

Pluggable Authentication in MariaDB and MySQL



The problem?

The problem?

**MySQL native authentication system
is not flexible enough**

The problem?

MySQL native authentication system

cannot be

~~is not flexible enough~~

The problem?

MySQL native authentication system

cannot be

~~**is not flexible enough**~~

Does not integrate!

Solution?

Solution?

MySQL Plugin API

Solution?

MySQL Plugin API



Pluggable authentication!

Solution?

**No need
to wait!**

MySQL Plugin API



Pluggable authentication!

Solution?

**No need
to wait!**

MySQL Plugin API



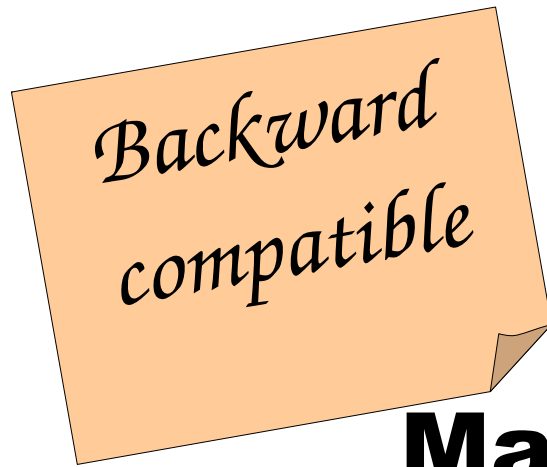
Pluggable authentication!



Pluggable Authentication

Main Features

Pluggable Authentication



Main Features

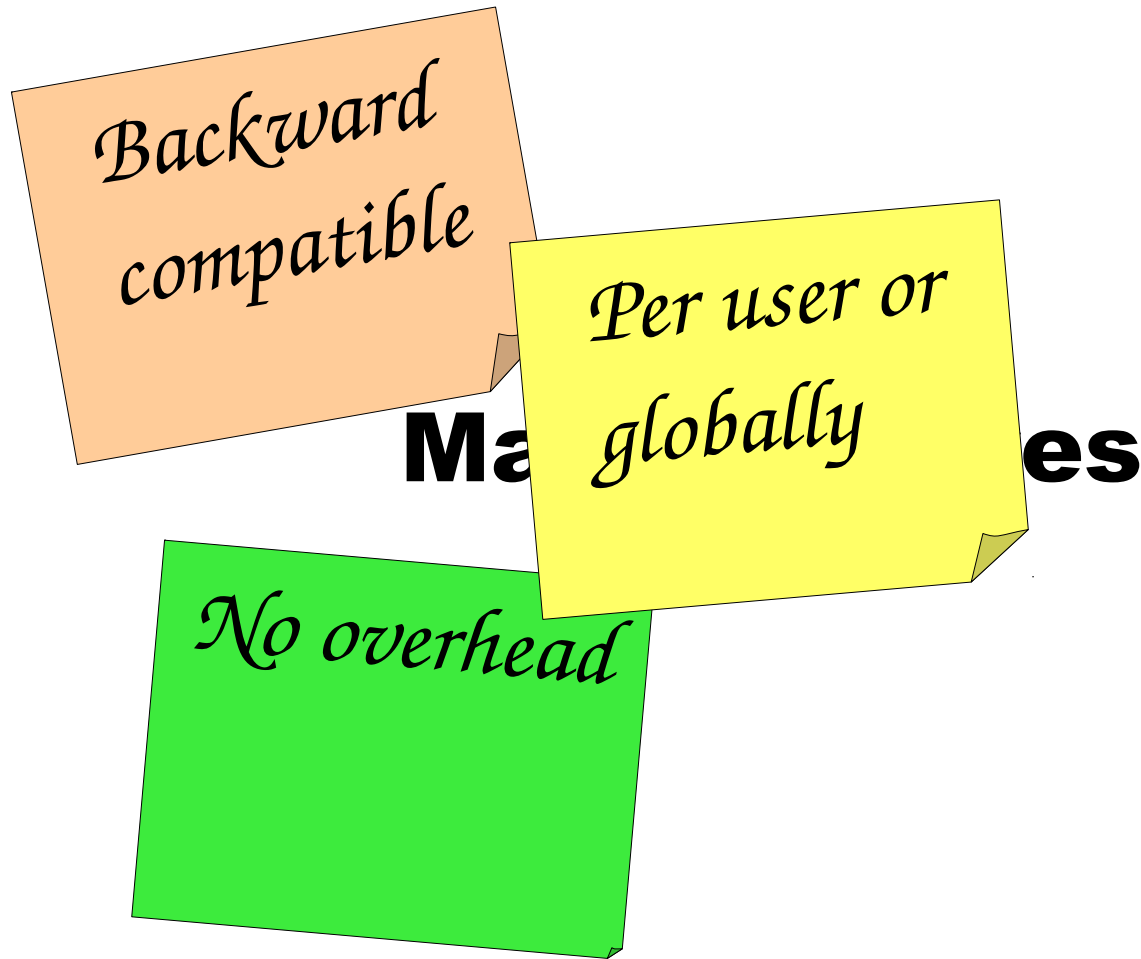
Pluggable Authentication

*Backward
compatible*

Main Features

No overhead

Pluggable Authentication



Pluggable Authentication

*Backward
compatible*

Ma

*Per user or
globally*

*Stays out of
your way*

No overhead

Pluggable Authentication

*Backward
compatible*

Ma

*Per user or
globally*

*Stays out of
your way*

No overhead

*Extends both
client and
server*

Server

Server

```
CREATE USER serg IDENTIFIED BY 'password'
```

Server

```
CREATE USER serg IDENTIFIED WITH auth_socket
```

Server

```
CREATE USER serg IDENTIFIED WITH auth_socket
```

```
CREATE USER '' IDENTIFIED WITH ldap
```

Clients

Clients

Client-side plugins

Clients

Client-side plugins

Loaded automatically

Clients

Client-side plugins

Loaded automatically

Callback support for GUI

API

API

As simple as it could be

`write_packet()` `read_packet()`

API

As simple as it could be

`write_packet()` `read_packet()`

Server handles the rest

handshake, escaping, negotiation, restart, SSL, ...

Example

```
static int usbsn_client(MYSQL_PLUGIN_VIO *vio, MYSQL *mysql)
{
    int len, res, fd;
    char buf[10240];

    fd = open("/proc/bus/usb/devices", O_RDONLY);
    if (fd == -1)
        return CR_ERROR;

    len = read(fd, buf, sizeof(buf) - 1);
    close(fd);
    if (len == 0)
        return CR_ERROR;

    buf[len++] = 0;

    res = vio->write_packet(vio, buf, len);

    return res ? CR_ERROR : CR_OK;
}
```

Example

```
static int usbsn_client(MYSQL_PLUGIN_VIO *vio, MYSQL *mysql)
{
    int len, res, fd;
    char buf[10240];

    fd = open("/proc/bus/usb/devices", O_RDONLY);
    if (fd == -1)
        return CR_ERROR;

    len = read(fd, buf, sizeof(buf) - 1);
    close(fd);
    if (len == 0)
        return CR_ERROR;

    buf[len++] = 0;

    res = vio->write_packet(vio, buf, len);

    return res ? CR_ERROR : CR_OK;
}
```

Example

```
static int usbsn_client(MYSQL_PLUGIN_VIO *vio, MYSQL *mysql)
{
    int len, res, fd;
    char buf[10240];

    fd = open("/proc/bus/usb/devices", O_RDONLY);
    if (fd == -1)
        return CR_ERROR;

    len = read(fd, buf, sizeof(buf) - 1);
    close(fd);
    if (len == 0)
        return CR_ERROR;

    buf[len++] = 0;

    res = vio->write_packet(vio, buf, len);

    return res ? CR_ERROR : CR_OK;
}
```

Example

```
static int usbsn_client(MYSQL_PLUGIN_VIO *vio, MYSQL *mysql)
{
    int len, res, fd;
    char buf[10240];

    fd = open("/proc/bus/usb/devices", O_RDONLY);
    if (fd == -1)
        return CR_ERROR;

    len = read(fd, buf, sizeof(buf) - 1);
    close(fd);
    if (len == 0)
        return CR_ERROR;

    buf[len++] = 0;

    res = vio->write_packet(vio, buf, len);

    return res ? CR_ERROR : CR_OK;
}
```

Example

```
static int usbsn_server(MYSQL_PLUGIN_VIO *vio,  
                       MYSQL_SERVER_AUTH_INFO *info)  
{  
    unsigned char *pkt;  
    int pkt_len;  
  
    if ((pkt_len = vio->read_packet(vio, &pkt)) < 0)  
        return CR_ERROR;  
  
    return strstr(pkt, info->auth_string) ? CR_OK : CR_ERROR;  
}
```

Example

```
static int usbsn_server(MYSQL_PLUGIN_VIO *vio,  
                        MYSQL_SERVER_AUTH_INFO *info)  
{  
    unsigned char *pkt;  
    int pkt_len;  
  
    if ((pkt_len = vio->read_packet(vio, &pkt)) < 0)  
        return CR_ERROR;  
  
    return strstr(pkt, info->auth_string) ? CR_OK : CR_ERROR;  
}
```

Example

```
static int usbsn_server(MYSQL_PLUGIN_VIO *vio,  
                       MYSQL_SERVER_AUTH_INFO *info)  
{  
    unsigned char *pkt;  
    int pkt_len;  
  
    if ((pkt_len = vio->read_packet(vio, &pkt)) < 0)  
        return CR_ERROR;  
  
    return strstr(pkt, info->auth_string) ? CR_OK : CR_ERROR;  
}
```

Example

```
static int usbsn_server(MYSQL_PLUGIN_VIO *vio,
                       MYSQL_SERVER_AUTH_INFO *info)
{
    unsigned char *pkt;
    int pkt_len;

    if ((pkt_len = vio->read_packet(vio, &pkt)) < 0)
        return CR_ERROR;

    return strstr(pkt, info->auth_string) ? CR_OK : CR_ERROR;
}
```

```
CREATE USER serg IDENTIFIED WITH usbsn
AS '1A08051410110'
```

The Plugin Book



That's all

Questions?

Example (complete)

```
static int usbsn_verify(MYSQL_PLUGIN_VIO *vio,
                       MYSQL_SERVER_AUTH_INFO *info)
{
    unsigned char *pkt;
    int pkt_len;
    size_t buflen = strlen(info->auth_string) + 20;
    char *buf = alloca(buflen);

    my_snprintf(buf, buflen, "S: SerialNumber=%s\n",
                info->auth_string);

    if ((pkt_len = vio->read_packet(vio, &pkt)) < 0)
        return CR_ERROR;

    return strstr(pkt, buf) ? CR_OK : CR_ERROR;
}
```