
tacacs_{plus}
Release alpha

Aug 02, 2023

1	TACACS+ Python client	1
1.1	Basic Installation and Usage	1
1.2	Programmatic Usage	3
2	API documentation	5
2.1	<code>tacacs_plus.client</code> module	5
2.2	<code>tacacs_plus.packet</code> module	7
2.3	<code>tacacs_plus.authentication</code> module	7
2.4	<code>tacacs_plus.authorization</code> module	8
2.5	<code>tacacs_plus.accounting</code> module	8
2.6	<code>tacacs_plus.flags</code> module	9
3	Indices and tables	11
	Index	13

CHAPTER 1

TACACS+ Python client

A TACACS+ client that supports authentication, authorization and accounting.

Unlike RADIUS, which was designed for similar purposes, the TACACS+ protocol offers basic packet encryption but, as with most crypto designed back then, it's **not secure** and definitely should not be used over untrusted networks.

This package has been successfully used with the free [tac_plus](#) TACACS+ server on a variety of operating systems.

1.1 Basic Installation and Usage

```
$ pip install tacacs_plus

$ tacacs_client -u myuser -H localhost authenticate
$ tacacs_client -u myuser -H localhost authenticate -t pap
$ tacacs_client -u myuser -H localhost -v authenticate -t chap
status: PASS

$ tacacs_client -u myuser -H localhost authorize -c service=shell cmd=show_
↪cmdarg=version
$ tacacs_client -u myuser -H localhost -v authorize -t pap -c service=shell cmd=show_
↪cmdarg=version
status: PASS

$ tacacs_client -u myuser -H localhost -v authorize -t pap -c service=junos-exec
status: REPL
av-pairs:
    allow-commands=^acommandregex$
    deny-commands=^anothercommandregex$

$ tacacs_client -u myuser -H localhost account -f start -c service=shell cmd=show_
↪cmdarg=version
$ tacacs_client -u myuser -H localhost account -f stop -c service=shell cmd=show_
↪cmdarg=version
```

(continues on next page)

(continued from previous page)

```
$ tacacs_client -h
usage: tacacs_client [-h] -u USERNAME -H HOST [-p PORT] [-l PRIV_LVL]
                    [-t {ascii,pap,chap}] [-r REM_ADDR] [-P VIRTUAL_PORT]
                    [--timeout TIMEOUT] [-d] [-v] [-k KEY]
                    {authenticate,authorize,account} ...

    Tacacs+ client with full AAA support:

    * Authentication supports both ascii, pap and chap.
    * Authorization supports AV pairs and single commands.
    * Accounting support AV pairs and single commands.

    NOTE: shared encryption key can be set via environment variable TACACS_PLUS_
    ↪KEY or via argument.
    NOTE: user password can be setup via environment variable TACACS_PLUS_PWD or
    ↪via argument.

positional arguments:
  {authenticate,authorize,account}
                                action to perform over the tacacs+ server
  authenticate              authenticate against a tacacs+ server
  authorize                  authorize a command against a tacacs+ server
  account                    account commands with accounting flags against a tacacs+
    ↪server

optional arguments:
  -h, --help                show this help message and exit
  -u USERNAME, --username USERNAME
                                user name
  -H HOST, --host HOST      tacacs+ server address
  -p PORT, --port PORT      tacacs+ server port (default 49)
  -l PRIV_LVL, --priv-lvl PRIV_LVL
                                user privilege level
  -t {ascii,pap,chap}, --authen-type {ascii,pap,chap}
                                authentication type
  -r REM_ADDR, --rem-addr REM_ADDR
                                remote address (logged by tacacs server)
  -P VIRTUAL_PORT, --virtual-port VIRTUAL_PORT
                                console port used in connection (logged by tacacs server)
  --timeout TIMEOUT
  -d, --debug                enable debugging output
  -v, --verbose              print responses
  -k KEY, --key KEY          tacacs+ shared encryption key

$ tacacs_client authenticate -h
usage: tacacs_client authenticate [-h] [-p PASSWORD]

optional arguments:
  -h, --help                show this help message and exit
  -p PASSWORD, --password PASSWORD
                                user password

$ tacacs_client authorize -h
usage: tacacs_client authorize [-h] -c CMDS [CMDS ...]

optional arguments:
```

(continues on next page)

(continued from previous page)

```

-h, --help            show this help message and exit
-c CMDS [CMDS ...], --cmds CMDS [CMDS ...]
                        list of cmds to authorize

$ tacacs_client account -h
usage: tacacs_client account [-h] -c CMDS [CMDS ...] -f {start,stop,update}

optional arguments:
  -h, --help            show this help message and exit
  -c CMDS [CMDS ...], --cmds CMDS [CMDS ...]
                        list of cmds to authorize
  -f {start,stop,update}, --flag {start,stop,update}
                        accounting flag

```

1.2 Programmatic Usage

```

#!/usr/bin/env python
from tacacs_plus.client import TACACSClient
from tacacs_plus.flags import TAC_PLUS_ACCT_FLAG_START, TAC_PLUS_ACCT_FLAG_WATCHDOG,
↪TAC_PLUS_ACCT_FLAG_STOP

cli = TACACSClient('host', 49, 'secret', timeout=10)

# authenticate user and pass
authen = cli.authenticate('username', 'password')
print "PASS!" if authen.valid else "FAIL!"

# authorize user and command
author = cli.authorize('username', arguments=[b"service=shell", b"cmd=show", b
↪"cmdargs=version"])
print "PASS!" if author.valid else "FAIL!"

# start accounting session for command
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_START, arguments=[b"service=shell",
↪b"cmd=show", b"cmdargs=version"])
print "PASS!" if acct.valid else "FAIL!"

# continue accounting session for another command
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_WATCHDOG, arguments=[b"service=shell
↪", b"cmd=debug", b"cmdargs=aaa"])
print "PASS!" if acct.valid else "FAIL!"

# close accounting session
acct = cli.account('username', TAC_PLUS_ACCT_FLAG_STOP, arguments=[b"service=shell", b
↪"cmd=exit"])
print "PASS!" if acct.valid else "FAIL!"

```


CHAPTER 2

API documentation

This is the tacacs_plus API documentation. It contains the documentation extracted from the docstrings of the various classes, methods, and functions in the tacacs_plus package. If you want to know what a certain function/method does, this is the place to look.

Contents

- *API documentation*
 - *tacacs_plus.client module*
 - *tacacs_plus.packet module*
 - *tacacs_plus.authentication module*
 - *tacacs_plus.authorization module*
 - *tacacs_plus.accounting module*
 - *tacacs_plus.flags module*

2.1 tacacs_plus.client module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/client.py

class tacacs_plus.client.**TACACSClient** (*host, port, secret, timeout=10, session_id=None, family=2, version_max=12, version_min=0*)

A TACACS+ authentication client. <https://datatracker.ietf.org/doc/draft-ietf-opsawg-tacacs>

An open source TACACS+ server daemon is available at http://www.shrubbery.net/tac_plus/

account (*username, flags, arguments=[], authen_type=1, priv_lvl=0, rem_addr='python_device', port='python_tty0'*)

Account with a TACACS+ server.

Parameters

- **username** –
- **flags** – TAC_PLUS_ACCT_FLAG_START, TAC_PLUS_ACCT_FLAG_WATCHDOG, TAC_PLUS_ACCT_FLAG_STOP
- **arguments** – The authorization arguments
- **authen_type** – TAC_PLUS_AUTHEN_TYPE_ASCII, TAC_PLUS_AUTHEN_TYPE_PAP, TAC_PLUS_AUTHEN_TYPE_CHAP
- **priv_lvl** – Minimal Required priv_lvl.
- **rem_addr** – AAA request source, default to TAC_PLUS_VIRTUAL_REM_ADDR
- **port** – AAA port, default to TAC_PLUS_VIRTUAL_PORT

Returns TACACSAccountingReply**Raises** socket.timeout, socket.error

authenticate (*username*, *password*, *priv_lvl*=0, *authen_type*=1, *chap_ppp_id*=None, *chap_challenge*=None, *rem_addr*='python_device', *port*='python_tty0')

Authenticate to a TACACS+ server with a username and password.

Parameters

- **username** –
- **password** –
- **priv_lvl** –
- **authen_type** – TAC_PLUS_AUTHEN_TYPE_ASCII, TAC_PLUS_AUTHEN_TYPE_PAP, TAC_PLUS_AUTHEN_TYPE_CHAP
- **chap_ppp_id** – PPP ID when authen_type == 'chap'
- **chap_challenge** – challenge value when authen_type == 'chap'
- **rem_addr** – AAA request source, default to TAC_PLUS_VIRTUAL_REM_ADDR
- **port** – AAA port, default to TAC_PLUS_VIRTUAL_PORT

Returns TACACSAuthenticationReply**Raises** socket.timeout, socket.error

authorize (*username*, *arguments*=[], *authen_type*=1, *priv_lvl*=0, *rem_addr*='python_device', *port*='python_tty0')

Authorize with a TACACS+ server.

Parameters

- **username** –
- **arguments** – The authorization arguments
- **authen_type** – TAC_PLUS_AUTHEN_TYPE_ASCII, TAC_PLUS_AUTHEN_TYPE_PAP, TAC_PLUS_AUTHEN_TYPE_CHAP
- **priv_lvl** – Minimal Required priv_lvl.
- **rem_addr** – AAA request source, default to TAC_PLUS_VIRTUAL_REM_ADDR
- **port** – AAA port, default to TAC_PLUS_VIRTUAL_PORT

Returns TACACSAuthenticationReply

Raises socket.timeout, socket.error

closing (***kws*)

send (*body, req_type, seq_no=1*)

Send a TACACS+ message body

Parameters

- **body** – packed bytes, i.e., *struct.pack(...)*
- **req_type** – TAC_PLUS_AUTHEN, TAC_PLUS_AUTHOR, TAC_PLUS_ACCT
- **seq_no** – The sequence number of the current packet. The first packet in a session **MUST** have the sequence number 1 and each subsequent packet will increment the sequence number by one. Thus clients only send packets containing odd sequence numbers, and TACACS+ servers only send packets containing even sequence numbers.

Returns TACACSPacket

Raises socket.timeout, socket.error

sock

version

2.2 tacacs_plus.packet module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/packet.py

class tacacs_plus.client.TACACSHeader (*version, type, session_id, length, seq_no=1, flags=0*)

packed

classmethod unpacked (*raw*)

version_max

version_min

class tacacs_plus.client.TACACSPacket (*header, body_bytes, secret*)

body

crypt

encrypted

seq_no

2.3 tacacs_plus.authentication module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/authentication.py

class tacacs_plus.authentication.TACACSAuthenticationStart (*username, auth_type, priv_lvl=0, data="", rem_addr='python_device', port='python_tty0'*)

```
    packed
class tacacs_plus.authentication.TACACSAuthenticationContinue (password,
                                                                data="", flags=0)

    packed
class tacacs_plus.authentication.TACACSAuthenticationReply (status,          flags,
                                                            server_msg, data)

    error
    getpass
    human_status
    invalid
    classmethod unpacked (raw)
    valid
```

2.4 tacacs_plus.authorization module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/authorization.py

```
class tacacs_plus.authorization.TACACSAuthorizationStart (username,          au-
                                                            then_method,  priv_lvl,
                                                            authn_type,  arguments,
                                                            rem_addr='python_device',
                                                            port='python_tty0')

    packed
class tacacs_plus.authorization.TACACSAuthorizationReply (status,          arg_cnt,
                                                            server_msg,    data,
                                                            arguments)

    error
    follow
    human_status
    invalid
    reply
    classmethod unpacked (raw)
    valid
```

2.5 tacacs_plus.accounting module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/accounting.py

```
class tacacs_plus.accounting.TACACSAccountingStart (username, flags, authen_method,
                                                    priv_lvl, authen_type, arguments,
                                                    rem_addr='python_device',
                                                    port='python_tty0')

    packed

class tacacs_plus.accounting.TACACSAccountingReply (status, server_msg, data)

    error
    follow
    human_status
    classmethod unpacked (raw)
    valid
```

2.6 tacacs_plus.flags module

source: https://github.com/ansible/tacacs_plus/blob/master/tacacs_plus/flags.py

this module contains all the constant flags used to implement the tacacs+ RFC.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

A

account () (tacacs_plus.client.TACACSClient method), 5
authenticate () (tacacs_plus.client.TACACSClient method), 6
authorize () (tacacs_plus.client.TACACSClient method), 6

B

body (tacacs_plus.client.TACACSPacket attribute), 7

C

closing () (tacacs_plus.client.TACACSClient method), 7
crypt (tacacs_plus.client.TACACSPacket attribute), 7

E

encrypted (tacacs_plus.client.TACACSPacket attribute), 7
error (tacacs_plus.accounting.TACACSAccountingReply attribute), 9
error (tacacs_plus.authentication.TACACSAuthenticationReply attribute), 8
error (tacacs_plus.authorization.TACACSAuthorizationReply attribute), 8

F

follow (tacacs_plus.accounting.TACACSAccountingReply attribute), 9
follow (tacacs_plus.authorization.TACACSAuthorizationReply attribute), 8

G

getpass (tacacs_plus.authentication.TACACSAuthenticationReply attribute), 8

H

human_status (tacacs_plus.accounting.TACACSAccountingReply attribute), 9

human_status (tacacs_plus.authentication.TACACSAuthenticationReply attribute), 8
human_status (tacacs_plus.authorization.TACACSAuthorizationReply attribute), 8

I

invalid (tacacs_plus.authentication.TACACSAuthenticationReply attribute), 8
invalid (tacacs_plus.authorization.TACACSAuthorizationReply attribute), 8

P

packed (tacacs_plus.accounting.TACACSAccountingStart attribute), 9
packed (tacacs_plus.authentication.TACACSAuthenticationContinue attribute), 8
packed (tacacs_plus.authentication.TACACSAuthenticationStart attribute), 7
packed (tacacs_plus.authorization.TACACSAuthorizationStart attribute), 8
packed (tacacs_plus.client.TACACSHeader attribute), 7

R

reply (tacacs_plus.authorization.TACACSAuthorizationReply attribute), 8

S

send () (tacacs_plus.client.TACACSClient method), 7
seq_no (tacacs_plus.client.TACACSPacket attribute), 7
sock (tacacs_plus.client.TACACSClient attribute), 7

T

TACACSAccountingReply (class in tacacs_plus.accounting), 9
TACACSAccountingStart (class in tacacs_plus.accounting), 8
TACACSAuthenticationContinue (class in tacacs_plus.authentication), 8

TACACSAuthenticationReply (class in *tacacs_plus.authentication*), 8
TACACSAuthenticationStart (class in *tacacs_plus.authentication*), 7
TACACSAuthorizationReply (class in *tacacs_plus.authorization*), 8
TACACSAuthorizationStart (class in *tacacs_plus.authorization*), 8
TACACSClient (class in *tacacs_plus.client*), 5
TACACSHeader (class in *tacacs_plus.client*), 7
TACACSPacket (class in *tacacs_plus.client*), 7

U

unpacked() (*tacacs_plus.accounting.TACACSAccountingReply* class method), 9
unpacked() (*tacacs_plus.authentication.TACACSAuthenticationReply* class method), 8
unpacked() (*tacacs_plus.authorization.TACACSAuthorizationReply* class method), 8
unpacked() (*tacacs_plus.client.TACACSHeader* class method), 7

V

valid(*tacacs_plus.accounting.TACACSAccountingReply* attribute), 9
valid(*tacacs_plus.authentication.TACACSAuthenticationReply* attribute), 8
valid(*tacacs_plus.authorization.TACACSAuthorizationReply* attribute), 8
version (*tacacs_plus.client.TACACSClient* attribute), 7
version_max (*tacacs_plus.client.TACACSHeader* attribute), 7
version_min (*tacacs_plus.client.TACACSHeader* attribute), 7