What's new in Bareos 18

bareos
open source data protection

- Philipp Storz
- Frank Ueberschar

What's new in Bareos 18.2?

- *codebase:*
  - **715** files changed
  - **27.230** insertions(+)
  - **25.458** deletions(-)
  - use modern c++ 11 language features
  - repository reorganization
  - change codebase from legacy code to modern code
  - introduced gtest test framework
  - switched buildsystem from autoconf to cmake
- *functionality:*
  - privacy by default automatic tls
  - pam authentication

Repository reorganization

Status before

- Multiple repositories:
    - **bareos/bareos** : the core components (daemons)
    - **bareos/bareos-docs** : documentation
    - **bareos/bareos-regress** : regression tests
    - **bareos/bareos-vmware** : vmware-plugin
    - **bareos/bareos-webui** : bareos web user interface
    - **bareos/python-bareos** : bareos python tools

Status before

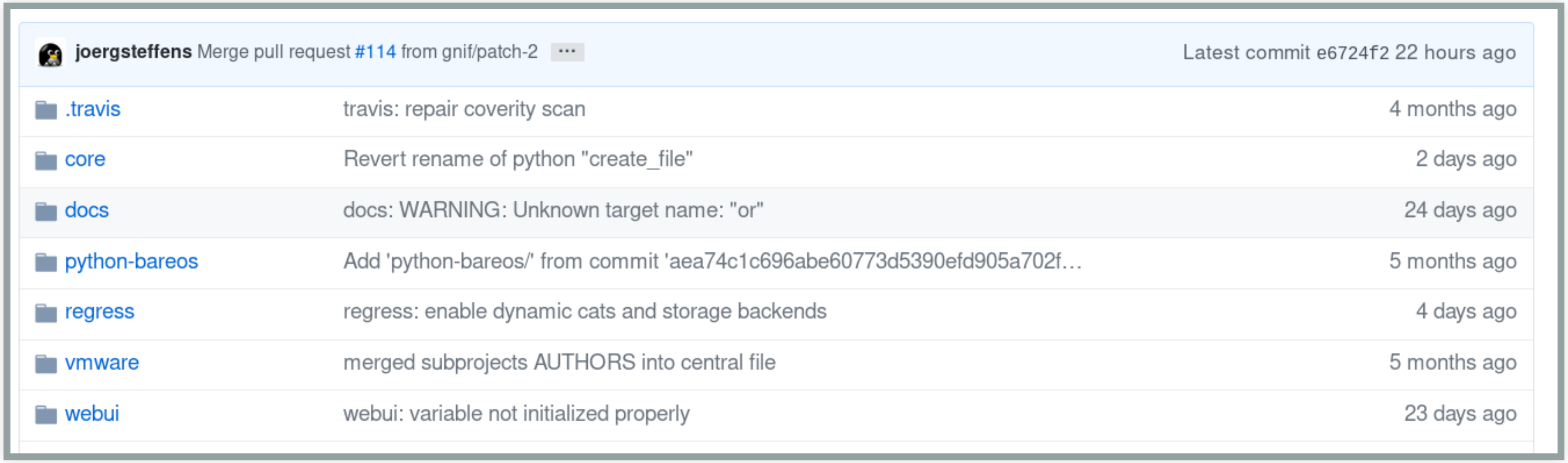
- problematic to keep repos in-sync
- releasing difficult as different repos need to be built and packaged
- unnecessary complexity

Bareos 18:

- All important repos have been merged via **subtree merge** into the bareos/bareos repo:
  - bareos/bareos -> **core/** subdirectory
  - bareos/bareos-docs -> **docs/** subdirectory
  - bareos/bareos-regress -> **regress/** subdirectory
  - bareos/bareos-vmware -> **vmware/** subdirectory
  - bareos/bareos-webui -> **webui/** subdirectory
  - bareos/python-bareos -> **python-bareos** subdirectory

Bareos 18:

- Everything in one repository
- Clean structure
- Everything is easily kept in sync
- Commit history of merged subtrees stays intact



joergsteffens Merge pull request #114 from gnif/patch-2 … Latest commit e6724f2 22 hours ago

.travis	travis: repair coverity scan	4 months ago
core	Revert rename of python "create_file"	2 days ago
docs	docs: WARNING: Unknown target name: "or"	24 days ago
python-bareos	Add 'python-bareos/' from commit 'aea74c1c696abe60773d5390efd905a702f…	5 months ago
regress	regress: enable dynamic cats and storage backends	4 days ago
vmware	merged subprojects AUTHORS into central file	5 months ago
webui	webui: variable not initialized properly	23 days ago

Switch to CMake build system

old build system: autoconf

- "GNU Autoconf is a tool for producing configure scripts for building, installing and packaging software on computer systems where a Bourne shell is available."
- Autoconf is **dependent on unix tools** and uses the **m4 language**.
- Changing the build process in autoconf is a pain
- Lots of work: makefiles.in need to be made manually
- Obscure functionality thru libtool usage

switch to cmake build system

- CMake is an open-source, cross-platform family of tools designed to build, test and package software.
- CMake can do what autoconf/automake does and more.
- CMake has much cleaner syntax and needs no external dependencies
- CMake **also runs on windows**
- CMake needs far less work than autoconf
- CMake gets the dependencies of the source files configured
- CMake generates all makefiles itself, **no hand-made Makefiles**

console Makefile.in from automake: **129 lines**

```
@MCOMMON@

srcdir = @srcdir@
VPATH = @srcdir@
.PATH: @srcdir@

# one up
basedir = ..
# top dir
topdir = ../..
# this dir relative to top dir
thisdir = src/console

DEBUG=@DEBUG@

first_rule: all
dummy:

#
CONSSRCS = console.c console_conf.c @CONS_SRC@
CONSOBJS = $(CONSSRCS:.c=.o)

GETTEXT_LIBS = @LIBINTL@
OPENSSL_LIBS_NONSHARED = @OPENSSL_LIBS_NONSHARED@
```

console CMakeLists.txt: **20 lines**

```
SET (BCONSSRCS console.cc console_conf.cc)
IF(HAVE_WIN32)
     LIST(APPEND BCONSSRCS ../win32/console/consoleres.rc)
ENDIF()

add_executable(bconsole ${BCONSSRCS})


set(CONSOLE_LINK_LIBRARIES bareos bareoscfg ${Readline_LIBRARY})

IF(HAVE_WIN32)
    LIST(APPEND CONSOLE_LINK_LIBRARIES bareosstatic)
ENDIF()

target_link_libraries(bconsole ${CONSOLE_LINK_LIBRARIES})

INSTALL(TARGETS bconsole DESTINATION "${bindir}")
INSTALL(TARGETS bconsole DESTINATION "${sbindir}")
INSTALL(FILES bconsole.conf DESTINATION "${configtemplatedir}")
```

Other advantages of cmake:

- Windows cross build:
  - before: hand-crafted makefiles independent from the unix build.
  - cmake using the definitions that the other code also uses.
    - changes are automatically also applied to windows cross builds!
- Qt tray monitor
  - before: needed qmake and a special qmake project
  - cmake does everything

Comparison of build systems in bareos

	autoconf (Bareos 17)
lines of code	3190 total lines in Makefile.in files
	36796 total lines in autoconf/* files
	36703 configure script.
sum	**76689**

	CMake (Bareos 18)
lines of code	2998 total in CMakeLists.txt files
	2296 total in core/cmake/* modules
sum	**5294**

Questions?

Automatic TLS

TLS with Bareos 17 (1):

- Per default the network communication is not encrypted
- TLS can be enabled
- Certificate Authority and certificates are mandatory
- TLS is only started *AFTER* authentication is complete!

TLS with Bareos 17 (2):

- CRAM-MD5 authentication is done in cleartext
- It is **not visible** if communication is encrypted or not
- Configuration is complicated:
- This configuration block needs to be added to many resources:

```
TLS Enable = yes
TLS Certificate = /etc/bareos/ssl/crt/bareos.crt
TLS CA Certificate File = /etc/bareos/ssl/crt/bareos-ca.pem
TLS Key = /etc/bareos/ssl/private/bareos.pem
TLS Allowed CN = bareos
TLS Verify Peer = no
```

Gloals for Bareos 18 (1):

- Per default the network communication is encrypted
- TLS is enabled by default
- Current status of encryption **clearly** visible
- No other network ports are needed

Gloals for Bareos 18 (2):

- Certificate authority and certificates are not needed
- Extra configuration is not needed
- TLS is started immediately
- CRAM-MD5 authentication is done inside of TLS Tunnel
- Full backward compatibility with old clients

About TLS-PSK

- Additional to TLS based on certificates, there are other options to establish TLS
- TLS-PSK can establish TLS based on shared secrets (Pre Shared Keys = PSK)
  - *identity*
  - *key*
- Bareos has shared secrets on both sides of each connection:
  - *name*
  - *password*
- Why not use the name as identity and password as key?
  - With TLS-PSK, we can do TLS without extra configuration!

TLS-PSK vs. TLS-Certificates

- Both TLS-PSK and TLS-Cert can be enabled at the same time
- Which one is used is determined by TLS during the initiation of the communication
- Existing TLS Certificates will be used

Using TLS-PSK:

- Gloals achieved:
    - ✓ Per default the network communication is encrypted
    - ✓ TLS is enabled by default
    - ✓ Certificate authority and certificates are not needed
    - ✓ Extra configuration is not needed
    - ✓ No other network ports are needed

Using TLS-PSK:

- Gloals still open:
  - ☐ Current status of encryption clearly visible
  - ☐ TLS is started immediately
  - ☐ CRAM-MD5 authentication is done inside of TLS Tunnel
  - ☐ Full backward compatibility with old clients

Tell encryption status

- console connection:

```
Connecting to Director localhost:8101
1000 OK: bareos-dir Version: 18.2.4rc1 (25 Sep 2018)
Secure connection with cipher ECDHE-PSK-CHACHA20-POLY1305
```

- job log:

```
Secure connection to Storage daemon at localhost:8103 with cipher ECDHE-PSK-CHACHA20-POLY1305 e
```

Protocol change: start TLS immediately

- Bareos 18 starts TLS immediately
- Inside of TLS Tunnel "old" CRAM-MD5 authentication is done

Status with telling the encryption status and protocol change to immediate TLS

- ✓ Bareos 18 talks immediately TLS
- ✓ CRAM-MD5 authentication is done inside of TLS Tunnel
- ✓ Current status of encryption clearly visible
- ☐ Full backward compatibility with old clients

Backward compatibility

- Prerequisites
  - Compatiblity only is intended for bareos clients
  - Director, Storage Daemon, Console need to be upgraded to Version 18
- Two kinds of connections exist:
  - (1) Incoming connections from old clients
  - (2) Outgoing connections to old clients

(1) Incoming connections from old clients

- Problem: It is not possible to listen on one port with TLS and with clear text at the same time
- TLS is plugged between the application and the network:
  - The application talks clear text to the TLS layer
  - The TLS layer encrypts and sends the data
  - The TLS layer receives the decrypts data
  - When TLS gets enabled, it does everything on its own
- Usually, a special port is used for TLS communication (http/https 80/443)
- This solution is not compatible with old clients

Incoming connections from old clients

- Bareos protocol messages always start with "Hello ...."
- MSG_PEEK option in the recv() command allows to peek into receive buffer:

```
ssize_t recv(int sockfd, void *buf, size_t len, int flags);
MSG_PEEK
     This  flag causes the receive operation to return data from the
     beginning of the receive queue without removing that data from the
     queue.  Thus, a subsequent receive call will return the same data
```

Incoming connections from old clients

Detect if old client is connecting

```
// src/lib/bsock.cc
bool BareosSocket::IsCleartextBareosHello()
{
  char buffer[12];
  memset(buffer, 0, sizeof(buffer));
  int ret = ::recv(fd_, buffer, 10, MSG_PEEK);
  if (ret == 10) {
    std::string hello("Hello ");
    std::string received(&buffer[4]);
    if (hello == received) { return true; }
  }
  return false;
}
```

Incoming connections from old clients

- The important part is that the data is **not removed** from the receive buffer
- This way, we can decide if we have a
    - Bareos Hello message -> go on with old protocol
    - TLS Client Hello message -> enable TLS
- as the peeking does not change the buffer, everything works!

Incoming connections from old clients

Solved!

Outgoing connections to old clients

- As we don't know what kind of client we have, we need to do client probing:
- Client probing:
    - Try to connect via TLS
        - success -> we are done and have a modern client
        - failure -> we might have an old client
            - try to connect with the old protocol
                - if that works, we have an legacy client
                - if that fails, we have a failure
- Client probing takes about 5 seconds in our test environments if a legacy client is connected

Outgoing connections to old clients

- First connection

```
Connecting to Client standard-fd at standard.bareos.org:9102
Try to establish a secure connection by immediate TLS handshake: Failed
Try to establish a secure connection by cleartext handshake: Cleartext co
standard-fd Version: 17.2.4 (21 Sep 2017) [ ... ]
```

- Second connection

```
 Connecting to Client standard-fd at standard.bareos.org:9102
 Using previously recognized cleartext handshake: Cleartext connection
 standard-fd Version: 17.2.4 (21 Sep 2017)  x86_64-redhat-linux-gnu [ ...
```

Incoming connections from old clients

Solved!

Automatic TLS is automagic TLS!

- ✓ Per default the network communication is encrypted
- ✓ TLS is enabled by default
- ✓ Certificate Authority and certificates are not needed
- ✓ Extra configuration is not needed
- ✓ No other network ports are needed
- ✓ TLS is started immediately
- ✓ CRAM-MD5 authentication is done inside of TLS Tunnel
- ✓ Full backward compatibility with old clients
- ✓ Current status of encryption clearly visible

Questions?

Pam user authentication

Why PAM?

- Users are implemented in Bareos as "named consoles"
- without PAM: Passwords for each console is stored in clear text in the configuration:

```
Console {
  Name = franku
  Profile = admin
  Password = "secret-password"
}
```

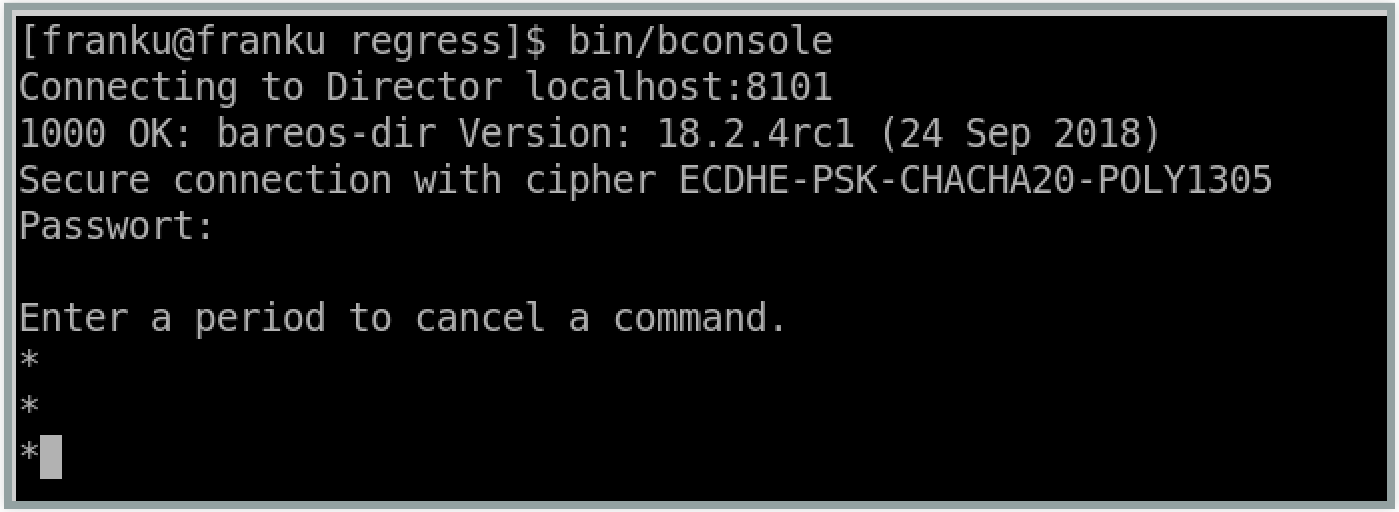
- with PAM: no cleartext password in configuration
- with PAM: password change does not need configuration change

PAM authentication

- PAM: Pluggable Authentication Module
- Choose how individual applications authenticate users
- Suite of shared libraries
- Configure i.e. in /etc/pam.d, no need to recompile application
- Bareos PAM implementation is a technical preview and still under development

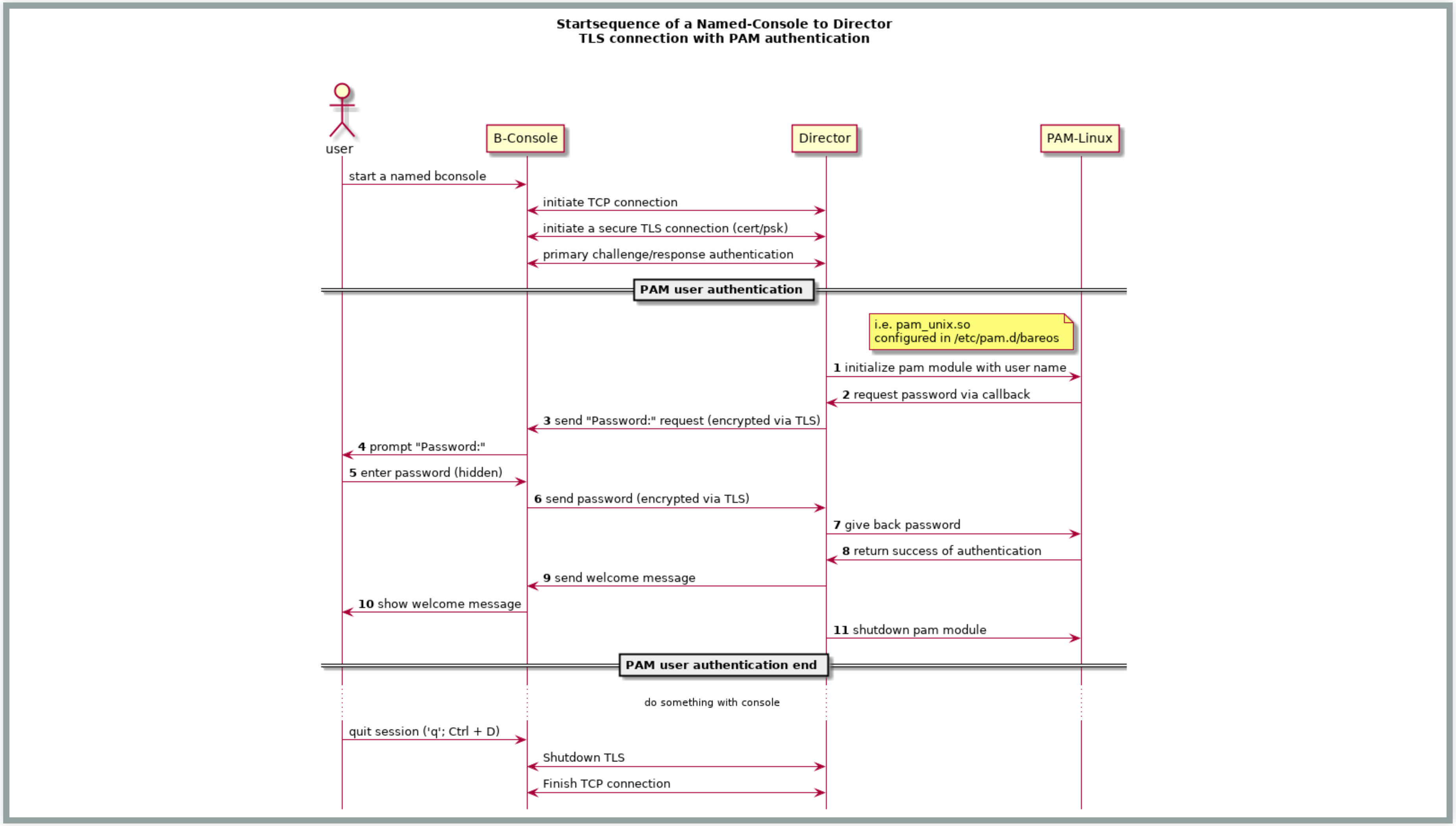
PAM authentication

- Used only for named console
- Name of the console is username



```
[franku@franku regress]$ bin/bconsole
Connecting to Director localhost:8101
1000 OK: bareos-dir Version: 18.2.4rc1 (24 Sep 2018)
Secure connection with cipher ECDHE-PSK-CHACHA20-POLY1305
Passwort:

Enter a period to cancel a command.
*
*
*
```



Startsequence of a Named-Console to Director
TLS connection with PAM authentication

user
B-Console
Director
PAM-Linux
start a named bconsole
initiate TCP connection
initiate a secure TLS connection (cert/psk)
primary challenge/response authentication
PAM user authentication
i.e. pam_unix.so
configured in /etc/pam.d/bareos
1 initialize pam module with user name
2 request password via callback
3 send "Password:" request (encrypted via TLS)
4 prompt "Password:"
5 enter password (hidden)
6 send password (encrypted via TLS)
7 give back password
8 return success of authentication
9 send welcome message
10 show welcome message
11 shutdown pam module
PAM user authentication end
do something with console
quit session ('q'; Ctrl + D)
Shutdown TLS
Finish TCP connection

How to enable PAM for bareos

- Name of the service is "bareos"
- Add file "bareos" to /etc/pam.d containing:

```
auth        required     pam_unix.so
```

- General parameter in Director-Config
- Parameter in each named Console (Director Resource)
- Add

```
"UsePamAuthentication = yes"
```

Questions?

Thank you

Bareos 18.2rc1 is available on download.bareos.org