The security pendulum

Julien Maisonneuve, Vijay K. Gurbani, Thomas Fossati
Alcatel-Lucent

Abstract

The lessons in the post-Snowden world have understandably alarmed anyone concerned with privacy and exposed weaknesses that may be used by a number of actors with intent to cause havoc. The instant reaction to the Snowden findings have, however, pushed the situation from a "no protection" default to a "maximum protection", breaking a number of network features that are provided by middle-boxes for the purposes of engineering, operations and maintenance of the network. We believe that there should be a balance between allowing privacy and retaining some of the useful features of middle-boxes. To study this tradeoff and advance the understanding of the constraint space and enable consensus over what is achievable we survey existing and emerging techniques to balance privacy and support of middle-boxes.

1 Introduction

The Snowden revelations on the practices of the intelligence community have prompted an understandable backlash from the Internet community. It is now clear this has changed the way we looked at security in the Web. From a model where the default was no encryption, we went to widespread recommendations to encrypt everything whatever the cost. These costs come in several types: some are related to the cost of encryption, but the most subtle come in the form of new constraints and practices that used to work but no longer do. The first tide of encryption came with systematic moves from HTTP to HTTPS by the major application providers and by key protection actors. Official statements from key Internet bodies [2], [3], have endorsed this transition. HTTPS prevents examination and modification of content and leaves only tracks of connection establishment on the network path. The second wave, which still has limited impact, is the introduction of HTTP/2 and the default encryption model of several major implementations. HTTP/2 introduces a twist to the issue of encryption because encrypted proxying effectively creates an opaque tunnel that prevents any examination of the web traffic. That option, even if it is not deployed systematically, is the ultimate in web traffic hiding. In the space of a few months, we have gone to a situation with relatively limited traffic hiding to nearly systematic traffic hiding. A number of services, which depended on some traffic visibility, have stopped working; more will follow as encryption

reaches further. The approach we'd like to advocate in this paper is that privacy is not a binary option: even if you consider that privacy is paramount, there are a lot of "technical" aspects in web communication which have limited privacy consequences for the end users but can be used gainfully by intermediaries to improve the quality of service, either at an individual or collective level. This paper is divided into four parts: Section 2 provides examples of the consequences of encryption, Section 3 examines trust issues, Section 4 offers a survey of existing approaches. Section 5 is a case study that considers the security and privacy implications of mapping the "mobile throughput guidance" framework [4] into the SPUD prototype [6].

2 Consequences of ubiquitous encryption

Other documents [13] provide an overview of the consequences of ubiquitous encryption on a large panel of services and practices. It is not our intent to duplicate this work and we will simply insist on examples that are typical of everyday usage. The main consideration is that encryption (e.g. with HTTPS) removes any opportunity to examine or modify Web traffic. This removes opportunities to add many web-related services (such as parental control or malware filtering) or optimizations (such as image and video resizing/trans-rating, and most forms of caching). Web Caching relies on the ability to compare URLs and store content on an intermediary node for later consumption. HTTPS breaks caching because caches cannot read clear-text content, and encrypted text is different for different clients. Without significant changes in the Web caching practices, encryption will make caching less and less likely. Caching was hitherto largely transparent to both consumers and producers; it is unlikely that this will continue to be the case. The breaking of this "default behavior" will certainly mean that caching will be less prevalent in the future web as actors have to make conscious decisions (which may entail costs and responsibilities) for something that was automatic before. The other aspect to consider is the increased complexity and/or lower effectiveness of traffic engineering, congestion management and diagnostic functions. One for all, the ability to partition traffic based on its delivery requirements (i.e. real-time versus batch, reliable versus unreliable), and decide how to treat it depending on the current congestion state on the RAN, is a vital function of the mobile access network. Traffic classification is currently based on DPI techniques, which are made ineffective by widespread traffic encryption. We argue that, in order to re-establish these essential network functions, the currently assumed threat and trust model [14] must evolve accordingly.

3 Tooling for trust

The basic assumption behind the ubiquitous encryption model appears to be that nothing can be trusted within the network and that there is no legitimate activity regarding content that could be performed at that level. This is a convenient assumption because

it catches all threats, but also because doing otherwise requires exposing additional complexity and establishing trust relationships that are difficult to build. Let's suppose we are ready to introduce a more sophisticated trust model. The current situation is that endpoints don't trust middle-boxes because there is no way to distinguish a bad middle-box from a good one. Indeed, if such is the case, opting not to trust anybody is a very sensible decision. Trust doesn't spring ex nihilo: we need tools that enable endpoints to trust the middle-boxes – i.e. allowing them to become aware of who operates (and is therefore responsible for) a given middle-box, and what kind of service does the middle-box implement. And the dual is just as important: why should a middle-box blindly trust the endpoints? We argue that in the vast majority of cases it shouldn't; therefore, our tools must support symmetrical trust establishment. But that's not enough: since we are potentially considering three party conversations involving users, content providers and the network at the same time, then a framework that aims at fully solving the "middle-box collaboration" problem should be based on a technology that can comfortably handle three-party trust establishment. Another important aspect is that of network service negotiation. The driving principle should be that the user has always the last word and, if given the right information, he/she is able to take the best decision.

4 A survey of existing or previously proposed tools

There are various lines of inquiry, both in standards forums like the IETF as well as in academic and industrial laboratories. Essentially, these strands straddle the spectrum from focusing on solutions that impact the application layer to ones that include the transport layer as well. We provide a brief survey of existing work in this area. The salient point present in all of the literature reviewed below is transparency, i.e., the user is aware of and may be able to veto the use of intermediaries being allowed into a communication channel. Druta et al. [15] present a rationale for why end-to-end use of TLS precludes network intermediaries that provide instrumental services (caching, media optimization). They further derive a broad set of characteristics of a possible solution, a primary one being the notion of "fine-grained" control of selected communication data objects by the user that get elevated to end-to-end security such that intermediaries are unable to obtain clear-text access to these objects that are flowing through them. The works by Fossati et al. [10] and Naylor et al. [12] provide avenues to implement such fine-grained control. Both these works build upon the widely deployed TLS protocol as used in HTTP. Zhou et al. [11] use "fine-grained" control as well, but their notion of control is in the form of a special web server serve mixed-content in a page such that content marked private is served over a protected connection (using the `https` scheme) and content marked public is sent over a clear-text channel (using the `http` scheme). The private connection also serves as the secure channel over which validation tags for the public content are handed over to the user agent, in a fashion similar to SRI [16].

5 Case study: mobile throughput guidance

At any point in time the mobile networks knows how much bandwidth is available between the RAN and the user's mobile device. The value, which is a function of the radio link quality reported by the mobile device, can be fed back into TCP to inform the server's choice about optimal cwnd sizing[1].

The Mobile Throughput Guidance (MTG) requirements and architecture document [4] introduces a framework that operates on these basis. The MTG in-band signalling protocol [5] proposes an implementation of said framework which uses a new TCP option (available in clear text or auth-encrypted fashion) to hand over the information from the RAN to the server. The document [5] also illustrates the results of an experiment involving video delivery to a mobile device, which shows substantial gain for both the network and the application compared to vanilla TCP.

5.1 Translating MTG into SPUD

This kind of signalling from network to the endpoint (and viceversa) could be realised adding a couple of new declarations to the SPUD prototype [6] phrasebook:

- an *application* declaration, sent by the application server to the network to request MTG information with a given frequency, e.g. in milliseconds;
- a *path* declaration, sent by a MTG capable network to the requesting server (at the requested rate) as an unsigned value representing the throughput available in the RAN, e.g. in Mbits/s.

5.2 Security and privacy (or why we need an armoured SPUD)

At present, SPUD lacks a mechanism to secure its messages; therefore, MTG could be implemented only in clear-text. The lack of message protection causes at least two security issues:

- any on-path box can inject rogue information to alter traffic dynamics, e.g. throttling the bandwidth available to one or more users;
- any on-path box posing as an application server can exfiltrate MTG information from the network.

The latter amplifies the privacy problem related to the localisation of the user relative to the serving tower, which could be guessed from the quality of the radio link implicitly

[1] The question as to whether this mechanism allows fair share of the link capacity between multiple different flows, as well as its impact on overall network stability have been debated quite extensively during the IETF 92 meeting. However, those potential issues are not relevant in this context.

encoded in the MTG value. In order to fix that, SPUD needs mutual authentication between the mobile network and the application server, and authenticated encryption at least for the messages flowing from the network to the application.

The point of this case study is to show that whenever the exchanged declarations have relevant security and/or privacy implications, the messaging framework must support some form of communication security. Establishment of an n-party secure group to support exchanging authenticated-encrypted declarations is the core problem to be solved. Ignoring that will reduce the number of use cases that can be successfully addressed by the framework to such a low number to make it irrelevant. On the other hand, trying to work around the core problem will create insecure solutions. Neither of these outcomes is desirable.

6 Conclusion

We have reached a point in which the Web security pendulum has swayed from one side (no encryption) and is reaching the vicinity of the other (full encryption), with adverse consequences that may not be fully understood. There are a few approaches that are currently being considered to limit the adverse consequences of ubiquitous encryption on in-network services. It is indeed possible to strike a balance between complete privacy and some level of traffic interaction by giving more control to the end parties (servers and consumers). However several of these approaches are likely to be more complex to use and deploy than the previous alternatives (no and full encryption), which preserved the simple interaction model of HTTP. Additional work is necessary to fully understand the consequences of ubiquitous encryption, assess the legitimacy of in-network services and chart a path to make them possible in the new encrypted landscape.

References

[1] S. Farrell, H. Tschofenig, "Pervasive Monitoring Is an Attack," *IETF BCP 188*, May 2014.

[2] IETF Internet Architecture Board, "IAB Statement on Internet Confidentiality," Nov 2014

[3] W3C TAG, "Securing the Web," Jan 2015

[4] A. Jain et al., "Requirements and reference architecture for Mobile Throughput Guidance Exposure," *IETF work in progress*, February 2015.

[5] A. Jain et al., "Mobile Throughput Guidance Inband Signalling Protocol," *IETF, work in progress*, March 2015.

[6] J. Hildebrand, B. Trammell, "Session Protocol for User Datagrams (SPUD) Prototype," *IETF work in progress*, March 2015.

[7] W. George et al., "Application Enabled Collaborative Networking Use Cases," *IETF work in progress*, July 2014.

[8] P. Fan et al., "Application Enabled Collaborative Networking: Problem Statement," *IETF work in progress*, July 2014.

[9] P. Fan et al., "Application Enabled Collaborative Networking Requirements," *IETF work in progress*, July 2014.

[10] T. Fossati, V. K. Gurbani and V. Kolesnikov, "Love all, trust few: On trusting intermediaries in HTTP" *ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ACM, 2015.

[11] Z. Zhou, T. Benson, "Towards a Safe Playground for HTTPS and MiddleBoxes with QoS2" *ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ACM, 2015.

[12] D. Naylor et al., "multi-context TLS (mcTLS): Enabling Secure In-Network Functionality in TLS", *ACM SIGCOMM 2015*, ACM, 2015

[13] Kathleen Moriarty, Al Morton et al., "Effect of Ubiquitous Encryption", IETF Internet-Draft, Work in Progress, Mar 2015

[14] E. Rescorla, "Meet the new threat model, same as the old threat model", Strengthening the Internet Against Pervasive Monitoring (STRINT) Workshop, 2014

[15] D. Druta, T. Fossati, M. Ihlar, G. Klas, D. Lopez and J. Reschke, "A rationale for fine-grained intermediary aware end-to-end protocols", IETF Internet-Draft, Work in Progress, Oct 2014.

[16] D. Akhawe, F. Marier, F. Braun, J. Weinberger, "Subresource Integrity" W3C Working Draft, July 2015