CS7NS5 Security and Privacy
AS1 Security and privacy considerations

Ted Johnson
School of Computer Science and Statistics
Trinity College Dublin

25 April 2024

Introduction

In this report, I present the security and privacy considerations I have made within my dissertation. I first include a short description of my research question, followed by its contributions to the fields of security and privacy. An outline of the security and privacy aspects of relevant background work is then given. The security and privacy challenges that influence the design of my solution are provided as well as the steps taken to address them. Finally, I discuss various security and privacy implications of the design which may impact its users and service operators.

Project overview

The title of my research topic is "A distributed deployment model for Encrypted Client Hello". Encrypted Client Hello (ECH) is a proposed extension to the Transport Layer Security protocol version 1.3 (TLS 1.3) which has begun to see implementation and adoption on the Internet [1–3]. ECH seeks to allow encryption of the ClientHello message, which can contain potentially sensitive information such as the Service Name Indication (SNI) and Application-Layer Protocol Negotiation (ALPN) extensions. This is partially achieved through serving many private domains behind a common ECH-service provider to form an anonymity set that conceals the true domain requested by the client.

Nottingham has previously cautioned against the introduction of centralisation through Internet standards [4]. Of particular relevance to ECH is his highlight of the adverse effect centralisation can have on infrastructure resilience and service availability through reliance on a single entity. This is especially detrimental to ECH where the effectiveness of its anonymity set grows with the number of private domains served by a single ECH-service provider. Nottingham also writes on susceptibility of centralisation to stifle "permissionless" innovation and induce an unhealthy monoculture which may result in less overall technological progress and robustness of the ECH protocol.

Additionally, allowing entirely independent servers to co-operate from across the Internet to provide ECH support for each other enables several distinct organisations and entities to work together to offer improved user privacy for all without the requirement for co-located servers nor the dependence of any on the availability on another. Consider here global networks of whistleblower services, investigative journalists and human rights non-profit organisations who share an interest in protecting the confidentiality of their members and users from persecution and retaliation.

For these reasons, my research work has been on identifying and evaluating practical models for deployment of ECH amongst several co-operating TLS servers, where each server operates both as the origin server of its own domains as well as an ECH provider for other participating servers. The model must address a number of implementation challenges, predominately related to ensuring the security of the protocol is not compromised and minimising the performance impact to the connection while strengthening service availability.

Background security and privacy aspects

Transport Layer Security (TLS) is a cryptographic protocol proposed by the Internet Engineering Task Force (IETF) which enables secure communication over public networks. Applications and services can establish an encrypted communication channel to transmit private information such that confidentiality, integrity and authenticity of the data can be ensured. In order to prevent eavesdropping, tampering and message forgery, TLS provides several security features based on a number of cryptographic mechanisms: All exchanged data is both encrypted to ensure confidentially and guaranteed to be unmodified by an intermediate party, as well as allowing for the authentication of the recipient and sender identities.

TLS 1.3 is the latest defined standard for the protocol, having been published in August 2018 and contributing to the deprecation of TLS 1.0 and TLS 1.1 in March 2021 [5, 6]. The version introduces many major changes over TLS 1.2, including the addition of a zero round trip time resumption (0-RTT) mode, further encryption and optimisation of the handshake and removal of outdated cryptographic algorithms and security mechanism with all key exchanges now providing forward secrecy. Furthermore, the usage of extensions to allow for additional functionality to be included within the handshake has been significantly expanded in TLS 1.3.

The Domain Name System (DNS) was designed by Mockapetris in 1984 as a naming system that associates hierarchical alphanumeric identifiers, referred to as domain names, with various resource records, like IP addresses [7, 8]. Notably, Mockapetris makes no mention of security nor privacy in the original DNS specification and such concerns have only begun to be addressed in recent years, as summarised by Bortzmeyer in 2015 [9]. This has largely been due the naming system information being perceived as public knowledge and not requiring security mechanisms. As such, DNS query and response communication have historically been sent unencrypted using the User Datagram Protocol (UDP). It has not been until the last decade with the revelations of widespread global surveillance that issues such as these have started to see much more attention.

As consequence of this, both DNS over TLS (DoT) and DNS over HTTPS (DoH) were conceived as methods for performing privacy-preserving DNS queries [10, 11]. These protocols add confidentiality and data integrity to DNS by encapsulating queries and responses inside secure TLS channels. The most notable difference between the standards is the port number used, as DoT traffic goes to the non-standard port 853 while DoH is served through the standard HTTPS port 443. This difference has led to some adoption problems with DoT when compared to DoH, as it is not unusual for network firewalls to prohibit traffic to non-standard ports. This also has the effect of making DoT usage being quite conspicuous, while DoH disguises itself amongst other HTTPS traffic. García et el. list these as factors when measuring a wider adoption of DoH in 2021 [12].

The operation of the ECH extension is supported by these adaptions within the TLS and DNS ecosystems. Its functionality is based on clients using the public key of an ECH-service provider to send an encrypted TLS 1.3 ClientHello message, which the provider decrypts and uses to proxy the TLS 1.3 connection to the true origin server. The provider must first generate an ECH encryption key pair and some associated metadata. This public key and metadata, referred to as an ECH configuration or ECHConfig, may then be shared out-of-band with ECH-enabled clients though a secure context like DoH. A client may then use this public key and metadata to construct a ClientHello message, named the ClientHelloOuter, holding unremarkable values for the provider alongside the ECH extension containing an encrypted ClientHello, named the ClientHelloInner, itself holding the real values for a private domain. To establish a TLS connection to the origin server of this domain, the client initiates a TLS connection using the ClientHelloOuter with the provider, which decrypts the ClientHelloInner and relays the connection to the origin server, which itself completes the TLS handshake with the client through the provider. Importantly, the provider is incapable of eavesdropping on this secure channel, as the TLS connection is authenticated and end-to-end encrypted between the client and origin server.

ECH uses the Hybrid Public Key Encryption (HPKE) specification for performing public key encryption [13]. HPKE defines a standard scheme for combining the benefits of asymmetric and symmetric cryptographic algorithms, such that the performance of symmetric cryptography can be gained where only the public key of the receiver is known. This is achieved through using the public key of the receiver to generate a symmetric encryption key as well as an encapsulated shared secret. This encapsulated shared secret can be sent to the receiver, which can generate the symmetric encryption key using its private key. Any ciphertext produced by the sender with the symmetric encryption key can now be decrypted by the receiver.

Design considerations

When researching and designing a solution for the distributed deployment of ECH, ensuring the security of the protocol could not be compromised was a major criteria which directly influenced several architectural decisions. The primary threat models considered were based on surveillance of network traffic for the purpose of inferring the true origin server, but it was also important to acknowledge a number of practical concerns regarding deployment.

For instance, one of the solution's purposed strategy for enabling distributed deployment of ECH is to take advantage of the commonly seen round-robin DNS technique. Round-robin DNS works by responding to DNS queries with multiple valid IP addresses in a random order from which the client selects one. By installing the IP addresses of all co-operating TLS servers into all private domain name DNS resource records, clients who resolve any of these domains will be directed any one of the servers. This has the additional benefit of allowing clients to immediately retry a connection with the next IP if the connection fails, although some clients might not implement this. However, a major flaw with this approach is requiring all servers to share an ECH private key because there is no way to specify which ECHConfig value should be used by the client when it selects an IP address. To avoid widely shared secrets, an alternative is suggested where a dynamic DNS service regularly substitute private domain name DNS resource records such that load is fairly balanced across servers. This allows us to specify specific IP address and ECHConfig value pairs for each domain name, alleviating the need for sharing private keys.

During the relay of an ECH-enabled connection to the true origin server, the ClientHelloInner is sent as a regular ClientHello without any security, entirely defeating the purpose of ECH. To solve this issue, server-server communication must have some form on encryption to maintain the confidentiality of the ClientHelloInner. Furthermore, as both the client-server and server-server channels are over public networks, this reveals a perfect attack surface for network observers to perform a correlation attack through analysis of server ingress and egress traffic. Even after an origin server completes the TLS handshake and establishes end-to-end encrypted communication with a client, these eavesdroppers can deduce which origin server the client is interacting with by finding patterns between ingress to egress times, packet counts and traffic behaviour [14]. To address these vulnerabilities, all server-server communication must be encrypted as well as masked using traffic obfuscation techniques such as traffic pacing and dummy packet mixing [15].

Further discussion

Outside of considerations made during the design stage, it is also important to acknowledge the privacy and security implications to users and service operators in addition to the assumptions made and likely risks associated with the implementation of the solution within the real world. For example, presently it is trivial to identify clients that are attempting to use ECH as the inclusion of the extension is easily recognisable in any TLS 1.3 ClientHello message. It is not improbable that such messages may be blocked or subjugated to special treatment by network infrastructure. Similarly, both suggested DNS resource record publication schemes provide easily accessible lists of all IP addresses participating in the co-operative ECH deployment anonymity set. It is conceivable that such information could be sensitive in certain contexts, as this could be proof of collaboration between two entities which would rather appear separate to others. Additionally, these members must possess some form of mutual trust regarding the concealment of their users because each ECH-service provider implicitly knows of the true origin server accessed by any user connection for which it has relayed. Lastly, it would be understandable that service operators may be hesitant to adopt the presented distributed deployment model for ECH as it creates a larger attack surface for malicious actors to try to exploit. An argument can be made that the increase in complexity within participating TLS servers increases the likelihood for misconfiguration or oversight which reveals a vulnerability of the system.

Summary

This report has illuminated a number of security and privacy considerations relevant to the background and design of my research project on distributed deployment of ECH. We have seen how the security properties of ECH are coupled with those of TLS 1.3 and DoH to deliver protection of potentially sensitive client information such as the SNI and the ALPN list. An overview of the security and privacy challenges associated with the project which motivated the design of the solution was provided, namely not compromising service availability, avoiding widely shared secrets, and disrupting traffic correlation attacks. Finally, a discussion on the impacts to the security and privacy of the users and service operators of the solution was given, including the assumption of amity among co-operating TLS servers, acknowledgment of an increased attack surface for co-operating TLS servers, and recognition that usage of ECH is currently easily detectable.

References

[1] Eric Rescorla et al. *TLS Encrypted Client Hello*. Internet-Draft draft-ietf-tls-esni-18. Work in Progress. Internet Engineering Task Force, Mar. 2024. 51 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/18/.

[2] Zisis Tsiatsikas, Georgios Karopoulos, and Georgios Kambourakis. “Measuring the adoption of TLS encrypted client hello extension and its forebear in the wild”. In: *European Symposium on Research in Computer Security*. Springer. 2022, pp. 177–190. DOI: 10.1007/978-3-031-25460-4_10.

[3] Christopher Wood Achiel van der Mandele Alessandro Ghedini and Rushil Mehra. *Encrypted Client Hello - the last puzzle piece to privacy*. Sept. 2023. URL: https://blog.cloudflare.com/announcing-encrypted-client-hello (visited on 03/24/2024).

[4] Mark Nottingham. *Centralization, Decentralization, and Internet Standards*. RFC 9518. Dec. 2023. DOI: 10.17487/RFC9518. URL: https://www.rfc-editor.org/info/rfc9518.

[5] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/info/rfc8446.

[6] Kathleen Moriarty and Stephen Farrell. *Deprecating TLS 1.0 and TLS 1.1*. RFC 8996. Mar. 2021. DOI: 10.17487/RFC8996. URL: https://www.rfc-editor.org/info/rfc8996.

[7] Paul Mockapetris. *Domain names - concepts and facilities*. RFC 1034. Nov. 1987. DOI: 10.17487/RFC1034. URL: https://www.rfc-editor.org/info/rfc1034.

[8] Paul Mockapetris. *Domain names - implementation and specification*. RFC 1035. Nov. 1987. DOI: 10.17487/RFC1035. URL: https://www.rfc-editor.org/info/rfc1035.

[9] Stéphane Bortzmeyer. *DNS Privacy Considerations*. RFC 7626. Aug. 2015. DOI: 10.17487/RFC7626. URL: https://www.rfc-editor.org/info/rfc7626.

[10] Zi Hu et al. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. May 2016. DOI: 10.17487/RFC7858. URL: https://www.rfc-editor.org/info/rfc7858.

[11] Paul E. Hoffman and Patrick McManus. *DNS Queries over HTTPS (DoH)*. RFC 8484. Oct. 2018. DOI: 10.17487/RFC8484. URL: https://www.rfc-editor.org/info/rfc8484.

[12] Sebastián García et al. “Large scale measurement on the adoption of encrypted DNS”. In: *arXiv e-prints* (July 2021). DOI: 10.48550/arXiv.2107.04436.

[13] Richard Barnes et al. *Hybrid Public Key Encryption*. RFC 9180. Feb. 2022. DOI: 10.17487/RFC9180. URL: https://www.rfc-editor.org/info/rfc9180.

[14] Adam Back, Ulf Möller, and Anton Stiglic. “Traffic analysis attacks and trade-offs in anonymity providing systems”. In: *International Workshop on Information Hiding*. Springer. 2001, pp. 245–257. DOI: 10.1007/3-540-45496-9_18.

[15] Xinwen Fu et al. “Analytical and empirical analysis of countermeasures to traffic analysis attacks”. In: *2003 International Conference on Parallel Processing, 2003. Proceedings.* IEEE. 2003, pp. 483–492. DOI: 10.1109/ICPP.2003.1240613.