ENCRYPTING DAEMON TRAFFIC WITH ZABBIX 3.0

```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

Zabbix communication

Server ↔ Agent

Server ↔ Proxy

Proxy ↔ Agent

Two problems

```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
            [2,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

Confidentiality
Authentication

Zabbix protocols

```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

What could be exposed?

- Host names
- SNMP community strings
- SSH, IPMI passwords


```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
            [0,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

What else could be exposed?

- Monitored data not that much...
- Logfile contents ?


```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
            [2,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

What do you use today?

*VPN?

stunnel?

SSH port forwarding?

Who's there?

Knock Knock

- Who's there?

Maya.

- Maya who?

MAAAYAA HEEE

MAAYAA HUU

MAYAA HAA

MAYAAA HA HA

Knock Knock

- Who's there?

Zabbix server.

- Really?

TAKE ALL THIS DATA

Encryption methods

- PSK
- Certificate based
- ~~Kerberos~~


```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

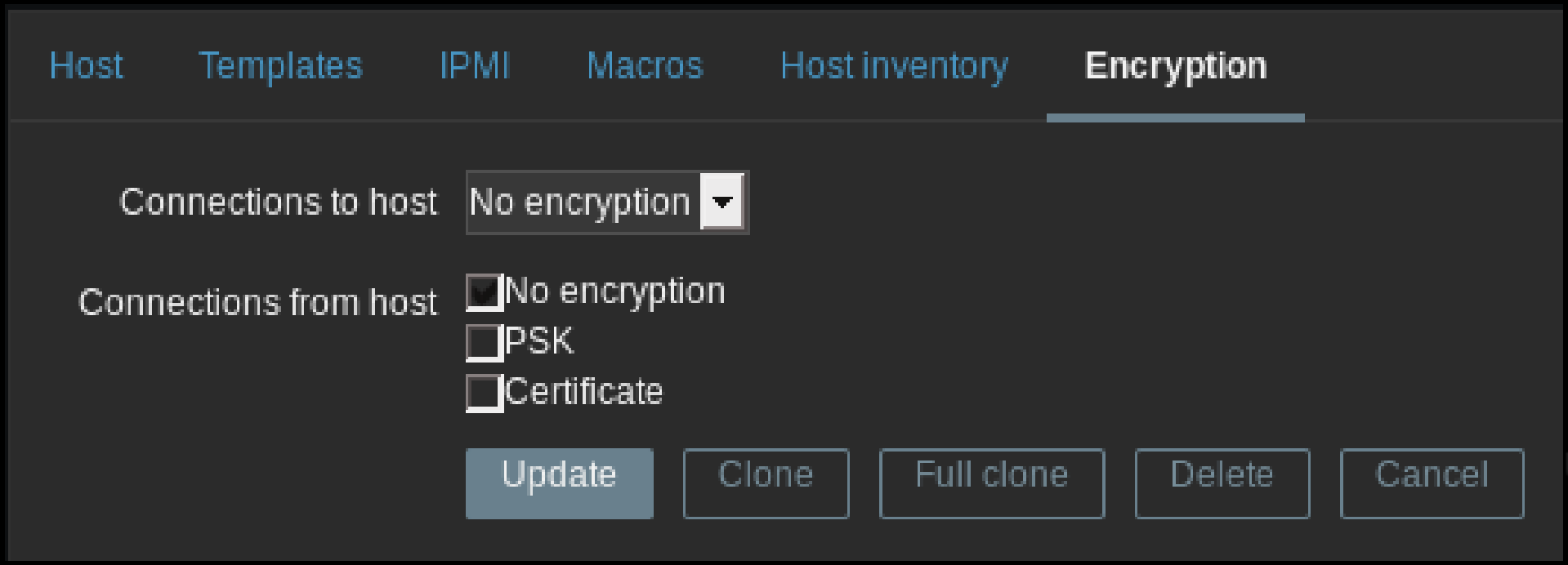
Support must be compiled in

```
 22104:20150911:103815.056 Starting Zabbix Server. Zabbix 3.0.0alpha2
(revision {ZABBIX_REVISION}).
 22104:20150911:103815.056 ****** Enabled features ******
 22104:20150911:103815.057 SNMP monitoring:            NO
 22104:20150911:103815.057 IPMI monitoring:            NO
 22104:20150911:103815.057 Web monitoring:            YES
 22104:20150911:103815.057 VMware monitoring:          NO
 22104:20150911:103815.057 SMTP authentication:       YES
 22104:20150911:103815.057 Jabber notifications:       NO
 22104:20150911:103815.057 Ez Texting notifications:  YES
 22104:20150911:103815.057 ODBC:                       NO
 22104:20150911:103815.057 SSH2 support:               NO
 22104:20150911:103815.057 IPv6 support:               NO
 22104:20150911:103815.057 TLS support:               YES
 22104:20150911:103815.057 ******************************
```

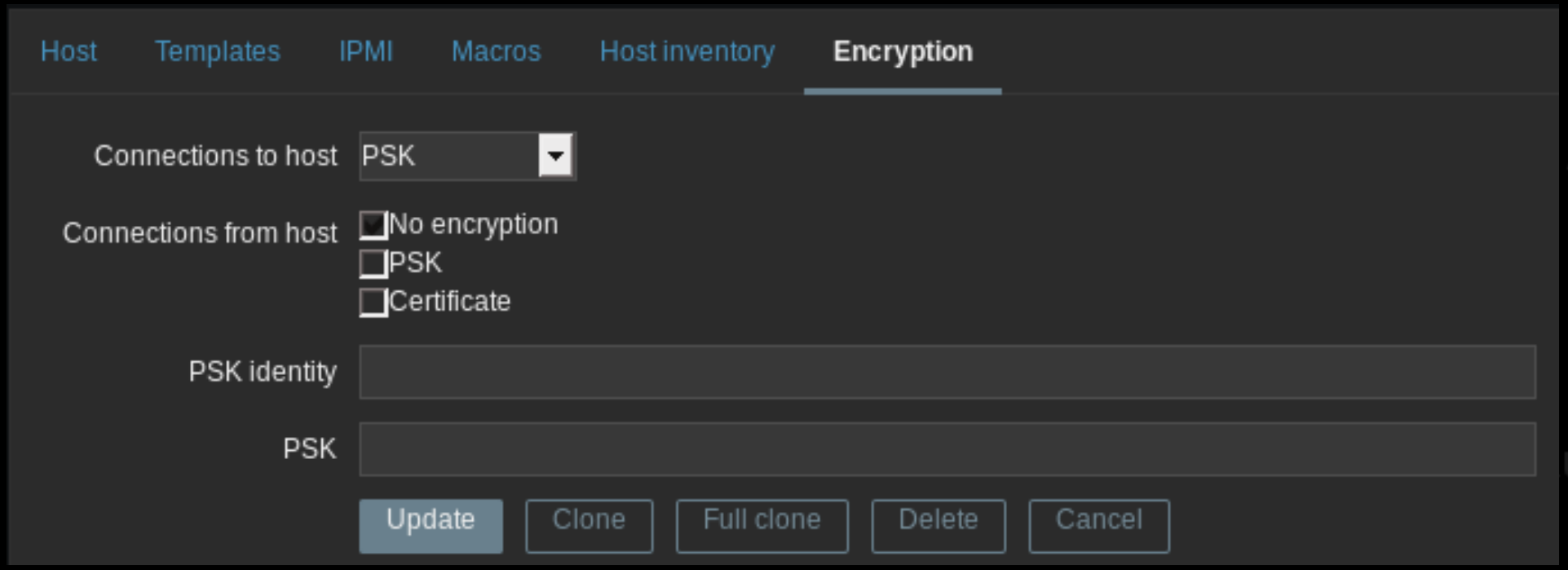
Upgraded, what now?

- Passive agent

Host properties – new tab



Let's enable PSK



Hostlist with enabled encryption

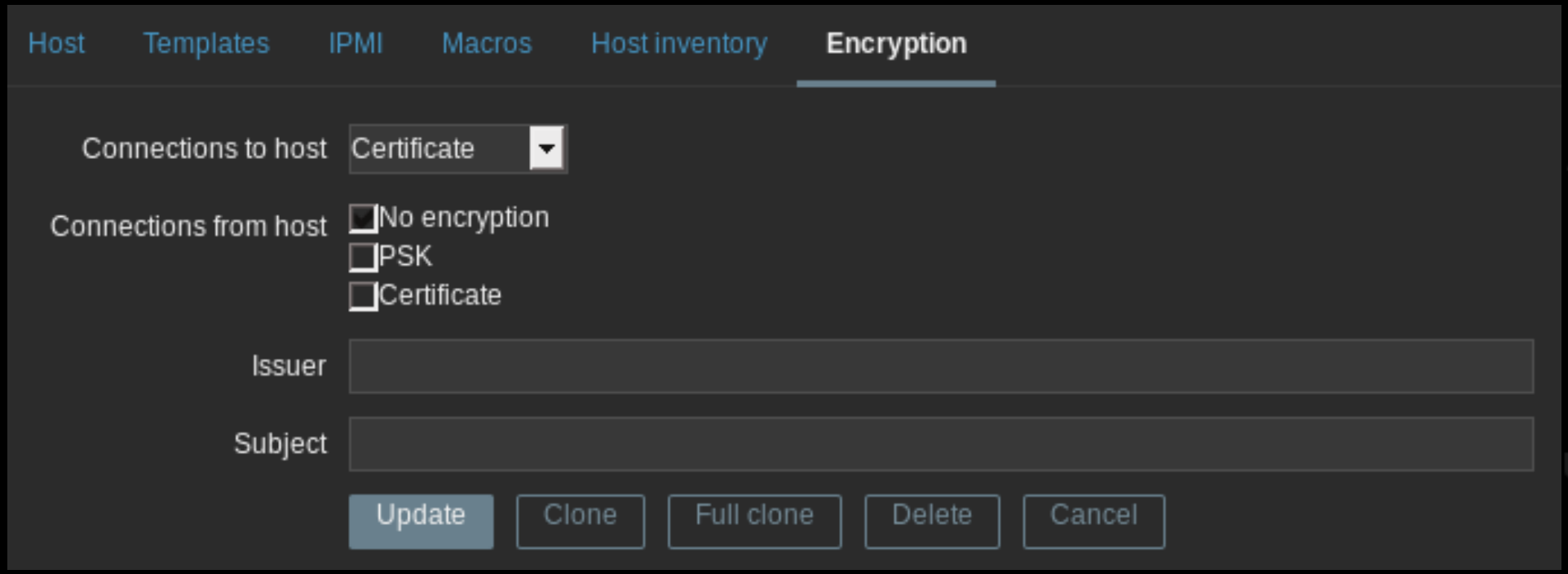
Agent side

- TLSAccept
  - unencrypted, psk, cert (multiple)
- TLSPSKIdentity
- TLSPSKFile

```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                                                    "public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```


That's it. Encrypted traffic.

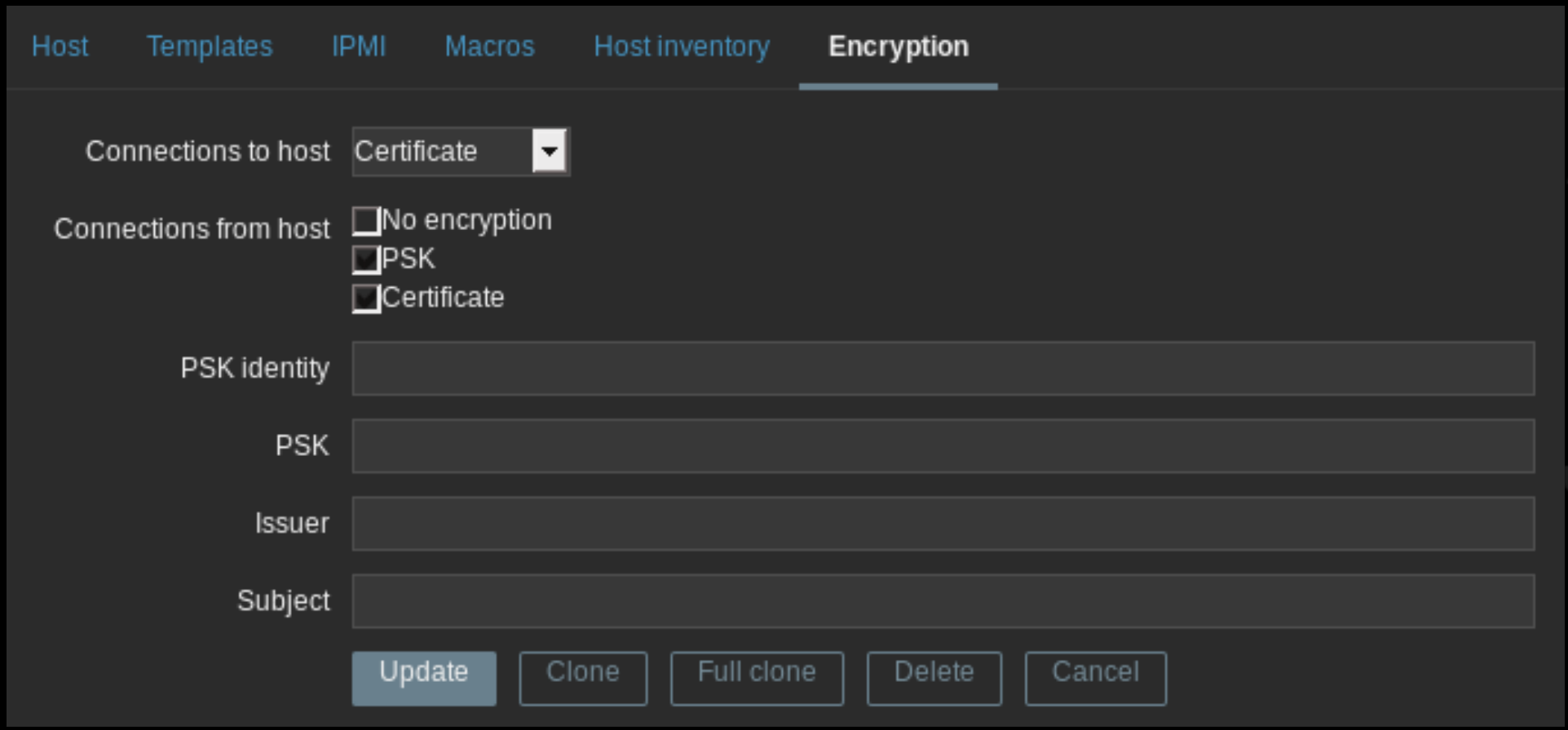
Certificates



Certificates, agent side

- TLSCAFile
- TLSCRLFile
- TLSServerCertIssuer
- TLSServerCertSubject
- TLSCertFile
- TLSKeyFile

Active agent



Active agent, agent side

- TLSConnect
  - unencrypted, psk, cert (one of)
- TLSPSKIdentity
- TLSPSKFile

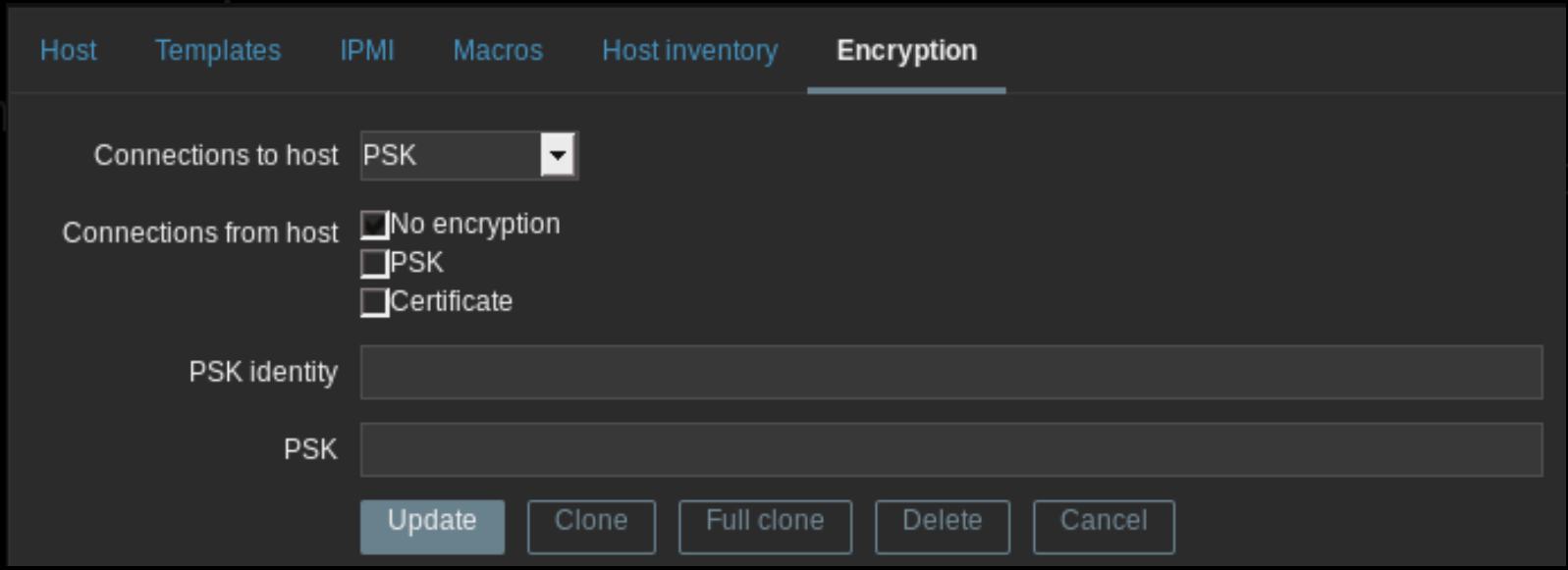

```
{
    "globalmacro" : {
        "fields" : ["hostmacroid","macro","value"],
        "data" : [
            [2,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

```
{
    "globalmacro" : {
            "fields" : ["globalmacroid","macro","value"],
            "data" : [
            ]
    },
    "hosts" : {
            "fields" : ["hostid","host","status","ipmi_auth
            "data" : [
```

I don't follow, when upgrading I do what?

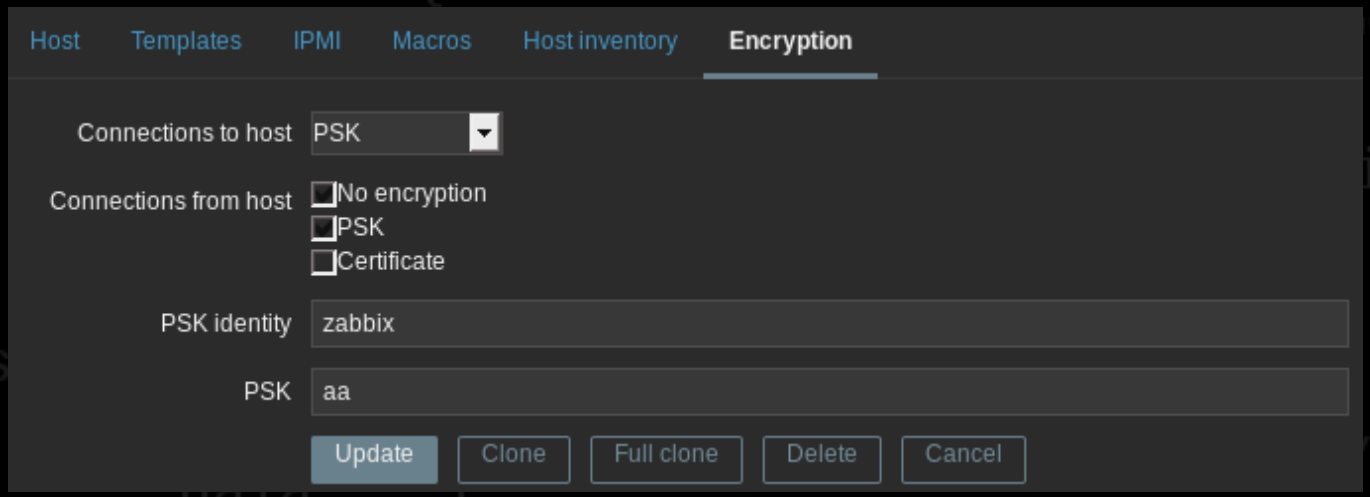
Upgrading passive agents

- Set TLSAccept=unencrypted,psk
- Set TLSPSKIdentity, TLSPSKFile
- Set connections to PSK in host properties



Upgrading active agents

- Enable unencrypted & PSK in host properties



Set TLSConnect=psk in the agent config

Set TLSPSKIdentity, TLSPSKFile

Reminder

```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

Encryption is optional

Backend library support

- OpenSSL
- GnuTLS
- PolarSSL/mbedTLS


```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
            [2,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

Doing things properly

- You want your developers pedantic
- You want your beer at room temperature


```
{
    "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
    },
    "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

Less hair pulling

- Good error messages
- Error messages


```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
            [2,"{$SNMP_COMMUNITY}","public"]
        ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

What about other tools?

Yep, also supported for:

- zabbix_get
- zabbix_sender


```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

zabbix_get pre-3.0

usage:

zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] -k item-key

zabbix_get -h

zabbix_get -V

Get data from Zabbix agent.

Options:

-s --host host-name-or-IP Specify host name or IP address of a host

-p --port port-number Specify port number of agent running on the host. Default is 10050

-I --source-address IP-address Specify source IP address

-k --key item-key Specify key of the item to retrieve value for

-h --help Display this help message

-V --version Display version number

zabbix_get in 3.0

```
usage:
  zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] -k item-key
  zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address]
              --tls-connect cert --tls-ca-file CA-file
              [--tls-crl-file CRL-file] [--tls-agent-cert-issuer cert-issuer]
              [--tls-agent-cert-subject cert-subject]
              --tls-cert-file cert-file --tls-key-file key-file -k item-key
  zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address]
              --tls-connect psk --tls-psk-identity PSK-identity
              --tls-psk-file PSK-file -k item-key
  zabbix_get -h
  zabbix_get -V

Get data from Zabbix agent.

General options:
  -s --host host-name-or-IP  Specify host name or IP address of a host
  -p --port port-number      Specify port number of agent running on the host.
                             Default is 10050
  -I --source-address IP-address   Specify source IP address

  -k --key item-key          Specify key of the item to retrieve value for

  -h --help                  Display this help message
  -V --version               Display version number

TLS connection options:
  --tls-connect value        How to connect to agent. Values:
                             unencrypted - connect without encryption
                             psk         - connect using TLS and a pre-shared
                                           key
                             cert        - connect using TLS and a
                                           certificate

  --tls-ca-file CA-file      Full pathname of a file containing the top-level
                             CA(s) certificates for peer certificate
                             verification

  --tls-crl-file CRL-file    Full pathname of a file containing revoked
                             certificates

  --tls-agent-cert-issuer cert-issuer   Allowed agent certificate issuer

  --tls-agent-cert-subject cert-subject   Allowed agent certificate subject

  --tls-cert-file cert-file  Full pathname of a file containing the certificate
                             or certificate chain

  --tls-key-file key-file    Full pathname of a file containing the private key

  --tls-psk-identity PSK-identity   Unique, case sensitive string used to
                             identify the pre-shared key

  --tls-psk-file PSK-file    Full pathname of a file containing the pre-shared
                             key

Example(s):
  zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"

  zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" \
    --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file \
    --tls-agent-cert-issuer \
    "CN=Signing CA,OU=IT operations,O=Example Corp,DC=example,DC=com" \
    --tls-agent-cert-subject \
    "CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com" \
    --tls-cert-file /home/zabbix/zabbix_get.crt \
    --tls-key-file /home/zabbix/zabbix_get.key

  zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" \
    --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" \
    --tls-psk-file /home/zabbix/zabbix_agentd.psk
```

zabbix_sender in 3.0

- ...even longer help output
- Same functionality – TLS, PSK, unencrypted


```
{
    "globalmacro" : {
        "fields" : ["globalmacroid","macro","value"],
        "data" : [
                [2,"{$SNMP_COMMUNITY}","public"]
            ]
    },
    "hosts" : {
        "fields" : ["hostid","host","status","ipmi_auth
        "data" : [
```

Sender commandline

$ zabbix_sender -z 127.0.0.1 -s host -k item_key -o value


```
{
        "globalmacro" : {
                "fields" : ["globalmacroid","macro","value"],
                "data" : [
                        [2,"{$SNMP_COMMUNITY}","public"]
                ]
        },
        "hosts" : {
                "fields" : ["hostid","host","status","ipmi_auth
                "data" : [
```

Sender commandline

```
$ zabbix_sender -z 127.0.0.1
-s host -k item_key -o value
--tls-connect psk --tls-psk-file psk
--tls-psk-identity zabbix
```

tcpdump before, after

```
E...T.@.@.......
.....W'/.1W...w.
...V..........@
...?ZBXD.^......
.{"request":"sen
der.data","data"
:[{"host":"impor
tant.host","key"
:"secret","value
":"revealed"}]}
```

```
E..<.X@.@.xa....
....../.qO.....
.....0.........
.............
E..4.Y@.@.xh....
....../.qO..D.h
...V.(.........
....
E....Z@.@.x.....
....../.qO..D.h
...V............
........^...Z..]
..)pk...2..qz.-0
.........h.W\w..
.........-.#....
................
................
```

```
.......
E..4.[@.@.xf....
....../.qO..D..
...V.(.........
....
E....\@.@.x.....
....../.qO..D..
...V............
...............z
abbix..........@
...l{...,.oN.]..
.Ry..{..*.W...{.
....%....l.*]+.H
```

```
....w..J.e8.*...
E....]@.@.w.....
....../.qPQ.D..
...^............
.............=..
...J..*5.mmP^..'
..5.....X..'H+.b
.).q....b....1..
.x./lL..%...R..3
.g.|...AGH....!.
.KU...K....6S./~
....".....;u....
&.....qPw.O....a
..W..1.x]
```

Executive summary

- Built-in encryption (easy to use)
- Strong authentication
- Painless upgrade – encryption is optional
- Can be enabled for selected proxies, agents

Technical summary

- Built-in encryption
- Painless upgrade

Technical summary

- Connection methods
  - PSK
  - TLS
  - Unencrypted
- Supported libraries
  - OpenSSL
  - GnuTLS
  - PolarSSL/mbetTLS

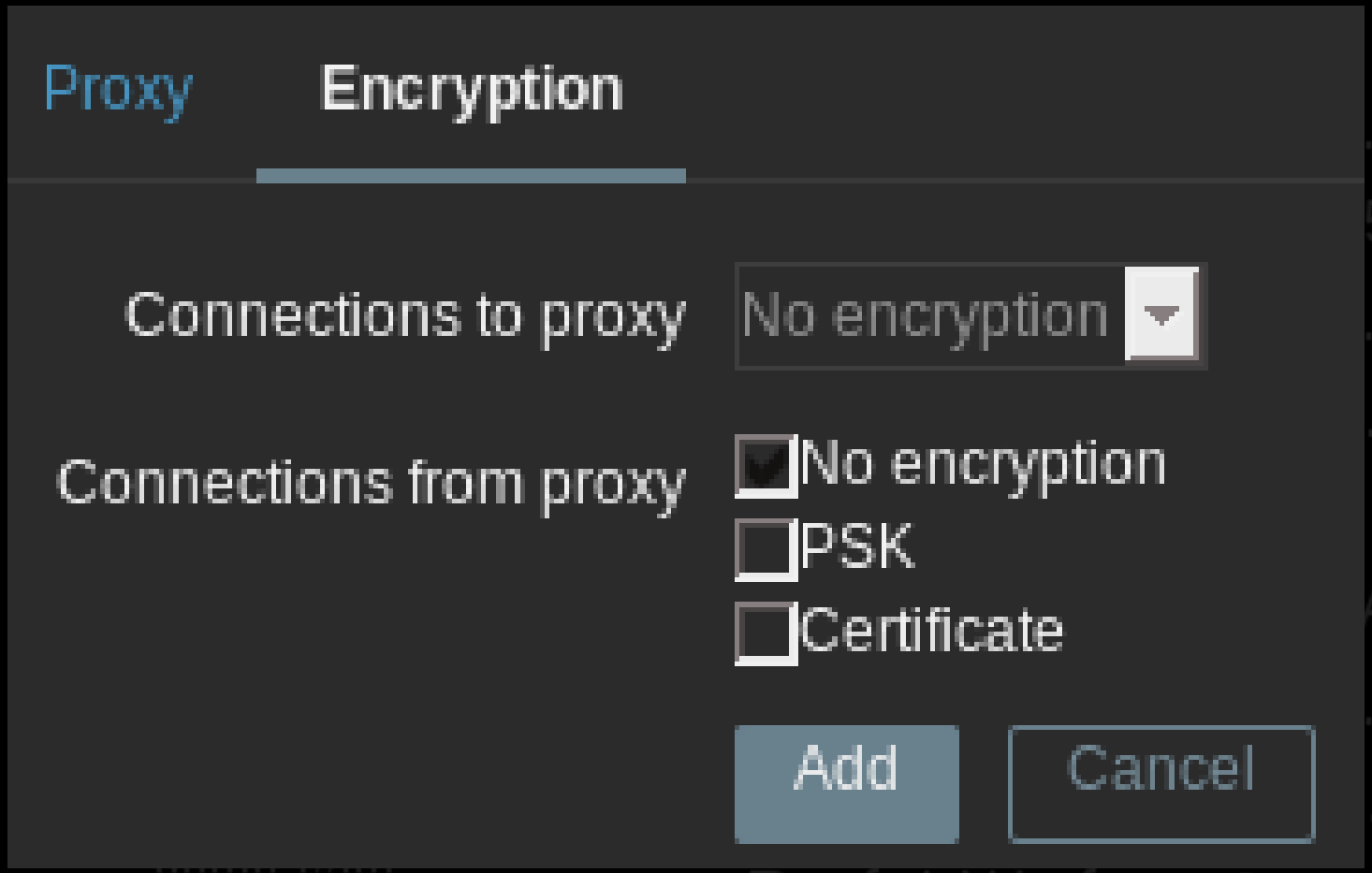
Things to keep in mind

Not for plain agent

Overhead – problem with passive, less with active

- Increase BufferSend for the active agent

Proxy, you said "proxy"!



WORKSFORME

Do you plan to use encryption?

Have you tested it already?

Devs, involved in encryption work, are on IRC

Question time

IRC

Developers are here, use the chance

Encrypt away