Яндекс

PGBootCamp.ru

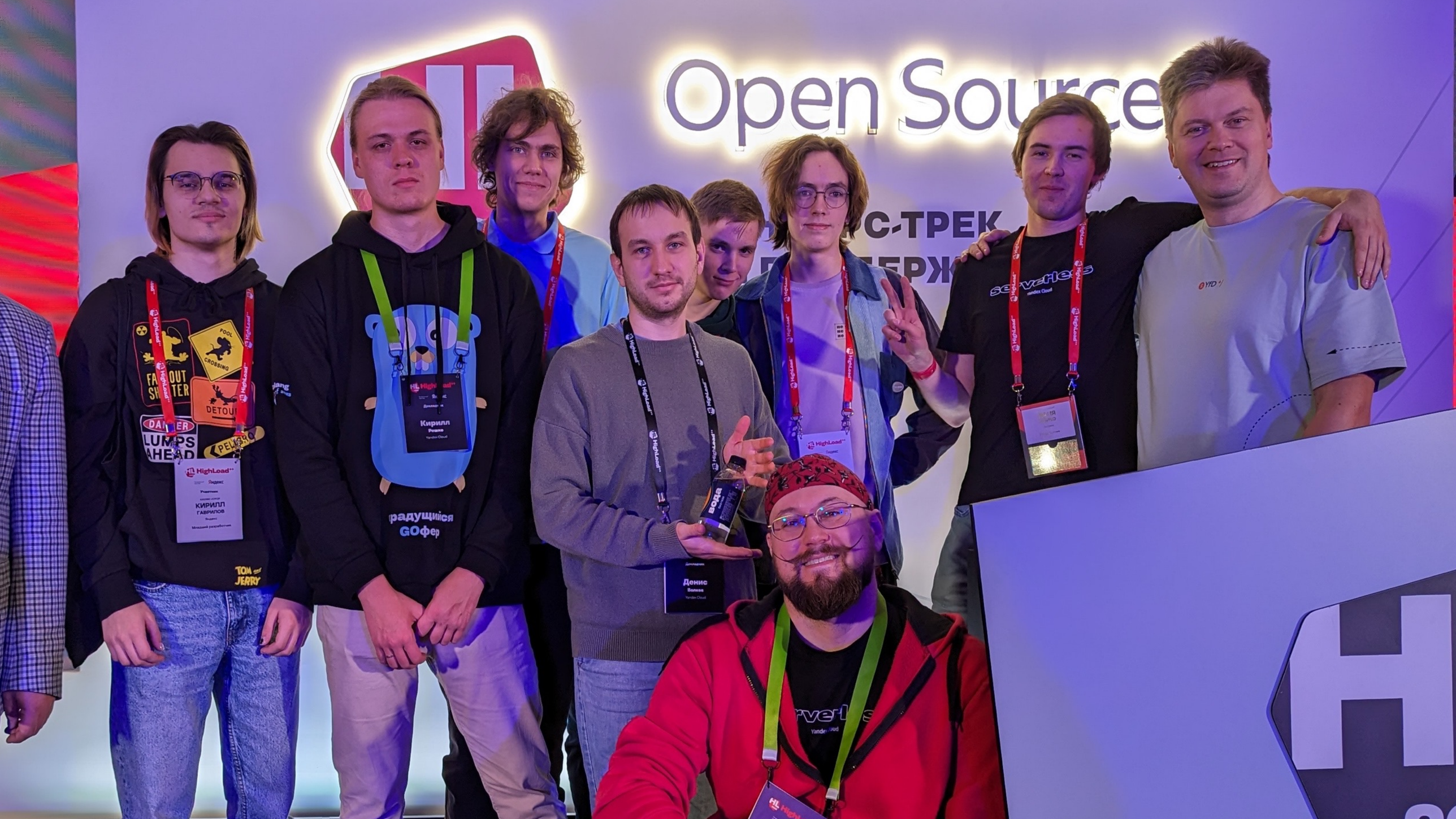
Проблемы PQ

Андрей Бородин, Postgres Contributor

Разрабатывал Postgres, GreenplumDB, Redshift

Мэйнтейнер:

- Odyssey
- WAL-G
- SPQR


Open Source
С-ТРЕК
Кирилл
Решко
Yandex Cloud
радущийся
GOфер
Денис
Волков
Yandex Cloud
serverless
Yandex Cloud
YYD

Чего хотят разработчики?

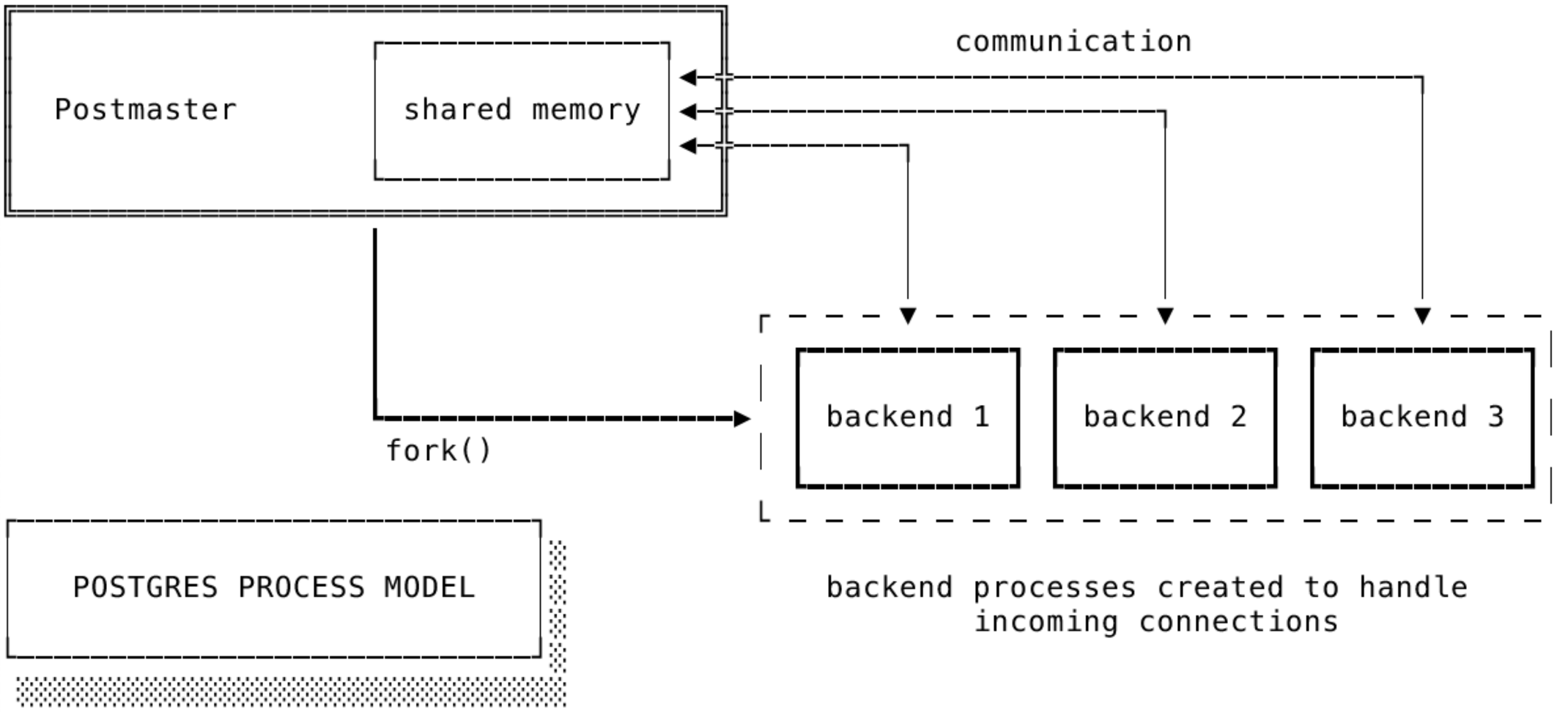
- **Request-response**

Чего хотят разработчики?

Request-response

А не:

- Startup cost
- Dealing with reconnection
- Dealing with partial state
- Managing swarm of connections in app


Postmaster
shared memory
communication
fork()
backend 1
backend 2
backend 3
POSTGRES PROCESS MODEL
backend processes created to handle
incoming connections

Postgres on the wire

A look at the PostgreSQL wire protocol

Jan Urbański
j.urbanski@wulczer.org

Ducksboard

PGCon 2014, Ottawa, May 23

https://beta.pgcon.org/2014/schedule/attachments/330_postgres-for-the-wire.pdf

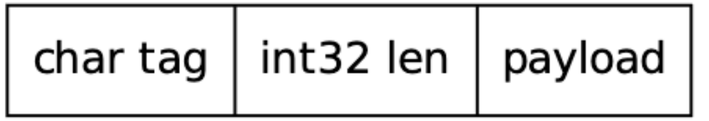
Protocol versions

- the 2.0 protocol got introduced in 6.4, around 1999
  - protocol versioning got added in the previous release
- the 3.0 got introduced in 7.4, in 2003
- the server still supports protocol 1.0!
- 3.0 has some new features
  - extended query protocol
  - COPY improvements
  - overall better frame structure

FEBE frame format

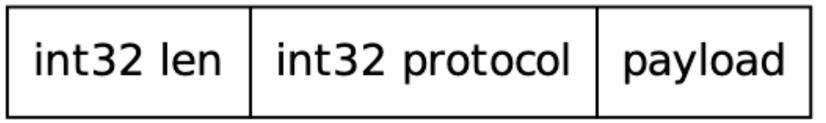
Virtually all messages start with an ASCII identifier, followed by length and payload.

Regular packet



The exception is the startup packet, which starts with the length followed by the protocol version.

Startup packet



```go
func (p *PgFortuneBackend) handleStartup() error {
	startupMessage, err := p.backend.ReceiveStartupMessage()
	if err != nil {
		return fmt.Errorf("error receiving startup message: %w", err)
	}

	switch startupMessage.(type) {
	case *pgproto3.StartupMessage:
		buf := mustEncode((&pgproto3.AuthenticationOk{}).Encode(nil))
		buf = mustEncode((&pgproto3.ReadyForQuery{TxStatus: 'I'}).Encode(buf))
		_, err = p.conn.Write(buf)
		if err != nil {
			return fmt.Errorf("error sending ready for query: %w", err)
		}
	case *pgproto3.SSLRequest:
		_, err = p.conn.Write([]byte("N"))
		if err != nil {
			return fmt.Errorf("error sending deny SSL request: %w", err)
		}
		return p.handleStartup()
	default:
		return fmt.Errorf("unknown startup message: %#v", startupMessage)
	}

	return nil
}
```

https://github.com/jackc/pgx/blob/master/pgproto3/example/pgfortune/server.go

Cancellation

Cancel request

int32 len	int32 cancelcode	int32 pid	int32 secret

- the cancel key is transmitted by the server upon connection
- cancelling queries requires opening separate connection
- another dummy protocol version is sent to ask for cancellation
- the cancellation message includes the process ID and a 32 bit key
  - theoretically open to replay attacks, but can be sent over SSL
  - libpq does not, so most applications will transmit it in the open

Handling errors

ErrorResponse

'E'	int32 len	char code	str value	\0	char code	str value	\0	...	\0

- the ErrorResponse message is sent for all kinds of errors
  - both for authentication errors and client errors
- it is a list of key-value fields
  - in 2.0 it was just a string, in 3.0 it has structure
  - example error fields are: message, detail, hint and error position
  - detailed down to the source file and line, great for fingerprinting

Simple query frames

Query

'Q'	int32 len	str query

RowDescription

'T'	int32 len	int16 numfields

+

str col	int32 tableoid	int16 colno	int32 typeoid	int16 typelen	int32 typmod	int16 format

. . .

CommandComplete

'C'	int32 len	str tag

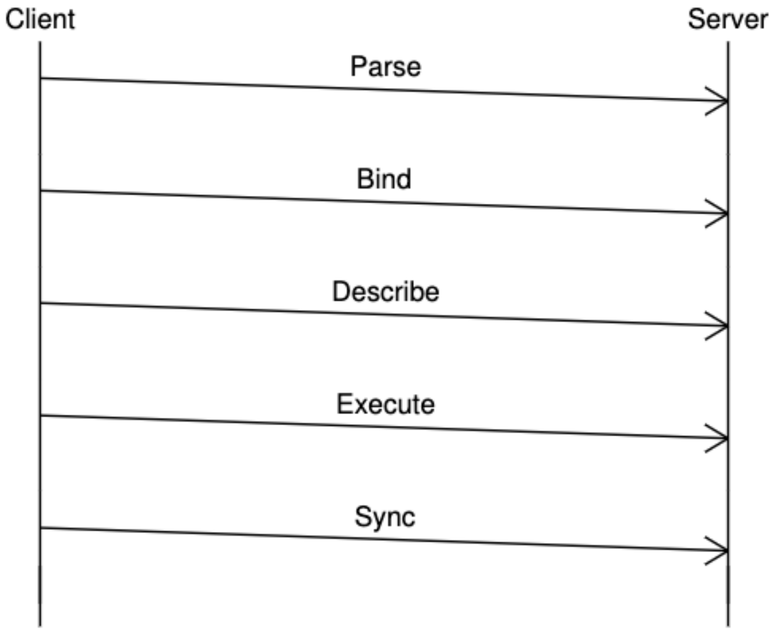
DataRow

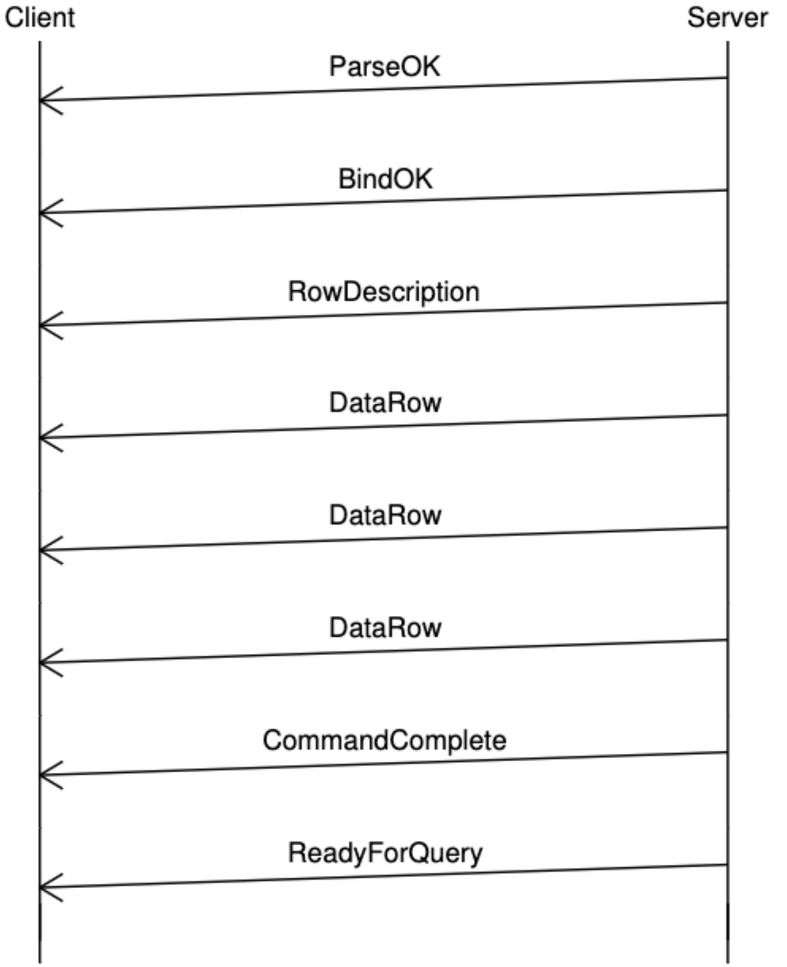
'D'	int32 len	int16 numfields	int32 fieldlen	char[fieldlen] data	...

ReadyForQuery

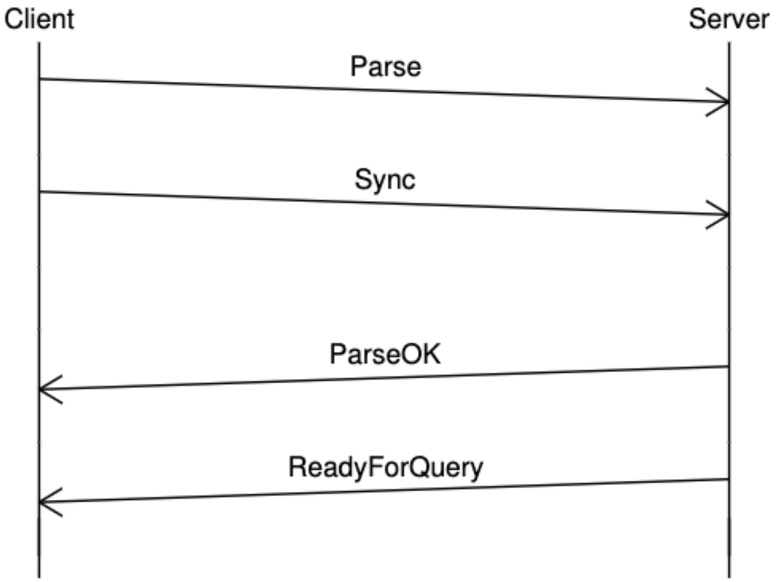
'Z'	int32 len	'I' or 'T' or 'E'

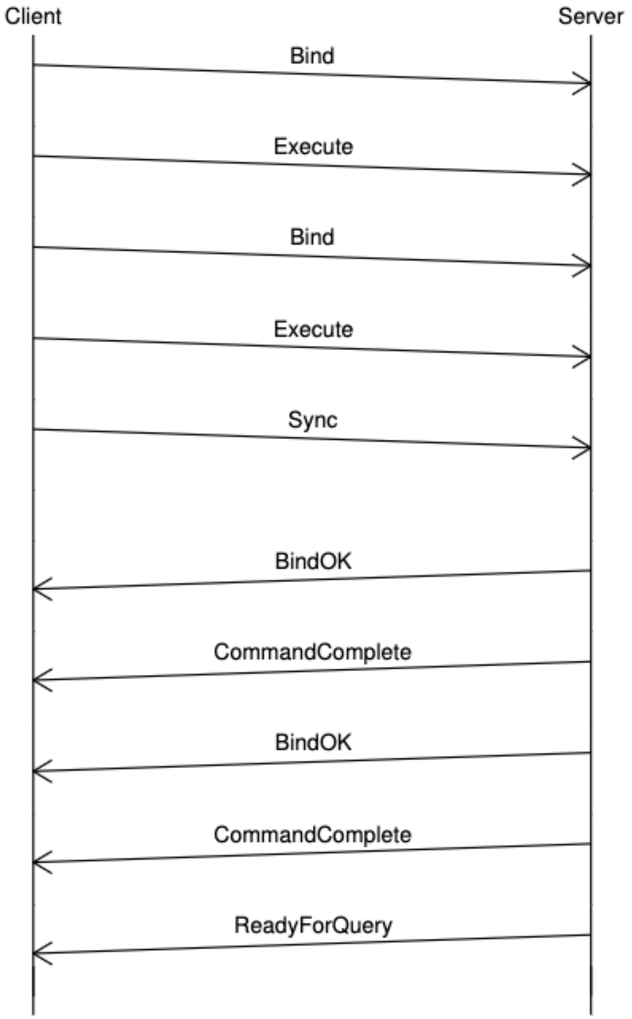
Extended query protocol flow



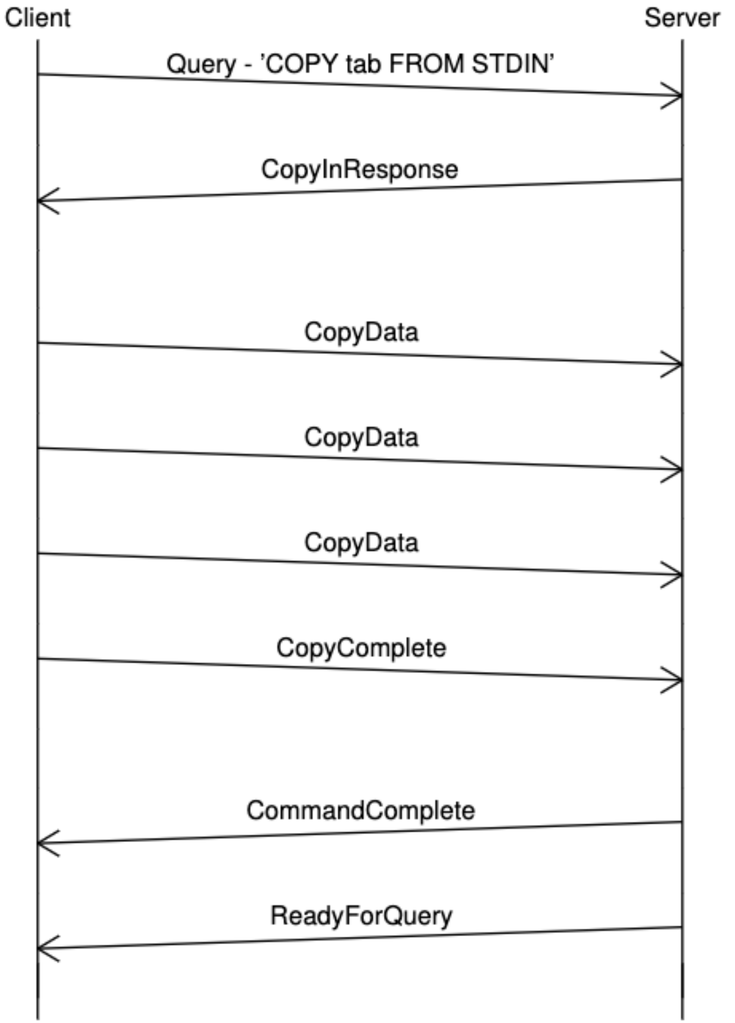


Advanced extended protocol usage

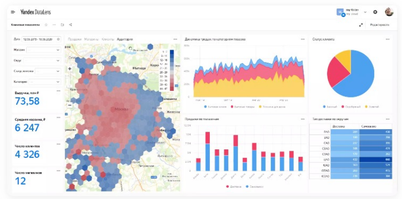




COPY subprotocol flow



Топология кластера в Облаке



OLAP\MR

RW Queries

Logical replication

WAL

Yandex
Object
Storage

Backup

Async replication

Sync replication

RO Queries

Что делает PQ?

- Запрос-ответ
  text\binary formats

Что делает PQ?

- Запрос-ответ
- FDW
- Notices

Что делает PQ?

- Запрос-ответ
- FDW
- Notices
- Listen-Notify

Что делает PQ?

- Запрос-ответ
- FDW
- Notices
- Listen-Notify
- Basebackup

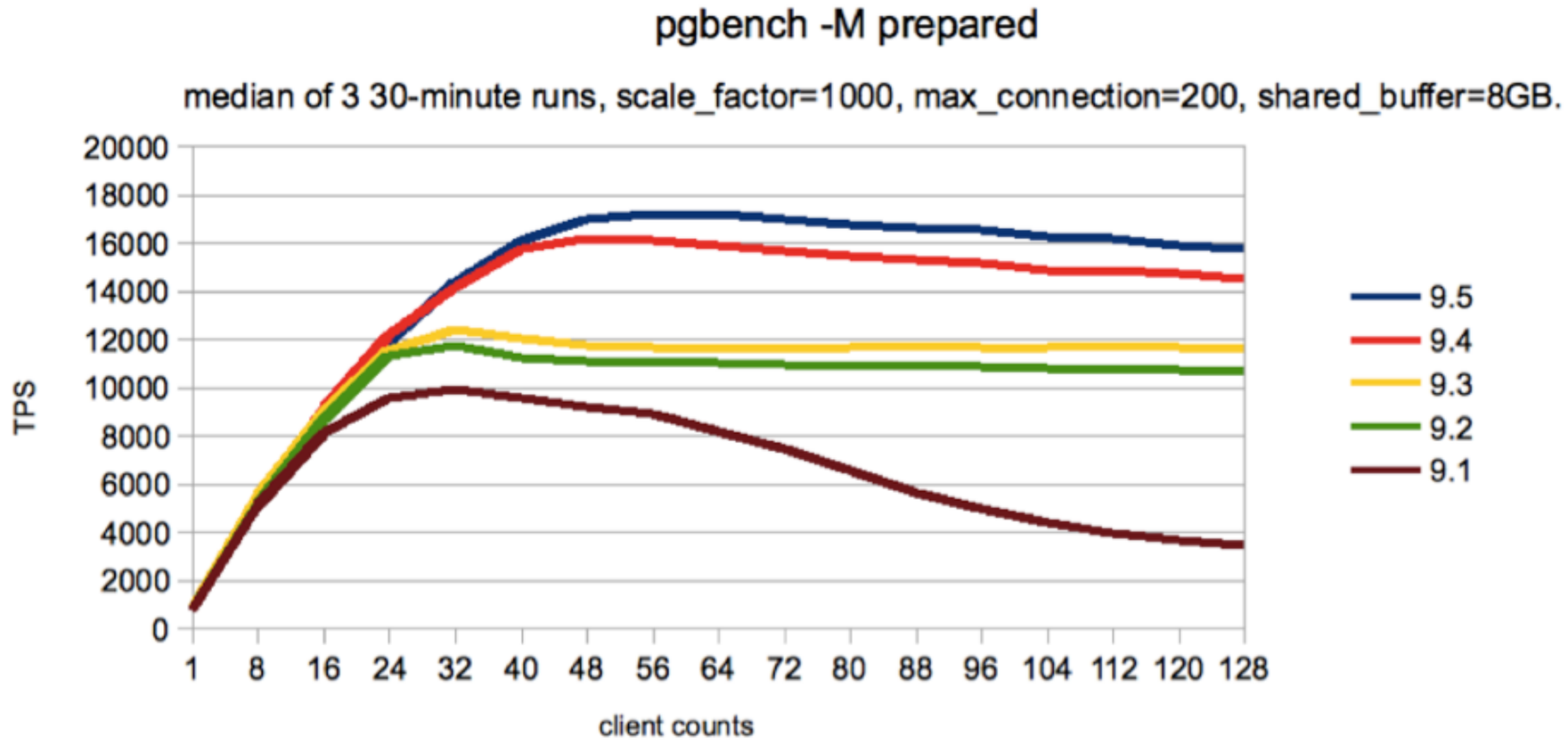
Что делает PQ?

- Запрос-ответ
- FDW
- Notices
- Listen-Notify
- Basebackup
- Walsender\Walreciever

Что делает PQ?

- Запрос-ответ
- FDW
- Notices
- Listen-Notify
- Basebackup
- Walsender\Walreciever
- Pg_rewind
- Proxies!

Proxies



https://www.enterprisedb.com/blog/powering-postgres-performance-advances

Proxies do more!

- PAUSE + Upgrade
- Retry
- Sharding
- Encryption termination

etc

Auth methods

- MD5 auth
- SCRAM-Auth
- Cert auth
- PAM
- LDAP

CRIME attack: secrets and user data may be compressed together

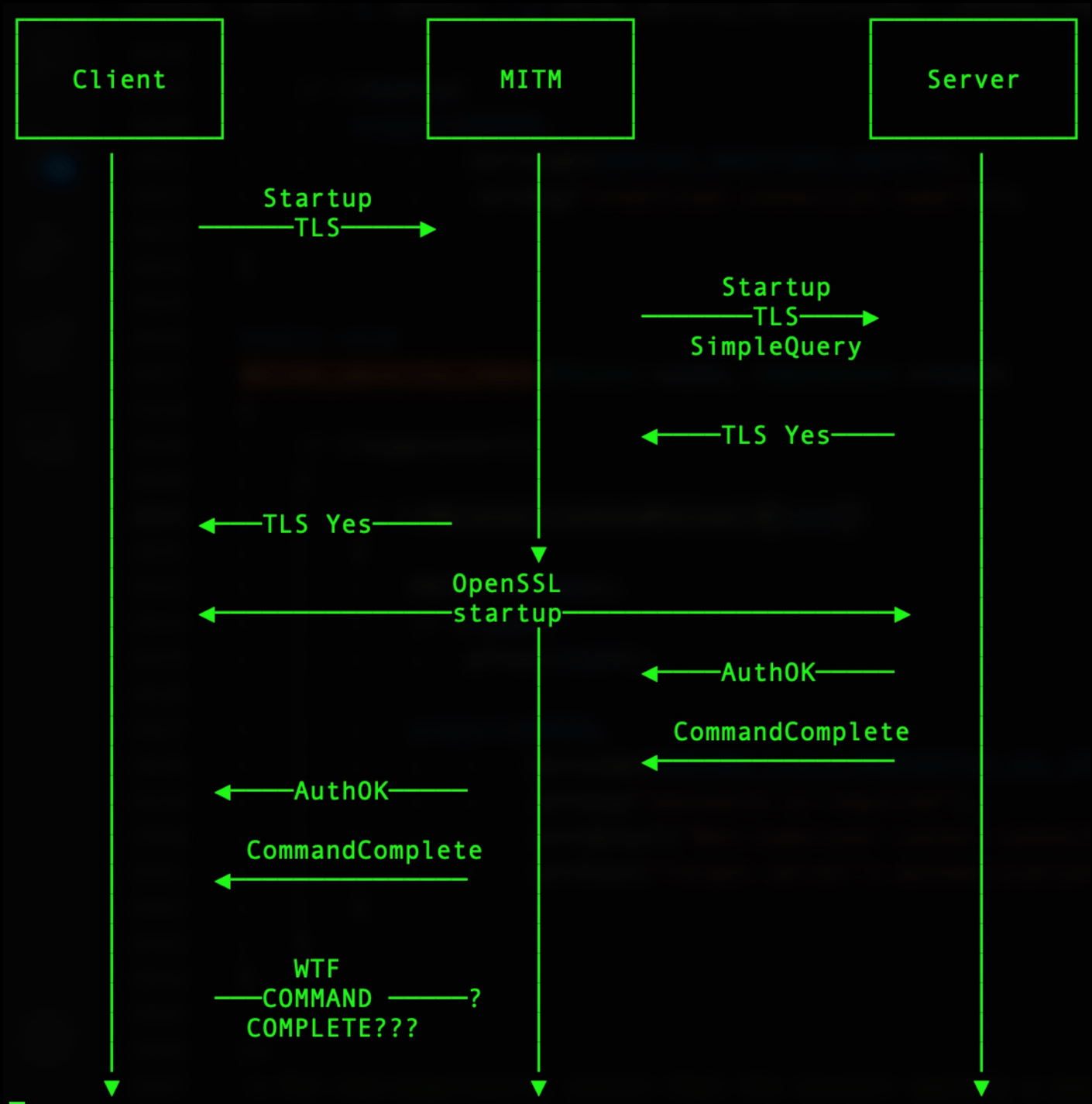
- Startup (username: website, password: secret_data)
- SimpleQuery(“select * from news where user_id = $1”, user_data)

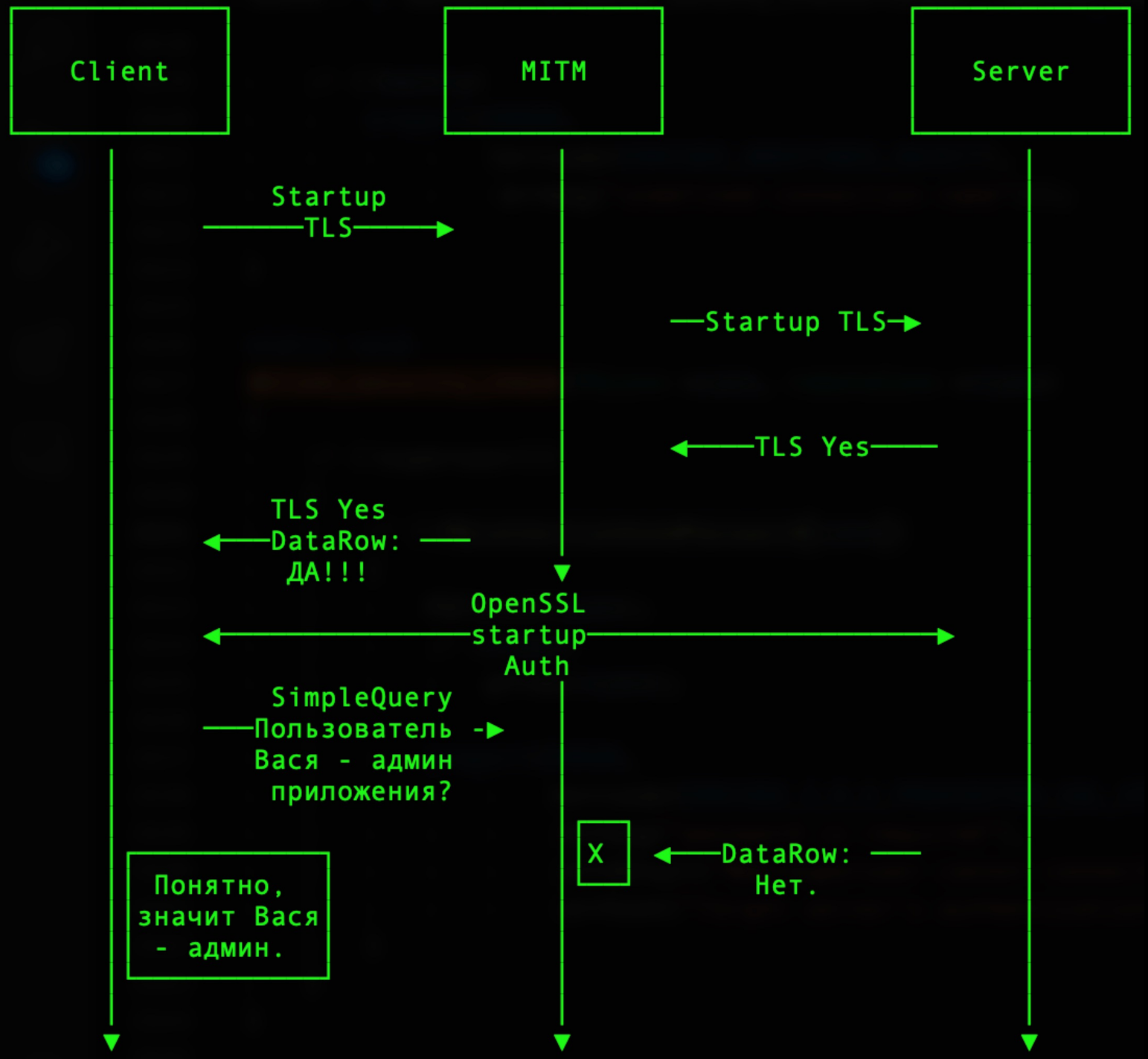
Length of resulting package reveals if user data is similar to secret data

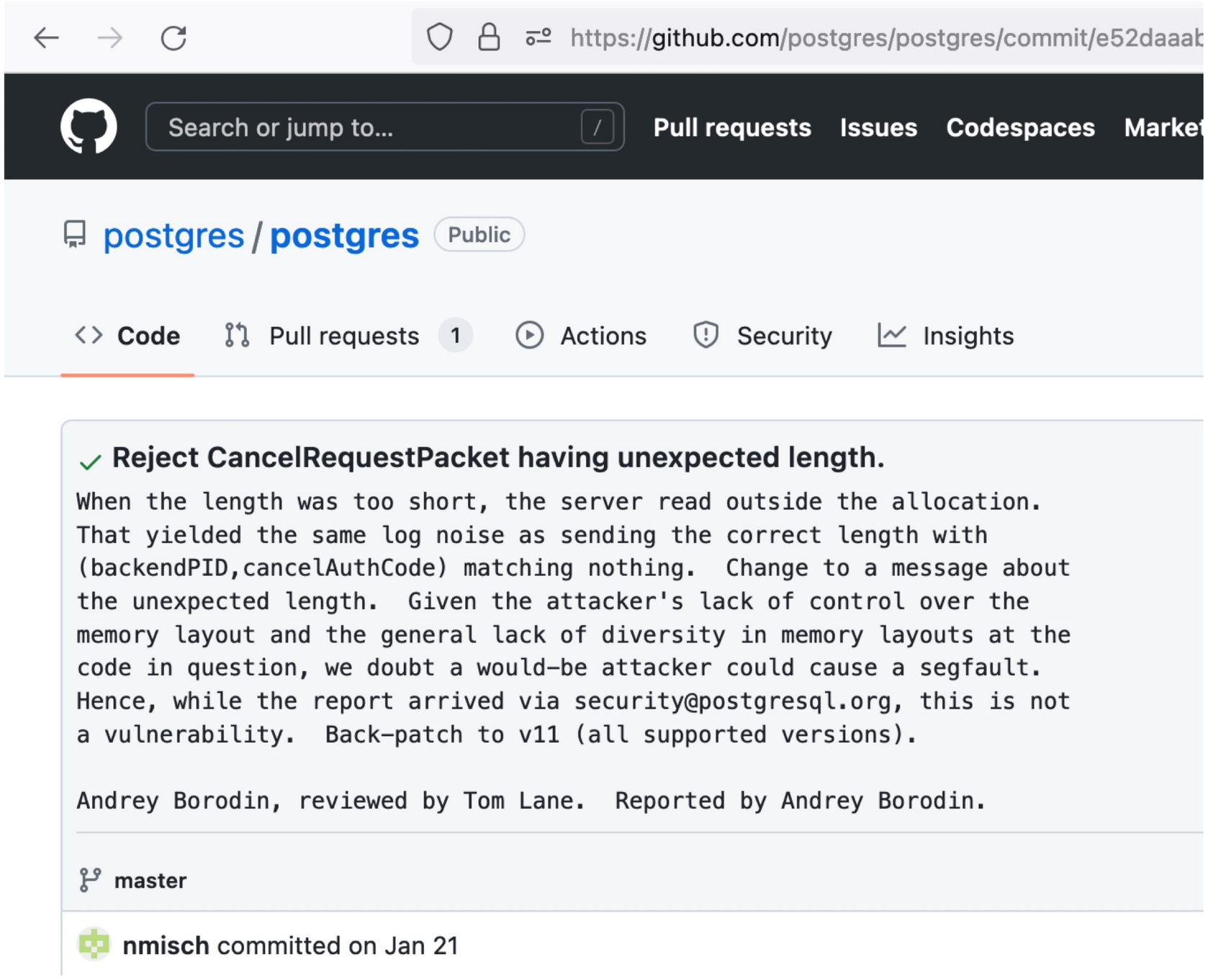
CVE-2021-23214: TLS аутентификация **8.1**

Fixed in 14.1, 13.5, 12.9,11.14,10.19, 9.6.24 (11 ноября 2021)

X


Client
MITM
Server
Startup
TLS
Startup
TLS
SimpleQuery
TLS Yes
TLS Yes
OpenSSL
startup
AuthOK
CommandComplete
AuthOK
CommandComplete
WTF
COMMAND
COMPLETE???


Client
MITM
Server
Startup TLS
Startup TLS
TLS Yes
TLS Yes DataRow: ДА!!!
OpenSSL startup Auth
SimpleQuery Пользователь Вася - админ приложения?
X
DataRow: Нет.
Понятно, значит Вася - админ.


https://github.com/postgres/postgres/commit/e52daaab
Search or jump to...
/
Pull requests
Issues
Codespaces
Market
postgres / postgres
Public
Code
Pull requests 1
Actions
Security
Insights
Reject CancelRequestPacket having unexpected length.
When the length was too short, the server read outside the allocation.
That yielded the same log noise as sending the correct length with
(backendPID,cancelAuthCode) matching nothing. Change to a message about
the unexpected length. Given the attacker's lack of control over the
memory layout and the general lack of diversity in memory layouts at the
code in question, we doubt a would-be attacker could cause a segfault.
Hence, while the report arrived via security@postgresql.org, this is not
a vulnerability. Back-patch to v11 (all supported versions).
Andrey Borodin, reviewed by Tom Lane. Reported by Andrey Borodin.
master
nmisch committed on Jan 21

7 src/backend/postmaster/postmaster.c

```
@@ -2016,6 +2016,13 @@ ProcessStartupPacket(Port *port, bool ssl_done, bool gss_done)
2016 2016
2017 2017       if (proto == CANCEL_REQUEST_CODE)
2018 2018       {
     2019 +         if (len != sizeof(CancelRequestPacket))
     2020 +         {
     2021 +             ereport(COMMERROR,
     2022 +                     (errcode(ERRCODE_PROTOCOL_VIOLATION),
     2023 +                      errmsg("invalid length of startup packet")));
     2024 +             return STATUS_ERROR;
     2025 +         }
2019 2026           processCancelRequest(port, buf);
2020 2027           /* Not really an error, but we don't want to proceed further */
2021 2028           return STATUS_ERROR;
```

Яндекс

PGBootCamp.ru

Спасибо!

Андрей Бородин

Postgres Contributor