# PQConnect

https://www.pqconnect.net

**Daniel J. Bernstein**

joint work with:

Tanja Lange, Jonathan Levin, Bo-Yin Yang

# CVE-2025-21617

"Guzzle OAuth Subscriber signs Guzzle requests using OAuth 1.0. Prior to 0.8.1, Nonce generation does not use sufficient entropy nor a cryptographically secure pseudorandom source. This can leave servers vulnerable to replay attacks when TLS is not used. This vulnerability is fixed in 0.8.1."

# CVE-2024-9355

"A vulnerability was found in Golang FIPS OpenSSL. This flaw allows a malicious user to randomly cause an uninitialized buffer length variable with a zeroed buffer to be returned in FIPS mode. It may also be possible to force a false positive match between non-equal hashes when comparing a trusted computed hmac sum to an untrusted input sum if an attacker can send a zeroed buffer in place of a pre-computed sum. It is also possible to force a derived key to be all zeros instead of an unpredictable value. This may have follow-on implications for the Go TLS stack."

# CVE-2024-8287

"Anbox Management Service, in versions 1.17.0 through 1.23.0, does not validate the TLS certificate provided to it by the Anbox Stream Agent. An attacker must be able to machine-in-the-middle the Anbox Stream Agent from within an internal network before they can attempt to take advantage of this."

# CVE-2024-8285

"A flaw was found in Kroxylicious. When establishing the connection with the upstream Kafka server using a TLS secured connection, Kroxylicious fails to properly verify the server's hostname, resulting in an insecure connection. For a successful attack to be performed, the attacker needs to perform a Man-in-the-Middle attack or compromise any external systems, such as DNS or network routing configuration. This issue is considered a high complexity attack, with additional high privileges required, as the attack would need access to the Kroxylicious configuration or a peer system. The result of a successful attack impacts both data integrity and confidentiality."

# CVE-2024-8096

"When curl is told to use the Certificate Status Request TLS extension, often referred to as OCSP stapling, to verify that the server certificate is valid, it might fail to detect some OCSP problems and instead wrongly consider the response as fine. If the returned status reports another error than 'revoked' (like for example 'unauthorized') it is not treated as a bad certficate."

# CVE-2024-8007

"A flaw was found in the openstack-tripleo-common component of the Red Hat OpenStack Platform (RHOSP) director. This vulnerability allows an attacker to deploy potentially compromised container images via disabling TLS certificate verification for registry mirrors, which could enable a man-in-the-middle (MITM) attack."

# CVE-2024-7383

"A flaw was found in libnbd. The client did not always correctly verify the NBD server's certificate when using TLS to connect to an NBD server. This issue allows a man-in-the-middle attack on NBD traffic."

# CVE-2024-7346

"Host name validation for TLS certificates is bypassed when the installed OpenEdge default certificates are used to perform the TLS handshake for a networked connection. This has been corrected so that default certificates are no longer capable of overriding host name validation and will need to be replaced where full TLS certificate validation is needed for network security. The existing certificates should be replaced with CA-signed certificates from a recognized certificate authority that contain the necessary information to support host name validation."

# CVE-2024-7206

"SSL Pinning Bypass in eWeLink Some hardware products allows local ATTACKER to Decrypt TLS communication and Extract secrets to clone the device via Flash the modified firmware"

# CVE-2024-6119

In OpenSSL: “Applications performing certificate name checks (e.g., TLS clients checking server certificates) may attempt to read an invalid memory address resulting in abnormal termination of the application process.”

# TLS is a very large programming project

Broad TLS deployment requires integrating TLS into many protocols and many more applications.
Some ways to see how big this project is:

- Wide range of applications in TLS-related CVEs.

# TLS is a very large programming project

Broad TLS deployment requires integrating TLS into many protocols and many more applications.
Some ways to see how big this project is:

- Wide range of applications in TLS-related CVEs.
- Different programming languages for applications drive demand for TLS libraries in many languages.

# TLS is a very large programming project

Broad TLS deployment requires integrating TLS into many protocols and many more applications.
Some ways to see how big this project is:

- Wide range of applications in TLS-related CVEs.
- Different programming languages for applications drive demand for TLS libraries in many languages.
- GitHub search for "SSL" finds $>2$ million PRs. Spot-checks indicate that most of these really are SSL, not something else by the same name.

# TLS is a very large programming project

Broad TLS deployment requires integrating TLS into many protocols and many more applications.
Some ways to see how big this project is:

- Wide range of applications in TLS-related CVEs.
- Different programming languages for applications drive demand for TLS libraries in many languages.
- GitHub search for "SSL" finds >2 million PRs. Spot-checks indicate that most of these really are SSL, not something else by the same name.
- HTTPS percentage in Firefox web-page loads: about 30% in 2015, 80% in 2020, 80% in 2025. Do the other 20% not implement TLS? Not have it configured?

# We have a security problem right now

Large-scale attackers are recording ciphertexts today in the hope of breaking them with future quantum computers.

# We have a security problem right now

Large-scale attackers are recording ciphertexts today in the hope of breaking them with future quantum computers.

TLS response:

- Upgrade to post-quantum TLS.
- Also, for all of the data that isn't even encrypted today, keep working on increasing TLS deployment.
- Also, keep working on fixing TLS vulnerabilities.

# We have a security problem right now

Large-scale attackers are recording ciphertexts today in the hope of breaking them with future quantum computers.

TLS response:

- Upgrade to post-quantum TLS.
- Also, for all of the data that isn't even encrypted today, keep working on increasing TLS deployment.
- Also, keep working on fixing TLS vulnerabilities.

Hmmm. While this gigantic TLS project is continuing, is there anything else we can do to protect users?

# An easier path: VPNs

VPNs have a big software-engineering advantage:
they protect unmodified applications!

A VPN client typically routes all outgoing traffic through an encrypted tunnel to a proxy specified in the VPN config.

# An easier path: VPNs

VPNs have a big software-engineering advantage:
they protect unmodified applications!

A VPN client typically routes all outgoing traffic through an encrypted tunnel to a proxy specified in the VPN config.

VPNs supporting post-quantum crypto include Mullvad, Rosenpass, and VPNs based on OpenSSH (which in early 2022 upgraded `sntrup761` from experimental to the default KEX).

# But VPNs provide incomplete protection

The client's traffic is exposed to the VPN proxy, and is exposed on the network between the proxy and the server.

# But VPNs provide incomplete protection

The client's traffic is exposed to the VPN proxy, and is exposed on the network between the proxy and the server.

With more effort, you can add a specific server to the VPN config to create an end-to-end tunnel to that server.

# But VPNs provide incomplete protection

The client's traffic is exposed to the VPN proxy, and is exposed on the network between the proxy and the server.

With more effort, you can add a specific server to the VPN config to create an end-to-end tunnel to that server.

How often do users actually go to this effort?
How do they build and maintain the lists of supporting servers?

# A Boring Private Network (BPN): PQConnect

Like a VPN, PQConnect protects all applications.
Unlike a VPN, PQConnect *automatically* builds end-to-end post-quantum tunnels to any server that supports PQConnect.

# A Boring Private Network (BPN): PQConnect

Like a VPN, PQConnect protects all applications.
Unlike a VPN, PQConnect *automatically* builds end-to-end post-quantum tunnels to any server that supports PQConnect.

To set up PQConnect client: Install the PQConnect software.
No need for server-specific config.

# A Boring Private Network (BPN): PQConnect

Like a VPN, PQConnect protects all applications.
Unlike a VPN, PQConnect *automatically* builds end-to-end post-quantum tunnels to any server that supports PQConnect.

To set up PQConnect client: Install the PQConnect software. No need for server-specific config.

To set up PQConnect server: Install the PQConnect software; publish an announcement that the server name supports PQConnect. No need for client-specific config.

# A Boring Private Network (BPN): PQConnect

Like a VPN, PQConnect protects all applications.
Unlike a VPN, PQConnect *automatically* builds end-to-end post-quantum tunnels to any server that supports PQConnect.

To set up PQConnect client: Install the PQConnect software. No need for server-specific config.

To set up PQConnect server: Install the PQConnect software; publish an announcement that the server name supports PQConnect. No need for client-specific config.

A PQConnect client connecting to that name notices the announcement, automatically builds a tunnel to the server.

# How the PQConnect announcement works

Client connects to, e.g., bench.cr.yp.to. DNS data:

```
bench.cr.yp.to. 30 IN CNAME
  pq1...d185qfglcnsh273lv901sld020w020l0.cr.yp.to.

pq1...d185qfglcnsh273lv901sld020w020l0.cr.yp.to.
  30 IN A 131.193.32.110
```

Non-PQConnect client: "Connect to 131.193.32.110."
PQConnect client: "Aha, pq1... is telling me the server's PQConnect public key. I'll set up a PQConnect tunnel."

# Routing traffic from unmodified applications

The PQConnect software delivers modified DNS data locally:

```
bench.cr.yp.to. 30 IN CNAME
  pq1...d185qfglcnsh273lv901sld020w020l0.cr.yp.to.

pq1...d185qfglcnsh273lv901sld020w020l0.cr.yp.to.
  30 IN A 10.43.0.2
```

The application sends packets to 10.43.0.2, an address managed locally by the PQConnect software. The PQConnect software encrypts the packets to send through the tunnel.

# Addressing DNS forgery

Attacker forging `bench.cr.yp.to 30 IN A 214.29.60.3` breaks TLS security by obtaining a Let's Encrypt certificate. Also disables PQConnect.

Three PQConnect response strategies:

- To the extent that DNS security tools are rolled out, they automatically protect PQConnect announcements.
- PQConnect can also be used to protect DNS, because PQConnect applies to all packets.
- PQConnect supports high-security `pq1...` links.

# PQConnect resources

Linux software release+docs: https://www.pqconnect.net

New chat server: https://zulip.pqconnect.net
—sadly down this week because TU/e is under attack.

Paper to appear at NDSS 2025: "PQConnect: Automated Post-Quantum End-to-End Tunnels".