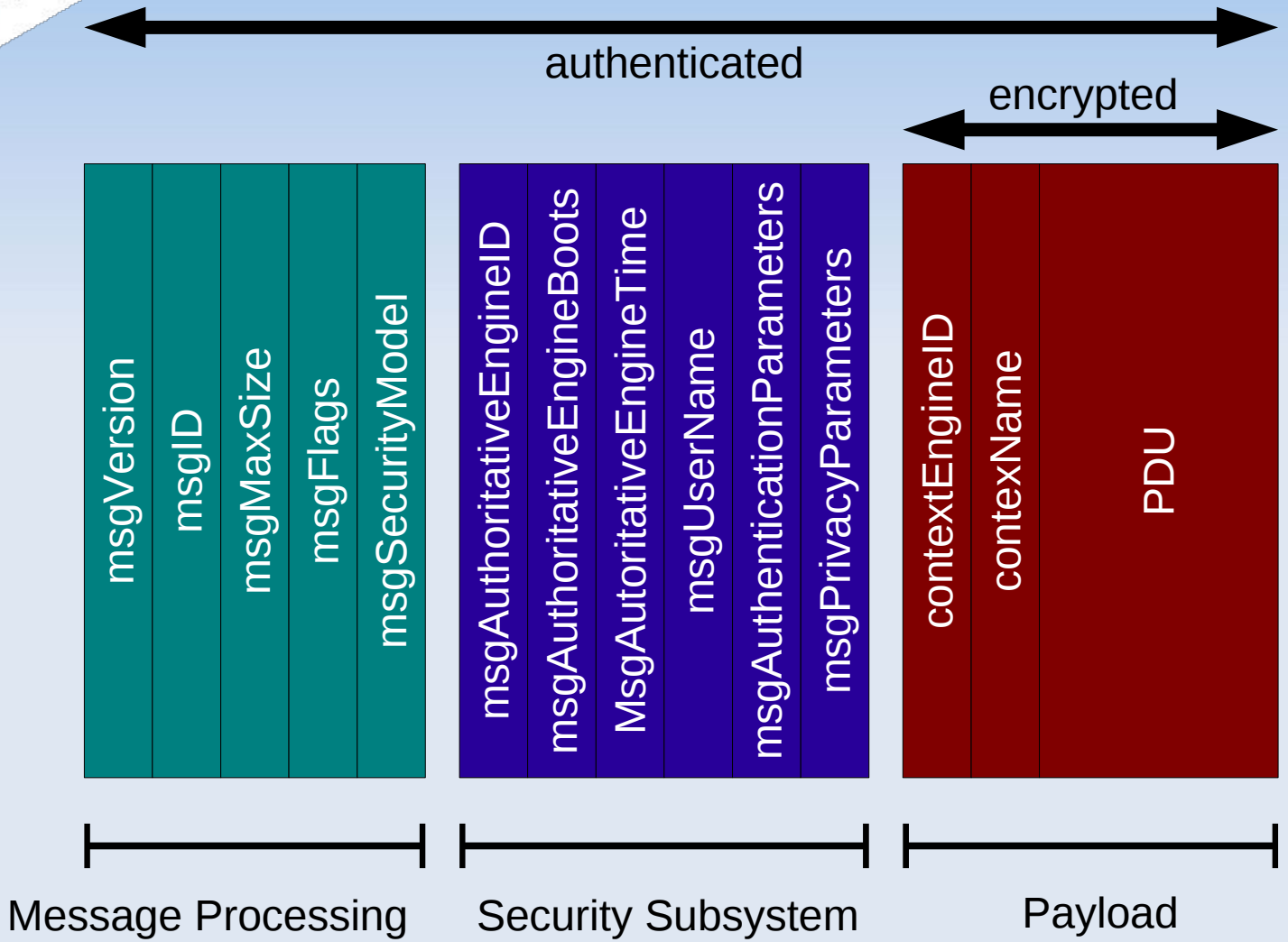
- Each SNMP entity is defined by its `snmpEngineID`.
  - RFC3411 offers some methods calculating an unique ID.
- Every entity calculates it own ID.
- Bevore two entities exchange information the ID of the other has to be known. SNMPv3 defines a autodiscovery process.


# More Items ...

- SNMPv3 defines many (!) new items. Imporant are:

- Username
- Security Level
  from *noAuthNoPriv* to *authPriv*.
- Authentication protocol MD5 or SHA1
- Authentication passphrase
- Privacy Protocol (DES, AES) with passphrase

# The SNMPv3 packet




# 1. Create Users

- Create users (=`secName`), or map communities to a `secName`.

- USM – Option

```
createUser username (MD5|SHA) \
   authpassphrase [DES|AES] privpassphrase
```

Sample:

```
createUser misch MD5 verysecret
```

- v1 and v2c – Option

| *com2sec* | *SECNAME* | *SOURCE* | *COMMUNITY* |
|---|---|---|---|
| z.B.: com2sec | readonly | 192.168.1.0/24 | public |


# 2. Group Users

- Next users are grouped together:

```
group  GROUP     secModel    secName
```

- Sample:

```
group  MyROSystem    v1      readonly
group  MyROGroup     usm     misch
```


# 3. Create Views

- Views define a part of the complete OID tree.

- Different groups have access to different views.
- Views also can be defined exclusive.

```
view   VNAME    TYPE   OID    [MASK]
```

- Sample:

```
view   all    included   .1   80
view   sys    included   .1.3.6.1.2.1.1
view interfaces included 1.3.6.1.2.1.2
```


# 4. Define Access

- All definitions are combined to an *access*:

```
access GROUP context secModel secLevel match read write notif
```

- Sample:

```
access MyROSystem „“ any noauth exact system none none
access MyROGroup „“ usm priv exact all none none
```


# Simple Usermanagement

- The options `rouser` and `rwuser` provide for a simplified management:

```
rouser USER [noauth|auth|priv [OID | …]]
```

- `noauth`: No authentication for this user.
- `auth`: The user has to authenticate.
- `priv`: The communication will be encrypted.


# v3 in snmpcmd

- The command line offers a lot of options for v3 use.

```
-a authProtocol MD5|SHA
-A authPassphrase

-x privProtocol  DES|AES
-X privPassphrase

-l secLevel    noAuthNoPriv … authPriv
-u secName
-v 3
```


# v3 In The Config File

- No fun typing all the v3 optins. Therefor the file `~/.snmp/snmp.conf` exists:

```
defAuthType           MD5|SHA
defAuthPassphrase     passphrase
defSecurityLevel      noAuthNoPriv … authPriv
defSecurityName       Username
DefVersion            3
defPrivType           DES|AES
defPrivPassphrase     passphrase
```

- Also possible: `defPassphrase`


# Simple ...

- Having these entries a simple

```
snmpwalk host .system
```

… works again.

- You get: Simple Network Management!


# SNMP Myth

- „SNMP is not secure“
  - Yes! But the design of SNMPv1 was never ment to be secure.
  - SNMPv3 is secure. All messages can be authenticated and encrypted.
  - SNMPv3 offers a role based access model.

- „SNMP is not safe“ (Traps are not acknowledged)
  - SNMPv3 offers *Informs* that are being acknowledged.
- „SNMP floods the net / overloads my router“
  - Depends on the implementation, i.e. on *you*!
  - A wrong DNS-Server also can flood the net.

Thank you very much for you attention!

Questions?