

Opportunistic Security: Some Protection Most of the Time
draft-dukhovni-opportunistic-security-03

Abstract

This memo introduces the "Opportunistic Security" (OS) protocol design pattern. Protocol designs based on OS depart from the established practice of employing cryptographic protection against both passive and active attacks, or no protection at all. As a result, with OS at least some cryptographic protection should be provided most of the time. For example, the majority of Internet SMTP traffic is now opportunistically encrypted. OS designs remove barriers to the widespread use of encryption on the Internet. The actual protection provided by opportunistic security depends on the advertised security capabilities of the communicating peers.

This document promotes designs in which cryptographic protection against both passive and active attacks can be rolled out incrementally as new systems are deployed, without creating barriers to communication.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 16, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1.	Introduction	2
2.	Terminology	4
3.	The Opportunistic Security Design Pattern	5
4.	Opportunistic Security Design Principles	7
5.	Example: Opportunistic TLS in SMTP	8
6.	Security Considerations	9
7.	Acknowledgements	9
8.	References	9
8.1.	Normative References	9
8.2.	Informative References	10
	Author's Address	10

[1.](#) Introduction

Historically, Internet security protocols have emphasized comprehensive "all or nothing" cryptographic protection against both passive and active attacks. With each peer, such a protocol achieves either full protection or else total failure to communicate (hard fail). As a result, operators often disable these security protocols at the first sign of trouble, degrading all communications to cleartext transmission. Protection against active attacks requires authentication. The ability to authenticate any potential peer on the Internet requires a key management approach that works for all.

Designing and deploying a key management for the whole Internet is for now an unsolved problem. For example, the Public Key Infrastructure (PKI) used by the web (often called the "Web PKI") is based on broadly trusted public certification authorities (CAs). The Web PKI has too many trusted authorities and imposes burdens that not all peers are willing to bear. Web PKI authentication is vulnerable to MiTM attacks when the peer reference identifier ([\[RFC6125\]](#)) is obtained indirectly over an insecure channel, perhaps via an MX or SRV lookup. With so many certification authorities, which not everyone is willing to trust, the communicating parties don't always agree on a mutually trusted CA. Without a mutually trusted CA, authentication fails, leading to communications failure. The above issues are compounded by operational difficulties. For example, a common problem is for site operators to forget to perform timely renewal of expiring certificates. In interactive applications,

security warnings are all too frequent, and end-users learn to actively ignore security problems.

DNS Security (DNSSEC [[RFC4033](#)]) can be used to leverage the global DNS infrastructure as a distributed authentication system. DNS-Based Authentication of Named Entities (DANE [[RFC6698](#)]) provides the guidelines and DNS records to use the DNS as an alternative to the Web PKI. However, at this time, DNSSEC is not sufficiently widely adopted to allow DANE to play the role of an Internet-wide any-to-any authentication system. Therefore, protocols that mandate authentication for all peers cannot generally do so via DANE. Opportunistic security protocols on the other hand, can begin to use DANE immediately, without waiting for universal adoption.

Other authentication mechanisms have been designed that do not rely on trusted third parties. The trust-on-first-use (TOFU) authentication approach makes a leap of faith (LoF, [[RFC4949](#)]) by assuming that unauthenticated public keys obtained on first contact will likely be good enough to secure future communication. TOFU is employed in SSH and in various certificate pinning designs. TOFU does not protect against an attacker who can hijack the first contact communication and requires more care from the end-user when systems update their cryptographic keys. TOFU can make it difficult to distinguish routine system administration from a malicious attack.

The lack of a global key management system means that for many protocols only a minority of communications sessions can be authenticated. When protocols only offer the choice between an authenticated encrypted channel or no protection, the result is that most traffic is sent in cleartext. The fact that most traffic is not encrypted makes pervasive monitoring easier by making it cost-effective, or at least not cost-prohibitive; see [[RFC7258](#)] for more detail.

To reach broad adoption, it must be possible to roll out support for unauthenticated encryption or authentication incrementally. Incremental rollout on the scale of the Internet means that for some considerable time security capabilities vary from peer to peer, and therefore protection against passive and active attacks needs to be applied selectively peer by peer.

We will use the phrase "opportunistically employed" to mean that the use of a type of protection or a security mechanism was tailored to the advertised capabilities of the peer. Both opportunistically employed encryption and opportunistically employed authentication need to avoid deployment roadblocks and need to be designed with care to "just work".

To enable broader use of encryption, it must be possible to opportunistically employ encryption with peers that support it without always demanding authentication. This defends against pervasive monitoring and other passive attacks, as even unauthenticated encryption (see definition in [Section 2](#)) is preferable to cleartext.

The opportunistic security design pattern involves stepping up from a baseline security policy compatible with all relevant peers to the most secure policy compatible with the capabilities of a given peer. Note, this is rather different from setting a high standard and attempting to determine (perhaps by asking the user) whether an exception can be made.

The risk of active attacks should not be ignored. The opportunistic security design pattern recommends that protocols employ multiple cryptographic protection levels, with encrypted transmission accessible to most if not all peers. Protocol designers are encouraged to produce protocols that can securely determine which peers support authentication, and can then establish authenticated communication channels resistant to active attacks with those peers.

Operators must be able specify explicit security policies that override opportunistic security where appropriate.

[2](#). Terminology

Perfect Forward Secrecy (PFS): As defined in [\[RFC4949\]](#).

Man-in-the-Middle (MiTM) attack: As defined in [\[RFC4949\]](#).

Trust on First Use (TOFU): In a protocol, TOFU typically consists of accepting and storing an asserted identity, without authenticating that assertion. Subsequent communication using the cached key/credential is secure against an MiTM attack, if such an attack did not succeed during the (vulnerable) initial communication. The SSH protocol makes use of TOFU. The phrase "leap of faith" (LoF, [\[RFC4949\]](#)) is sometimes used as a synonym.

Authenticated encryption: Encryption using a session establishment method in which at least the initiator (client) authenticates the identity of the acceptor (server). This is required to protect against both passive and active attacks. Authenticated encryption can be one-sided, where only the client authenticates the server. One-sided authentication of the server by the client is the most common deployment used with Transport Layer Security (TLS, [\[RFC5246\]](#)) for "secure browsing" in which client authentication is typically performed at another layer in the protocol.

Authenticated encryption can also be two-sided or "mutual". Mutual authentication plays a role in mitigating active attacks when the client and server roles change in the course of a single session. Authenticated encryption is not synonymous with use of AEAD [[RFC5116](#)]. AEAD algorithms can be used with either authenticated or unauthenticated peers.

Unauthenticated Encryption: Encryption using a session establishment method that does not authenticate the identities of the peers. In typical usage, this means that the initiator (client) has not verified the identity of the target (server), making MiTM attacks possible. Unauthenticated encryption is not synonymous with non-use of AEAD [[RFC5116](#)].

3. The Opportunistic Security Design Pattern

Opportunistic Security is a protocol design pattern that aims to remove barriers to the widespread use of encryption on the Internet. A related goal is broader adoption of protection against active attacks, by enabling incremental deployment of authenticated encryption.

The opportunistic security design pattern supports multiple protection levels, while seeking the best protection possible.

The determination of which security mechanisms to use can vary from case to case and depends on the properties of the protocol in which OS is applied. In many cases, OS will result in negotiating channels with one of the following security properties:

- o No encryption (cleartext), which provides no protection against passive or active attacks.
- o Unauthenticated encryption (definition in [Section 2](#)), which protects only against passive attacks.
- o Authenticated encryption, (definition in [Section 2](#)) which protects against both passive and active attacks.

An opportunistic security protocol first determines the capabilities of a peer. This might include whether that peer supports authenticated encryption, unauthenticated encryption or perhaps only cleartext. After each peer has determined the capabilities of the other, the most preferred common security capabilities are activated. Peer capabilities can be advertised out-of-band or (negotiated) in-band.

An OS protocol may elect to apply more stringent security settings for authenticated encryption than for unauthenticated encryption. For example, the set of enabled TLS cipher-suites might be more restrictive when authentication is required.

Security services that "just work" are more likely to be deployed and enabled by default. It is vital that the capabilities advertised for an OS-compatible peer match the deployed reality. Otherwise, OS systems will detect such a broken deployment as an active attack and communication may fail.

This might mean that peer capabilities are further filtered to consider only those capabilities that are sufficiently operationally reliable, and any that work only "some of the time" are treated by an opportunistic security protocol as "not present" or "undefined".

Opportunistic security does not start with an over-estimate of peer capabilities only to settle for lesser protection when a peer fails to deliver. Rather, opportunistic security sets a minimum protection level expected of all peers, which is raised for peers that are capable of more.

Opportunistic security protocols should provide a means to enforce authentication for those peers for which authentication can be expected to succeed based on information advertised by the peer via DANE, TOFU or other means. With DANE, the advertisement that a peer supports authentication is downgrade-resistant. What is "opportunistic" here is the selective use of authentication for certain peers; much in the same way as unauthenticated encryption may be used "opportunistically" with peers capable of more than cleartext.

Enforcement of authentication is not incompatible with opportunistic security. If an OS-enabled peer (A) makes available authenticated key material, e.g., via DANE, to peer (B), then B should make use of this material to authenticate A, if B is OS-enabled and supports DANE.

With a peer that does not advertise authentication support, to which transmission even in cleartext is permissible, authentication (or the lack thereof) must never downgrade encrypted communication to cleartext. When employing opportunistic unauthenticated encryption, any enabled by default authentication checks need to be disabled or configured to soft-fail, allowing the unauthenticated encrypted session to proceed.

Cleartext support is for backwards compatibility with already deployed systems. Even when cleartext needs to be supported,

protocol designs based on opportunistic security prefer to encrypt, employing cleartext only with peers that do not appear to be encryption capable.

4. Opportunistic Security Design Principles

Coexist with explicit policy: Explicit security policy preempts opportunistic security. Administrators or users can elect to disable opportunistic security for some or all peers and set a fixed security policy not based on capabilities advertised or published by the peer. Alternatively, opportunistic security might be enabled only for specified peers, rather than by default. Opportunistic security never displaces or preempts explicit policy. Some applications or data may be too sensitive to employ opportunistic security, and more traditional security designs can be more appropriate in such cases.

Emphasize enabling communication: The primary goal of opportunistic security is to enable secure communication and maximize deployment. If potential peers are only capable of cleartext, then it may be acceptable to employ cleartext when encryption is not possible. If authentication is only possible for some peers, then it is likely best to require authentication for only those peers and not the rest. Opportunistic security needs to be deployable incrementally, with each peer configured independently by its administrator or user. Opportunistic security must not get in the way of two peers communicating when neither advertises or negotiates security services that are not in fact available or that don't function correctly.

Maximize security peer by peer: Opportunistic security strives to maximize security based on the capabilities of the peers. Communications traffic should generally be at least encrypted. Opportunistic security protocols may refuse to communicate with peers for which higher security is expected, but for some reason is not achieved. Advertised capabilities must match reality to ensure that the only condition under which there is a failure of communication is when one or both peers are under an active attack.

Employ PFS: Opportunistic Security should employ Perfect Forward Secrecy (PFS) to protect previously recorded encrypted communication from decryption even after a compromise of long-term keys.

No misrepresentation of security: Unauthenticated encrypted communication must not be misrepresented to users or in application logs of non-interactive applications as equivalent to communication over an authenticated encrypted channel.

5. Example: Opportunistic TLS in SMTP

Many Message Transfer Agents (MTAs, [[RFC5598](#)]) support the STARTTLS ([RFC3207](#)) ESMTP extension. MTAs acting as SMTP clients are generally willing to send email without TLS (and therefore without encryption), but will employ TLS (and therefore encryption) when the SMTP server announces STARTTLS support. Since the initial ESMTP negotiation is not cryptographically protected, the STARTTLS advertisement is vulnerable to MiTM downgrade attacks. Further, MTAs do not generally require peer authentication. Thus opportunistic TLS for SMTP only protects against passive attacks.

Therefore, MTAs that implement opportunistic TLS either employ unauthenticated encryption or deliver over a cleartext channel. Recent reports from a number of large providers suggest that the majority of SMTP email transmission on the Internet is now encrypted, and the trend is toward increasing adoption.

Not only is the STARTTLS advertisement vulnerable to active attacks, but also at present some MTAs that advertise STARTTLS exhibit various interoperability problems in their implementations. As a result, it is common practice to fall back to cleartext transmission not only when STARTTLS is not offered, but also when the TLS handshake fails, or even when TLS fails during message transmission. This is a reasonable trade-off, since STARTTLS protects only against passive attacks, and absent an active attack TLS failures are simply interoperability problems.

Some MTAs employing STARTTLS ([RFC3207](#)) abandon the TLS handshake when the server MTA fails authentication, only to immediately deliver the same message over a cleartext connection. Other MTAs have been observed to tolerate unverified self-signed certificates, but not expired certificates, again falling back to cleartext. These and similar implementation errors must be avoided. In either case, had these MTAs instead used the OS design pattern, the communication would still have been encrypted and therefore protected against passive attacks.

Protection against active attacks for SMTP is proposed in [\[I-D.ietf-dane-smtp-with-dane\]](#). That draft introduces the terms "Opportunistic TLS" and "Opportunistic DANE TLS", which are for now informal.

6. Security Considerations

Though opportunistic security potentially supports transmission in cleartext, unauthenticated encryption, or other cryptographic protection levels short of the strongest potentially applicable, the effective security for peers is not reduced. If a cryptographic capability is neither required by policy nor supported by the peer, nothing is lost by going without. Opportunistic security is strictly stronger than the alternative of providing no security services when maximal security is not applicable.

Opportunistic security coexists with and is preempted by any applicable non-opportunistic security policy. However, such non-opportunistic policy can be counter-productive when it demands more than many peers can in fact deliver. Non-opportunistic policy should be used with care, lest users find it too restrictive and act to disable security entirely.

7. Acknowledgements

I would like to thank Steve Kent. Some of the text in this document is based on his earlier draft. I would like to thank Dave Crocker, Peter Duchovni, Paul Hoffman, Steve Kent, Scott Kitterman, Martin Thomson, Nico Williams, Paul Wouters and Stephen Farrell for their helpful suggestions and support.

8. References

8.1. Normative References

- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", [RFC 3207](#), February 2002.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), March 2005.
- [RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", [RFC 5116](#), January 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", [RFC 6125](#), March 2011.

- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", [RFC 6698](#), August 2012.

8.2. Informative References

- [I-D.ietf-dane-smtp-with-dane]
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", [draft-ietf-dane-smtp-with-dane-11](#) (work in progress), August 2014.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", [RFC 4949](#), August 2007.
- [RFC5598] Crocker, D., "Internet Mail Architecture", [RFC 5598](#), July 2009.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", [BCP 188](#), [RFC 7258](#), May 2014.

Author's Address

Viktor Dukhovni
Two Sigma

Email: ietf-dane@dukhovni.org