PQConnect:
Automated Post-Quantum End-to-End Tunnels

https://www.pqconnect.net

Jonathan Levin

joint work with:

Daniel J. Bernstein, Tanja Lange, Bo-Yin Yang


AB 1896

Shipwrecks in Network Security

CVEs

CVE-2025-23419

In NGINX: "When multiple server blocks are configured to share the same IP address and port, an attacker can use session resumption to bypass client certificate authentication requirements on these servers. This vulnerability arises when TLS Session Tickets are used and/or the SSL session cache are used in the default server and the default server is performing client certificate authentication."

CVE-2025-23114

"A vulnerability in Veeam Updater component allows Man-in-the-Middle attackers to execute arbitrary code on the affected server. This issue occurs due to a failure to properly validate TLS certificate."

The list...

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-2025-1146	CrowdStrike uses industry-standard TLS (transport layer security) to secure communications from the Falcon sensor to the CrowdStrike cloud. CrowdStrike has identified a validation logic error in the Falcon sensor for Linux, Falcon Kubernetes Admission Controller, and Falcon Container Sensor where our TLS connection routine to the CrowdStrike cloud can incorrectly process server certificate validation. This could allow an attacker with the ability to control network traffic to potentially conduct a man-in-the-middle (MITM) attack. CrowdStrike identified this issue internally and released a security fix in all Falcon sensor for Linux, Falcon Kubernetes Admission Controller, and Falcon Container Sensor versions 7.06 and above. CrowdStrike identified this issue through our longstanding, rigorous security review process, which has been continually strengthened with deeper source code analysis and ongoing program enhancements as part of our commitment to security resilience. CrowdStrike has no indication of any exploitation of this issue in the wild. CrowdStrike has leveraged its world class threat hunting and intelligence capabilities to actively monitor for signs of abuse or usage of this flaw and will continue to do so. Windows and Mac sensors are not affected by this.
CVE-2025-0343	Swift ASN.1 can be caused to crash when parsing certain BER/DER constructions. This crash is caused by a confusion in the ASN.1 library itself which assumes that certain objects can only be provided in either constructed or primitive forms, and will trigger a preconditionFailure if that constraint isn't met. Importantly, these constraints are actually required to be true in DER, but that correctness wasn't enforced on the early node parser side so it was incorrect to rely on it later on in decoding, which is what the library did. These crashes can be triggered when parsing any DER/BER format object. There is no memory-safety issue here: the crash is a graceful one from the Swift runtime. The impact of this is that it can be used as a denial-of-service vector when parsing BER/DER data from unknown sources, e.g. when parsing TLS certificates.
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-2024-8603	A “Use of a Broken or Risky Cryptographic Algorithm” vulnerability in the SSL/TLS component used in B&R Automation Runtime versions before 6.1 and B&R mapp View versions before 6.1 may be abused by unauthenticated network-based attackers to masquerade as services on impacted devices.
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.

...goes...

CVE-2024-6119	Issue summary: 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. Impact summary: Abnormal termination of an application can a cause a denial of service. Applications performing certificate name checks (e.g., TLS clients checking server certificates) may attempt to read an invalid memory address when comparing the expected name with an `otherName` subject alternative name of an X.509 certificate. This may result in an exception that terminates the application program. Note that basic certificate chain validation (signatures, dates, ...) is not affected, the denial of service can occur only when the application also specifies an expected DNS name, Email address or IP address. TLS servers rarely solicit client certificates, and even when they do, they generally don't perform a name check against a reference identifier (expected identity), but rather extract the presented identity after checking the certificate chain. So TLS servers are generally not affected and the severity of the issue is Moderate. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.
CVE-2024-5800	Diffie-Hellman groups with insufficient strength are used in the SSL/TLS stack of B&R Automation Runtime versions before 6.0.2, allowing a network attacker to decrypt the SSL/TLS communication.
CVE-2024-56128	Incorrect Implementation of Authentication Algorithm in Apache Kafka's SCRAM implementation. Issue Summary: Apache Kafka's implementation of the Salted Challenge Response Authentication Mechanism (SCRAM) did not fully adhere to the requirements of RFC 5802 [1]. Specifically, as per RFC 5802, the server must verify that the nonce sent by the client in the second message matches the nonce sent by the server in its first message. However, Kafka's SCRAM implementation did not perform this validation. Impact: This vulnerability is exploitable only when an attacker has plaintext access to the SCRAM authentication exchange. However, the usage of SCRAM over plaintext is strongly discouraged as it is considered an insecure practice [2]. Apache Kafka recommends deploying SCRAM exclusively with TLS encryption to protect SCRAM exchanges from interception [3]. Deployments using SCRAM with TLS are not affected by this issue. How to Detect If You Are Impacted: If your deployment uses SCRAM authentication over plaintext communication channels (without TLS encryption), you are likely impacted. To check if TLS is enabled, review your server.properties configuration file for listeners property. If you have SASL_PLAINTEXT in the listeners, then you are likely impacted. Fix Details: The issue has been addressed by introducing nonce verification in the final message of the SCRAM authentication exchange to ensure compliance with RFC 5802. Affected Versions: Apache Kafka versions 0.10.2.0 through 3.9.0, excluding the fixed versions below. Fixed Versions: 3.9.0 3.8.1 3.7.2 Users are advised to upgrade to 3.7.2 or later to mitigate this issue. Recommendations for Mitigation: Users unable to upgrade to the fixed versions can mitigate the issue by: - Using TLS with SCRAM Authentication: Always deploy SCRAM over TLS to encrypt authentication exchanges and protect against interception. - Considering Alternative Authentication Mechanisms: Evaluate alternative authentication mechanisms, such as PLAIN, Kerberos or OAuth with TLS, which provide additional layers of security.
CVE-2024-5535	Issue summary: Calling the OpenSSL API function SSL_select_next_proto with an empty supported client protocols buffer may cause a crash or memory contents to be sent to the peer. Impact summary: A buffer overread can have a range of potential consequences such as unexpected application beahviour or a crash. In particular this issue could result in up to 255 bytes of arbitrary private data from memory being sent to the peer leading to a loss of confidentiality. However, only applications that directly call the SSL_select_next_proto function with a 0 length list of supported client protocols are affected by this issue. This would normally never be a valid scenario and is typically not under attacker control but may occur by accident in the case of a configuration or programming error in the calling application. The OpenSSL API function SSL_select_next_proto is typically used by TLS applications that support ALPN (Application Layer Protocol Negotiation) or NPN (Next Protocol Negotiation). NPN is older, was never standardised and is deprecated in favour of ALPN. We believe that ALPN is significantly more widely deployed than NPN. The SSL_select_next_proto function accepts a list of protocols from the server and a list of protocols from the client and returns the first protocol that appears in the server list that also appears in the client list. In the case of no overlap between the two lists it returns the first item in the client list. In either case it will signal whether an overlap between the two lists was found. In the case where SSL_select_next_proto is called with a zero length client list it fails to notice this condition and returns the memory immediately following the client list pointer (and reports that there was no overlap in the lists). This function is typically called from a server side application callback for ALPN or a client side application callback for NPN. In the case of ALPN the list of protocols supplied by the client is guaranteed by libssl to never be zero in length. The list of server protocols comes from the application and should never normally be expected to be of zero length. In this case if the SSL_select_next_proto function has been called as expected (with the list supplied by the client passed in the client/client_len parameters), then the application will not be vulnerable to this issue. If the application has accidentally been configured with a zero length server list, and has accidentally passed that zero length server list in the client/client_len parameters, and has additionally failed to correctly handle a "no overlap" response (which would normally result in a handshake failure in ALPN) then it will be vulnerable to this problem. In the case of NPN, the protocol permits the client to opportunistically select a protocol when there is no overlap. OpenSSL returns the first client protocol in the no overlap case in support of this. The list of client protocols comes from the application and should never normally be expected to be of zero length. However if the SSL_select_next_proto function is accidentally called with a client_len of 0 then an invalid memory pointer will be returned instead. If the application uses this output as the opportunistic protocol then the loss of confidentiality will occur. This issue has been assessed as Low severity because applications are most likely to be vulnerable if they are using NPN instead of ALPN - but NPN is not widely used. It also requires an application configuration or programming error. Finally, this issue would not typically be under attacker control making active exploitation unlikely. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue. Due to the low severity of this issue we are not issuing new releases of OpenSSL at this time. The fix will be included in the next releases when they become available.
CVE-2024-5445	Ecosystem Agent version 4 < 4.1.5.2597 and Ecosystem Agent version 5 < 5.1.4.2473 did not properly validate SSL/TLS certificates, which could allow a malicious actor to perform a Man-in-the-Middle and intercept traffic between the agent and N-able servers from a privileged network position.
CVE-2024-53275	Home-Gallery.org is a self-hosted open-source web gallery to browse personal photos and videos. In 1.15.0 and earlier, the default setup of home-gallery is vulnerable to DNS rebinding. Home-gallery is set up without TLS and user authentication by default, leaving it vulnerable to DNS rebinding. In this attack, an attacker will ask a user to visit their website. The attacker website will then change the DNS records of their domain from their IP address to the internal IP address of the home-gallery instance. To tell which IP addresses are valid, we can rebind a subdomain to each IP address we want to check, and see if there is a response. Once potential candidates have been found, the attacker can launch the attack by reading the response of the web server after the IP address has changed. When the attacker domain is fetched, the response will be from the home-gallery instance, not the attacker website, because the IP address has been changed. Due to a lack of authentication, home-gallery photos can then be extracted by the attacker website.

...on...

CVE-2024-5288	An issue was discovered in wolfSSL before 5.7.0. A safe-error attack via Rowhammer, namely FAULT+PROBE, leads to ECDSA key disclosure. When WOLFSSL_CHECK_SIG_FAULTS is used in signing operations with private ECC keys, such as in server-side TLS connections, the connection is halted if any fault occurs. The success rate in a certain amount of connection requests can be processed via an advanced technique for ECDSA key recovery.
CVE-2024-5261	Improper Certificate Validation vulnerability in LibreOffice "LibreOfficeKit" mode disables TLS certification verification LibreOfficeKit can be used for accessing LibreOffice functionality through C/C++. Typically this is used by third party components to reuse LibreOffice as a library to convert, view or otherwise interact with documents. LibreOffice internally makes use of "curl" to fetch remote resources such as images hosted on webservers. In affected versions of LibreOffice, when used in LibreOfficeKit mode only, then curl's TLS certification verification was disabled (CURLOPT_SSL_VERIFYPEER of false) In the fixed versions curl operates in LibreOfficeKit mode the same as in standard mode with CURLOPT_SSL_VERIFYPEER of true. This issue affects LibreOffice before version 24.2.4.
CVE-2024-52330	ECOVACS lawnmowers and vacuums do not properly validate TLS certificates. An unauthenticated attacker can read or modify TLS traffic, possibly modifying firmware updates.
CVE-2024-52329	ECOVACS HOME mobile app plugins for specific robots do not properly validate TLS certificates. An unauthenticated attacker can read or modify TLS traffic and obtain authentication tokens.
CVE-2024-5148	A flaw was found in the gnome-remote-desktop package. The gnome-remote-desktop system daemon performs inadequate validation of session agents using D-Bus methods related to transitioning a client connection from the login screen to the user session. As a result, the system RDP TLS certificate and key can be exposed to unauthorized users. This flaw allows a malicious user on the system to take control of the RDP client connection during the login screen-to-user session transition.
CVE-2024-50692	SunGrow WiNet-S V200.001.00.P027 and earlier versions contains hardcoded MQTT credentials that allow an attacker to send arbitrary commands to an arbitrary inverter. It is also possible to impersonate the broker, because TLS is not used to identify the real MQTT broker. This means that MQTT communications are vulnerable to MitM attacks at the TCP/IP level.
CVE-2024-50046	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Prevent NULL-pointer dereference in nfs42_complete_copies() On the node of an NFS client, some files saved in the mountpoint of the NFS server were copied to another location of the same NFS server. Accidentally, the nfs42_complete_copies() got a NULL-pointer dereference crash with the following syslog: [232064.838881] NFSv4: state recovery failed for open file nfs/pvc-12b5200d-cd0f-46a3-b9f0-af8f4fe0ef64.qcow2, error = -116 [232064.839360] NFSv4: state recovery failed for open file nfs/pvc-12b5200d-cd0f-46a3-b9f0-af8f4fe0ef64.qcow2, error = -116 [232066.588183] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000058 [232066.588586] Mem abort info: [232066.588701] ESR = 0x0000000096000007 [232066.588862] EC = 0x25: DABT (current EL), IL = 32 bits [232066.589084] SET = 0, FnV = 0 [232066.589216] EA = 0, S1PTW = 0 [232066.589340] FSC = 0x07: level 3 translation fault [232066.589559] data abort info: [232066.589683] ISV = 0, ISS = 0x00000007 [232066.589842] CM = 0, WnR = 0 [232066.589967] user pgtable: 64k pages, 48-bit VAs, pgdp=0000200095eff400 [232066.590231] [0000000000000058] pgd=08001100ae100003, p4d=08001100ae100003, pud=08001100ae100003, pmd=08001100b3c00003, pte=0000000000000000 [232066.590757] Internal error: Oops: 96000007 [#1] SMP [232066.590958] Modules linked in: rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace fscache netfs ocfs2_dlmfs ocfs2_stack_o2cb ocfs2_dlm vhost_net vhost_iotlb tap tun ipt_rpfilter xt_multiport ip_set_hash_ip ip_set_hash_net xfrm_interface xfrm6_tunnel tunnel4 tunnel6 esp4 ah4 wireguard libcurve25519_generic veth xt_addrtype xt_set nf_conntrack_netlink ip_set_hash_ipportnet ip_set_hash_ipportip ip_set_bitmap_port ip_set_hash_ipport dummy ip_set ip_vs_sh ip_vs_wrr ip_vs_rr ip_vs ipmi_devintf ipmi_msghandler ip_tables vfio_pci vfio_virqfd vfio_iommu_type1 vfio dm_mirror dm_region_hash dm_log dm_mod nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 br_netfilter bridge stp llc fuse xfs libcrc32c ast drm_vram_helper qla2xxx drm_kms_helper syscopyarea crct10dif_ce sysfillrect ghash_ce sysimgblt sha2_ce fb_sys_fops cec sha256_arm64 sha1_ce drm_ttm_helper ttm nvme_fc igb sbsa_gwdt nvme_fabrics drm nvme_core i2c_algo_bit i40e scsi_transport_fc megaraid_sas aes_neon_bs [232066.596953] CPU: 6 PID: 4124696 Comm: 10.253.166.125- Kdump: loaded Not tainted 5.15.131-9.cl9_ocfs2.aarch64 #1 [232066.597356] Hardware name: Great Wall . \x93\x86...RF6260 V5/GWMSSE2GL1T, BIOS T656FBE_V3.0.18 2024-01-06 [232066.597721] pstate: 20400009 (nzCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [232066.598034] pc : nfs4_reclaim_open_state+0x220/0x800 [nfsv4] [232066.598327] lr : nfs4_reclaim_open_state+0x12c/0x800 [nfsv4] [232066.598595] sp : ffff8000f568fc70 [232066.598731] x29: ffff8000f568fc70 x28: 0000000000001000 x27: ffff21003db33000 [232066.599030] x26: ffff8000055521ae0 x25: ffff0100f98fa3f0 x24: 0000000000000001 [232066.599319] x23: ffff8000009200008 x22: ffff21003db33040 x21: ffff21003db33050 [232066.599628] x20: ffff410172fe9e40 x19: ffff410172fe9e00 x18: 0000000000000000 [232066.599914] x17: 0000000000000000 x16: 0000000000000004 x15: 0000000000000000 [232066.600195] x14: 0000000000000000 x13: ffff8000086e685a8 x12: 00000000eac0c6e6 [232066.600498] x11: 00000000000000 ---truncated---
CVE-2024-49762	Pterodactyl is a free, open-source game server management panel. When a user disables two-factor authentication via the Panel, a `DELETE` request with their current password in a query parameter will be sent. While query parameters are encrypted when using TLS, many webservers (including ones officially documented for use with Pterodactyl) will log query parameters in plain-text, storing a user's password in plain text. Prior to version 1.11.8, if a malicious user obtains access to these logs they could potentially authenticate against a user's account; assuming they are able to discover the account's email address or username separately. This problem has been patched in version 1.11.8. There are no workarounds at this time. There is not a direct vulnerability within the software as it relates to logs generated by intermediate components such as web servers or Layer 7 proxies. Updating to `v1.11.8` or adding the linked patch manually are the only ways to avoid this problem. As this vulnerability relates to historical logging of sensitive data, users who have ever disabled 2FA on a Panel (self-hosted or operated by a company) should change their passwords and consider enabling 2FA if it was left disabled. While it's unlikely that their account swill be compromised by this vulnerability, it's not impossible. Panel administrators should consider clearing any access logs that may contain sensitive data.
CVE-2024-49369	Icinga is a monitoring system which checks the availability of network resources, notifies users of outages, and generates performance data for reporting. The TLS certificate validation in all Icinga 2 versions starting from 2.4.0 was flawed, allowing an attacker to impersonate both trusted cluster nodes as well as any API users that use TLS client certificates for authentication (ApiUser objects with the client_cn attribute set). This vulnerability has been fixed in v2.14.3, v2.13.10, v2.12.11, and v2.11.12.
CVE-2024-49195	Mbed TLS 3.5.x through 3.6.x before 3.6.2 has a buffer underrun in pkwrite when writing an opaque key pair
CVE-2024-48075	A Heap buffer overflow in the server-site handshake implementation in Real Time Logic SharkSSL from 09/09/24 and earlier allows a remote attacker to trigger a Denial-of-Service via a malformed TLS Client Key Exchange message.

TLS is a huge programming project

Deploying TLS requires integration into many protocols and applications (similar with DTLS/QUIC)

TLS is a huge programming project

Deploying TLS requires integration into many protocols and applications (similar with DTLS/QUIC)

Long list of TLS-related CVEs $\implies$ this is not trivial

TLS is a huge programming project

Deploying TLS requires integration into many protocols and applications (similar with DTLS/QUIC)

Long list of TLS-related CVEs $\implies$ this is not trivial

Big holes in deployment: Many applications still don't use it at all

TLS is a huge programming project

Deploying TLS requires integration into many protocols and applications (similar with DTLS/QUIC)

Long list of TLS-related CVEs $\implies$ this is not trivial

Big holes in deployment: Many applications still don't use it at all

And... we need to make it Post-Quantum...

TLS is a huge programming project

Deploying TLS requires integration into many protocols and applications (similar with DTLS/QUIC)

Long list of TLS-related CVEs $\implies$ this is not trivial

Big holes in deployment: Many applications still don't use it at all

And... we need to make it Post-Quantum... ASAP!

Meanwhile...

This article is more than **11 years old**

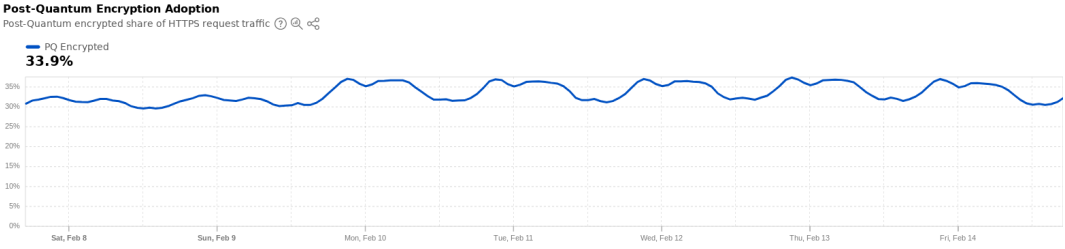
XKeyscore: NSA tool collects 'nearly everything a user does on the internet'

- XKeyscore gives 'widest-reaching' collection of online data
- NSA analysts require no prior authorization for searches
- Sweeps up emails, social media activity and browsing history
- NSA's XKeyscore program – read one of the presentations

Progress with PQ-TLS

Progress with PQ-TLS

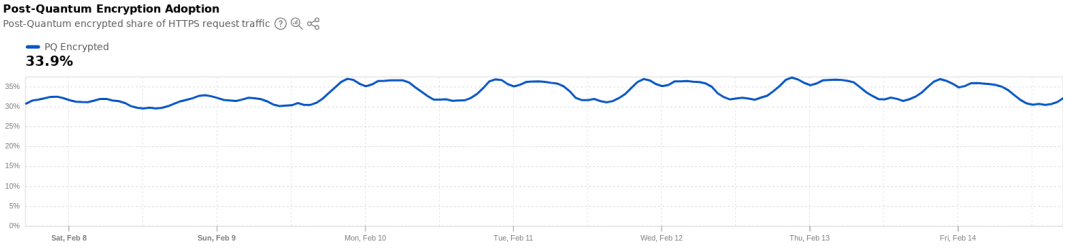
Cloudflare reports roughly 34% of its TLS connections using PQC [1].



[1] https://radar.cloudflare.com/adoption-and-usage

Progress with PQ-TLS

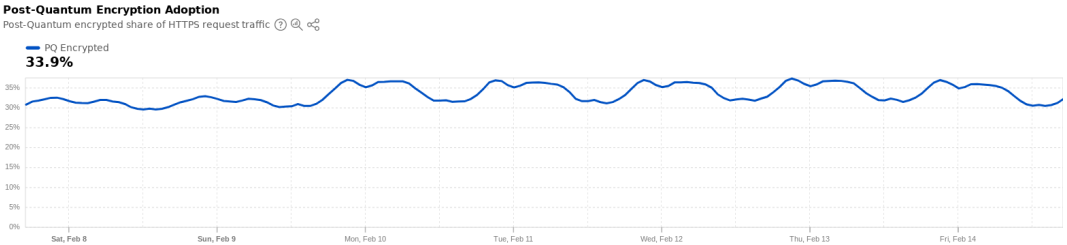
Cloudflare reports roughly 34% of its TLS connections using PQC [1].



[1] https://radar.cloudflare.com/adoption-and-usage

Progress with PQ-TLS

Cloudflare reports roughly 34% of its TLS connections using PQC [1].



But still a long way to go to universal PQ-TLS deployment

[1] https://radar.cloudflare.com/adoption-and-usage

But.. can we do more?

Without needing to upgrade every application?


WE HAVE A
TUNNEL FOR THAT

VPNs

VPNs have a big software-engineering advantage:

VPNs

VPNs have a big software-engineering advantage:

Applications are protected automatically, without modification.

VPNs

VPNs have a big software-engineering advantage:

Applications are protected automatically, without modification.

$\rightsquigarrow$ Add protection now while waiting for integration with PQ-TLS!

VPNs

Some already use post-quantum cryptography, such as:

- Mullvad,
- Rosenpass, and
- OpenSSH-based VPNs (`sntrup761` by default since 2022).

Typical VPN Usage

Typical VPN Usage

1. VPN client routes traffic through encrypted tunnel to *proxy*.

Typical VPN Usage

1. VPN client routes traffic through encrypted tunnel to *proxy*.

   Problem: Usually not end-to-end

Typical VPN Usage

1. VPN client routes traffic through encrypted tunnel to *proxy*.

   Problem: Usually not end-to-end
2. VPN client talks directly through encrypted tunnel to (multiple) *pre-configured* end-points

Typical VPN Usage

1. VPN client routes traffic through encrypted tunnel to *proxy*.

   Problem: Usually not end-to-end
2. VPN client talks directly through encrypted tunnel to (multiple) *pre-configured* end-points

   Problem: Need to configure endpoints in advance

PQ-VPN

PQ-VPN
&& end-to-end?

PQ-VPN
&& end-to-end?
&& automatic peer discovery?

PQConnect

Like a VPN, PQConnect protects network traffic for all applications.

PQConnect

Like a VPN, PQConnect protects network traffic for all applications.

Unlike a VPN, PQConnect *automatically* discovers peers and creates end-to-end post-quantum tunnels with them.

PQConnect configuration

Client: Install PQConnect.

PQConnect configuration

Client: Install PQConnect.

Server: Install PQConnect, and publish an announcement that the server name supports PQConnect.

PQConnect configuration

Client: Install PQConnect.

Server: Install PQConnect, and publish an announcement that the server name supports PQConnect.

 No peer-specific config needed 

Server Identification

How do PQConnect clients discover PQConnect servers?
(...without sending lots of extra requests?)

Server Identification

Typical DNS query:

```
www.ndss-symposium.org. IN A?
```

Server Identification

Typical DNS query:

```
www.ndss-symposium.org. IN A?
```

Typical DNS response:

```
www.ndss-symposium.org. IN A 104.18.8.22
www.ndss-symposium.org. IN A 104.18.9.22
```

Server Identification

Sometimes a bit more complicated.

DNS query:

```
www.amazon.com. IN A?
```

Server Identification

Sometimes a bit more complicated.

DNS query:

```
www.amazon.com. IN A?
```

DNS Response:

```
www.amazon.com. IN CNAME tp.47cf2c8c9-frontier.amazon.com.
tp.4[...].amazon.com. IN CNAME d3ag4hukkh62yn.cloudfront.net.
d3ag4hukkh62yn.cloudfront.net. IN A 13.33.202.88
```

Server Identification

Sometimes a bit more complicated.
DNS query:

```
www.amazon.com. IN A?
```

DNS Response:

```
www.amazon.com. IN CNAME tp.47cf2c8c9-frontier.amazon.com.
tp.4[...].amazon.com. IN CNAME d3ag4hukkh62yn.cloudfront.net.
d3ag4hukkh62yn.cloudfront.net. IN A 13.33.202.88
```

CNAMEs returned automatically.
Client follows chain to IP address, CNAMEs (usually) *ignored*.

Server Identification

But we don't *have* to ignore them

- DNS query:
  `www.pqconnect.net. in A?`

Server Identification

But we don't *have* to ignore them

- DNS query:

  `www.pqconnect.net. in A?`
- DNS Response:

  ```
  www.pqconnect.net. IN CNAME pq1htvv9k4wk[...].pqconnect.net.
  pq1htvv9k4wk[...].pqconnect.net. IN A 131.193.32.108
  ```

Server Identification

`pq1htvv9k4wkfcmpx6rufjlt1qrr4mnv[...].pqconnect.net`

"pq1" $\rightarrow$ "I support PQConnect"
"htvv9k4w[...]" $\rightarrow$ "My public key hash is htvv9k4w[...]"

PQConnect clients see announcement and establish a tunnel.
Non-PQConnect clients see IP address and connect normally.
No extra requests sent to non-PQConnect servers

Capturing Application Traffic

PQConnect filters packets to inspect incoming DNS responses.

Capturing Application Traffic

PQConnect filters packets to inspect incoming DNS responses.

PQConnect view:

```
IP 168.95.1.1.53 > 192.168.81.142.54712
www.ndss-symposium.org IN A 104.18.8.22
```

Capturing Application Traffic

PQConnect filters packets to inspect incoming DNS responses.

PQConnect view:

```
IP 168.95.1.1.53 > 192.168.81.142.54712
www.ndss-symposium.org IN A 104.18.8.22
```

Not interesting $\rightarrow$ Accept

Capturing Application Traffic

PQConnect filters packets to inspect incoming DNS responses.

PQConnect view:

```
IP 168.95.1.1.53 > 192.168.81.142.54712
www.ndss-symposium.org IN A 104.18.8.22
```

Not interesting → Accept

```
IP 168.95.1.1.53 > 192.168.81.142.59959
www.pqconnect.net. IN CNAME pq1[...].pqconnect.net.
pq1[...].pqconnect.net. IN A 131.155.69.126
```

NICE! We found a supporting server →
Rewrite 131.155.69.126 to local address that routes to PQConnect (e.g., 10.59.0.2)

Capturing Application Traffic

Application view:

Capturing Application Traffic

Application view:

```
getaddrinfo('www.ndss-symposium.org', 80) = ('104.18.8.22', 80)
```

Capturing Application Traffic

Application view:

`getaddrinfo('www.ndss-symposium.org', 80) = ('104.18.8.22', 80)`

$\rightarrow$ Send TCP handshake to 104.18.8.22

Capturing Application Traffic

Application view:

`getaddrinfo('www.ndss-symposium.org', 80) = ('104.18.8.22', 80)`

$\rightarrow$ Send TCP handshake to 104.18.8.22

`getaddrinfo('www.pqconnect.net', 80) = ('10.59.0.2', 80)`

Capturing Application Traffic

Application view:

`getaddrinfo('www.ndss-symposium.org', 80) = ('104.18.8.22', 80)`

→ Send TCP handshake to 104.18.8.22

`getaddrinfo('www.pqconnect.net', 80) = ('10.59.0.2', 80)`

→ Send TCP handshake to 10.59.0.2

Capturing Application Traffic

Application view:

`getaddrinfo('www.ndss-symposium.org', 80) = ('104.18.8.22', 80)`

→ Send TCP handshake to 104.18.8.22

`getaddrinfo('www.pqconnect.net', 80) = ('10.59.0.2', 80)`

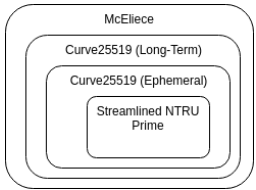
→ Send TCP handshake to 10.59.0.2

Connection now routed through PQConnect!

Hybrid Pre-/Post-Quantum KEX

PQConnect KEX uses 4 PKC schemes:

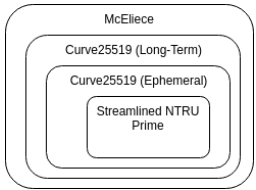
- Long-term: Classic McEliece & X25519
- Ephemeral: Streamlined NTRU Prime & X25519



Hybrid Pre-/Post-Quantum KEX

PQConnect KEX uses 4 PKC schemes:

- Long-term: Classic McEliece & X25519
- Ephemeral: Streamlined NTRU Prime & X25519

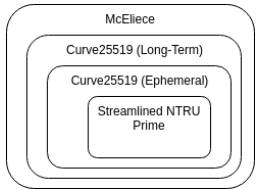


Each PKC scheme layered "inside of" the next.
Forces sequential attacks (vs. parallel)

Hybrid Pre-/Post-Quantum KEX

PQConnect KEX uses 4 PKC schemes:

- Long-term: Classic McEliece & X25519
- Ephemeral: Streamlined NTRU Prime & X25519



Each PKC scheme layered "inside of" the next.
Forces sequential attacks (vs. parallel)

Security properties of the handshake formally proven using Tamarin Prover[2].

[2] https://tamarin-prover.github.io/

Other nice things

- Long term keys are large, but efficiently cached
- Each packet symmetrically encrypted and authenticated with a unique key
- Fast Key Erasure: Symmetric keys erased upon use, or at the latest after two minutes

See paper for more details.

PQConnect resources

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

Get in touch: https://zulip.pqconnect.net

PQConnect resources

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

Get in touch: https://zulip.pqconnect.net

Questions?