Trust Issues with Opportunistic Encryption

Scott Rose, Stephen Nightingale, Doug Montgomery
{scottr, night, dougm@nist.gov}
National Institute of Standards and Technology (NIST)
January 15, 2014

Abstract

Recent revelations have shed light on the scale of eavesdropping on Internet traffic; violating the privacy of almost every Internet user. In response, protocol designers, engineers and service operators have begun deploying encryption (often opportunistic) to protect the confidentiality of users' communications. The lack of authentication in opportunistic encryption could have the perverse affect of putting more end users at risk: thinking that they are "secure", an end user may divulge private information to an imposter instead of the service they believe they have contacted. When adding protection mechanisms to protocols, designers and implementers should not downplay the importance of authentication in order to make opportunistic encryption easier to deploy.

Introduction

During the 88th Internet Engineering Task Force (IETF) meeting in November 2013, there was a technical plenary on how the IETF, as a group, should address pervasive monitoring of Internet traffic. It was decided that the IETF would pursue privacy-enhancing extensions to existing protocols and urge the implementation and use of encryption for all Internet traffic [1]. This has lead to a push to include the use of opportunistic encryption to provide for the confidentiality of traffic in protocols that traditionally either did not have the option, or the option was not widely used (e.g. Simple Mail Transfer Protocol [SMTP]).

Opportunistic encryption usually refers to having the option to encrypt the communication between two parties without requiring or providing authentication of either party. In other words, having confidentiality (stopping eavesdroppers) but not having authentication. For example, having two Mail Transfer Agents (MTA's) exchange email messages via SMTP over a Transport Layer Security (TLS) connection. End users also commonly perform setting up an opportunistic encrypted channel when using a web browser and seeing a HTTPS certificate warning. When users click the "Ok" (or similar) button to proceed with viewing the webpage, they are agreeing to use a connection that is not authenticated, but encrypted.

In many scenarios, this level of security is acceptable and should be encouraged to limit third party eavesdropping. However, in other communications, end users would like to have some level of authentication that the party they believe they are communicating with is, in fact, the party they are actually communicating with. Most people would not be concerned with authentication when viewing a webpage with

movie times, sports scores, etc. where little of their private information is shared, but when checking their credit scores, bank balance or medical records, most end users would like some assurance that they are communicating with their bank, doctor, etc. and not an imposter. Authentication is seen as an important component to most security protocols, but with the current pressure to combat pervasive monitoring on the Internet there is a concern that authentication will be deferred in favor of opportunistic encryption in a rushed deployment. Some current specification work going in in the IETF seems to encourage opportunistic encryption while downplaying the desire for authentication for some communication.

Example of Risks Involved with Opportunistic Encryption without Authentication

In many ways, an attacker taking advantage of opportunistic encryption without authentication is similar to an attacker subverting a Certificate Authority (CA) to issue a certificate for a given domain to the attacker. This type of attack has been documented several times in the past [2][3] and was successful to an extent in capturing users' private information.

In the above case, an attacker subverted the certificate validation mechanism in the protocol to impersonate a legitimate site. When relying on opportunistic encryption, the attacker does not even need to risk exposure in obtaining a false credential. The attacker can simply generate their own certificates and insert themselves in the path of the communication. To reduce exposure afterwards, they could act as a Man-in-the-Middle and forward the victim's messages to the legitimate site. When this attack succeeds, the victim is still vulnerable to eavesdropping, as well as more active attacks.

If the protocol specification calls for the use of opportunistic encryption and there are no end user options to require (or signal) authentication then the end users can develop a false sense of security. End users who do not understand the differences between confidentiality and authentication may believe they are protected when they are not, and expose private information to a potential attacker.

Dangers of Ignoring Authentication in Protocol Specifications

There is a risk when simply specifying a means to add opportunistic encryption to a protocol that an implementer may interpret that authentication is not needed or that authentication will never be desired. As a result, there is a risk that code to perform authentication (e.g. certificate validation) will not be added to implementations, as it would never be needed.

An example is the work-in-progress in the IETF to specify DNS-Based Authentication of Named Entities (DANE) support for SMTP servers that use TLS [4]. There is text in the draft that calls for service providers to signal to clients to not do full validation on the certificates presented by the contacted server. In practical operations, this makes sense. Often, SMTP servers do not have a collection

of root certs to use in validation and often there is no real-time recovery from errors until the problem is resolved by administrator action (i.e. installing new root certs).

However, in some communities there are local security policies in place that require all certificates to pass (or at least be subject to) full validation before continuing communication. Often, these policies were in place to handle other protocols (such as signed email or user access), but were made global for the entire organization or community. In order to remain in compliance with local policies, administrators will request that validation be done for their certificates. Some implementations may not correctly handle these requests. Additionally, administrators may not be able to purchase or configure implementations to perform validation when requested as implementers interpret specification SHOULD/SHOULD NOT keywords as the most likely scenario and do not include code to perform validation (as it would likely never be used according to the specification).

Use of Trust Infrastructures

Providing authentication requires more work (on all parties involved in communication) than opportunistic encryption, but the wheel does not have to be reinvented for every protocol. There are tools and resources available to protocol designers to add authentication support to protocols.

The most ubiquitous is the Domain Name System (DNS), which can be used to store certificate information using the new Resource Record Types (RRTypes) defined by the IETF DANE working group [5]. A client that understands how to query for DANE RRTypes can validate that the certificate presented during a TLS handshake matches what the authoritative domain holder claims, but could also confirm the CA used to obtain the certificate (if a CA issued it), or the local trust anchor used for the domain.

There are other options available for those that may not wish to rely on the existing CA/PKI for certificates. Examples such as the Certificate Transparency [6] work and Sovereign keys [7] use publicly available logs to build trust rather than just who issued the certificate. The concept is that since the certificate (and holder) are listed on several publicly visible third party services; attackers using spoofed certificates would be detectable by clients.

All of these solutions come with their own drawbacks, as well as all requiring more work (and time) being spent by the client in performing authentication. This would likely affect user experience and response time. New user education may also be needed to help users understand client configuration options that may be available as well as authentication results in protocols.

Conclusions

Encouraging the use of confidentiality in Internet communication will benefit the end user. Standards bodies like the IETF is correct in identifying that passive monitoring (pervasive or targeted) is an attack that protocols should protect

against. Opportunistic encryption can and should be used as a last resort to provide a minimal level of confidentiality to protect end users' privacy. Ideally, protocols should be specified to include the option to allow communicating parties to authenticate each other, as opportunistic encryption may provide a false sense of security in security ignorant end users.

Since opportunistic encryption entails no authentication, end users may believe their privacy is protected when in fact they are sending data to an imposter. Protocol designers should consider authentication as important as confidentiality since the designers cannot always determine when an end users would desire authentication. At the very least, protocol designers should not discourage authentication in order to make opportunistic encryption easier to implement.

References

[1] "We Will Strengthen the Internet" IETF blog Nov 2013 http://www.ietf.org/blog/2013/11/we-will-strengthen-the-internet/

[2] Microsoft Security Advisory 2607712 Fraudulent Digital Certificates Could Allow Spoofing. Microsoft Corp. Aug 2011. http://technet.microsoft.com/en-us/security/advisory/2607712

[3] " Microsoft, Yahoo, Google, Skype, Mozilla Sites Hit by Fraudulent Certificates". eWeek. March 2011. http://www.eweek.com/c/a/Security/Microsoft-Yahoo-Google-Skype-Mozilla-Sites-Hit-by-Fraudulent-Certificates-619996/

[4] V. Dukhovni, W. H. Hardakar. "SMTP security via opportunistic DANE TLS" Work in Progress. http://datatracker.ietf.org/doc/draft-ietf-dane-smtp-with-dane/

[5] DANE Working Group Charter http://datatracker.ietf.org/wg/dane/

[6] Certificate Transparency project homepage http://www.certificate-transparency.org/

[7] Sovereign Keys project homepage https://www.eff.org/sovereign-keys