

PERPASS
Internet-Draft
Intended status: Informational
Expires: May 08, 2014

H. Tschofenig

November 04, 2013

Tackling Pervasive Surveillance or How to improve Security of the
Internet?

draft-tschofenig-perpass-surveillance-01.txt

Abstract

Surveillance is the observation or monitoring of an individual's communications or activities. Surveillance is one of several privacy /security threats engineers try to take into account in their designs. The reports about pervasive monitoring of Internet traffic have, however, surprised many since the scale was not envisaged during the design of many Internet protocols even though the ambition to offer end-to-end security on the Internet dates back even to the 70ies. The approach to get access to meta-data as well as to communication content has taken forms that are largely indistinguishable from ordinary attacks.

This document explains the attacks in context of the larger Internet eco-system.

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 May 08, 2014.

Internet-Draft

Tackling Pervasive Surveillance

November 2013

Copyright Notice

Copyright (c) 2013 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Attack Surface	3
2.1.	Cryptographic Primitives	3
2.2.	Protocols and Architecture	4
2.3.	Implementations	5
2.4.	Deployment	6
3.	Security Considerations	6
4.	IANA Considerations	7
5.	Acknowledgments	7
6.	Normative References	7
	Author's Address	9

[1.](#) Introduction

Securing the Internet is a rather complicated task since the threat landscape has changed significantly over the last 20 years. For many of the recognized security weaknesses solutions have been developed and standardized. Unfortunately, the existence of specifications by itself is not enough: security protocols need to be implemented and deployed. Since many of the tougher security challenges suffer from a collective action problem it typically takes many years until widespread deployment has been reached (typically requiring sufficient energy and enough pain). The recently observed pervasive monitoring activities represent a new challenge to the Internet community and require us to review and revisit some earlier design decisions.

To fully understand the role of the IETF in this context it is useful to look at the types of attacks that are occurring. It quickly becomes clear that the development of many countermeasures is entirely within scope of the IETF. But the deployment and use is

outside out scope, and requires interactions with the larger Internet eco-system.

2. Attack Surface

The attack surface is categorized into four areas, as shown in Figure 1. In subsequent sections more details are provided and examples are listed.

Aging or Broken	Weak	Implementation	Insecure
Cryptographic	Protocol or	Bugs and other	Deployment
Primitives	Architectural	Vulnerabilities	Practices
	Foundation		

Figure 1: Attack Surface.

2.1. Cryptographic Primitives

Internet security relies on sound cryptographic primitives, such as hash functions, random number generators, integrity and encryption algorithms, etc. The basic design philosophy is that the strength of keyed algorithms relies on the length of the secret/private key. It is well-known that these cryptographic primitives "age" as processing power of computing hardware increases. This means that over time it is faster to search through the key space with the same amount of financial budget spent. (Note: How much of the key space an adversary has to search depends on a number of factors. Due to the birthday paradoxon it has to search on average 1/2 of the key space. It can actually be much lower when lower entropy secrets are used, such as passwords.) Researchers have also made improvements in analyzing the building blocks of these algorithms and new attack

techniques (such as side channel attacks). This has lead to a continued development of new cryptographic primitives.

The IETF has played a minor role in the work on cryptographic primitives. Instead, it has been a consumer of these building blocks and has therefore relied on others to select specifications and to provide guidance. As an exception one could see the publication of HMAC [20]. In fact, the crypto-community world-wide is rather small and for a variety of reasons the National Institute of Standards and Technology (NIST) has spearheaded many of these developments. The IETF security community has relied on NIST to provide guidance largely because no other groups have come forward to offer advice.

While there have been problems with weaknesses in cryptographic primitives (e.g., RC4 [1], [2], [3]) those have not been a substantial issue from a standardization point of view thanks to 'crypto-agility'. (Note that RC4 is not a NIST standard.) Crypto-agility is the ability of a protocol to adapt to evolving cryptographic algorithms and security requirements. This includes provisions that allow security protocols to adopt different cryptographic algorithms without substantial disruption to deployed implementations. This does not mean the implementation is unchanged and, of course, the need for backwards compatibility creates downgrade attacks.

Rumors about backdoors in specific elliptic curves, and random number generators have created some uncertainty about what algorithms are 'safe' to use. The crypto-community is still debating about the validity of these claims and investigating about how long weaknesses in standards should have been known to experts.

[2.2.](#) Protocols and Architecture

Internet protocols and communication architectures belong to the core expertise of the IETF. While security experts have been around in the early days of the IETF the security community grew over time after security considerations sections became a mandatory part of IETF specifications [21]. The overall understanding of security is still increasing thanks to education efforts, reviews from the security area directorate, and the push back from the IESG when questionable documents arrive.

Still, there are a number of challenges. For example, cryptographic attacks like BEAST [5], CRIME [6], and Lucky Thirteen [4] targeted the Transport Layer Security (TLS) protocol when specific algorithms are used with specific application layer protocols (such as HTTP). More difficult to deal with are security and privacy challenges with entire protocols architectures, as the design of email, instant messaging, voice over IP (VoIP), DNS, DHCP, etc. demonstrate. Often, insecure versions of a protocol are standardized and completed first before the secure version is developed. For example, consider security for HTTP, SIP, XMPP, eMail, etc. While this may not have a consequence on paper it certainly impacts follow-up implementations and deployments. Section 8 of [19] provides an interesting summary of the design tradeoffs that had been made in the real-time communication architecture as used by VoIP and instant messaging. The difficulty is often not in crafting a security solution at the level of a single specification, but rather ensuring that the protocol development of an entire communication architecture provides good security and privacy properties after many years of standardization when various different industry trends (such as cloud

computing, and the JavaScript-based Web), and the interests of participating parties re-shape the original design goals. Furthermore, often the attention is paid on protecting the payload of the content only and meta-data is exposed to service providers and other parties, particularly with server-centric communication architectures.

In many cases, the implications of certain design decisions are subtle. Two examples are:

Cookies: For example, the excitement of Web companies to use HTTP cookies [23], [22] as a replacement for cryptographic authentication was hard to anticipate.

VoIP Media Security: The large number of key exchange mechanisms standardized for VoIP (see [16], [17], [15]) might have provided ways to fulfill needs of different deployment scenarios but certainly confused the industry and didn't increase interoperability. New VoIP security authentication protocols are still proposed today.

Another example for an architectural weakness can be found with the

public key infrastructure (PKI) when the limitations of the PKI became apparent in 2011: DigiNotar, a Dutch certification authority, had a security breach and in the same year a Comodo affiliate was compromised. Both cases lead to fraudulent issue of certificates allowing man-in-the-middle attacks on TLS secured data Web interactions. There have been claims that the same architectural vulnerability has been exploited by the National Security Agency (NSA) in man-in-the-middle attacks [[14](#)].

Improving security and privacy for different communication protocols has been subject of discussion on the IETF perpass list [[9](#)]. Note that some discussions go beyond suggesting actions for the IETF; they belong to the discussion in [Section 2.3](#) and [Section 2.4](#). As another example of ongoing work is a document on best current practices for Transport Layer Security [[18](#)], which gathers experience from recent security attacks and recommends state-of-the-art ciphersuites.

[2.3](#). Implementations

Once standardization work is completed the specifications have to be implemented. Often those who develop the specifications are not necessary the same parties who implement the software. The specifications therefore have to offer enough context and be readable to avoid security problems via misinterpretation. Also, those who implement and those who deploy are also not necessarily the same set of people. For example, some developers write open source libraries

useful for a wide range of communities, as it is the case with OpenSSL or GnuTLS. Note: This description is rather simplified version of the typical IETF protocol development. In many cases, the development process is not linear since protocols are implemented while they are specified and implementation results are fed back into the standardization effort. Still, for many successful protocols implementations the number of those involved in implementations far exceeds the number of standardization participants.

Implementations may show a number of security weaknesses, such as lack of security features, quality of the implementations (e.g., implementations with insufficient penetration testing), weak pseudo-random number generators [[7](#)], [[10](#)], etc. Since the source code of many implementations is not available to the public, backdoors may be built-in, as it was rumored with [[8](#)].

Many implementations of Web applications, however, suffer from basic vulnerabilities (such as injection or cross-site scripting attacks), as the top-10 charts of the Open Web Application Security Project (OWASP) reveal [12]. Sometimes vendors make design decisions for their product implementations that lead to security vulnerabilities, for example when devices are shipped with default-passwords or with enabled debugging interfaces [13].

[2.4.](#) Deployment

Finally, implementations of various protocols are put together and complete systems are deployed. Those who deploy have to make decisions that go beyond pure protocol aspects; for example, they have to consider various configuration options. These deployment decisions have an important impact on the provided privacy and security properties. Examples include, backend server protocols secured only with "physical security" (i.e., without cryptographic security protection), email services without TLS protection, custom security designs (see, for example, WhatsApp [11]), etc. Depending on the jurisdiction within which a service is provided, those who deploy systems may assume certain for data retention, and support for lawful intercept.

[3.](#) Security Considerations

This entire document focuses on security.

[4.](#) IANA Considerations

This document does not require actions by IANA.

[5.](#) Acknowledgments

I would like to thank the IAB for encouraging me to turn my IAB-internal presentation into a document. I would also like to thank

Stephen Kent, Rene Struik, and Linus Nordberg for their detailed reviews.

6. Normative References

- [1] Fluhrer, S., Mantin, I., and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", Selected Areas in Cryptography , 2001.
- [2] ISOBE, T., OHIGASHI, T., WATANABE, Y., and M. MORII, "Full Plaintext Recovery Attack on Broadcast RC4", International Workshop on Fast Software Encryption , 2013.
- [3] AlFardan, N., Bernstein, D., Paterson, K., Poettering, B., and J. Schuldt, "On the Security of RC4 in TLS", Usenix Security Symposium 2013, 2013, <<https://www.usenix.org/conference/usenixsecurity13/security-rc4-tls>>.
- [4] AlFardan, N. and K. Paterson, "Lucky Thirteen: Breaking the TLS and DTLS Record Protocols", IEEE Symposium on Security and Privacy , 2013.
- [5] Rizzo, J. and T. Duong, "Browser Exploit Against SSL/TLS", 2011, <<https://packetstormsecurity.com/files/105499/Browser-Exploit-Against-SSL-TLS.html>>.
- [6] Rizzo, J. and T. Duong, "The CRIME Attack", EKOparty Security Conference 2012, 2012.
- [7] Ars Technica, "Stop using NSA-influenced code in our products, RSA tells customers", URL: <http://arstechnica.com/security/2013/09/stop-using-nsa-influence-code-in-our-product-rsa-tells-customers/>, Sep 2013.
- [8] Boing Boing, "Anti-Tor malware reported back to the NSA", URL: <http://boingboing.net/2013/08/05/anti-tor-malware-reported-back.html>, Aug 2013.
- [9] IETF, "PERPASS Mailing List", URL: <https://www.ietf.org/mail-archive/web/perpass/current/maillist.html>, Oct 2013.

- [10] Nadia Heninger, "New research: There's no need to panic

- over factorable keys-just mind your Ps and Qs", URL: <https://freedom-to-tinker.com/blog/nadiah/new-research-theres-no-need-panic-over-factorable-keys-just-mind-your-ps-and-qs/>, Oct 2013.
- [11] fileperms Blog, "WhatsApp is broken, really broken", URL: <http://fileperms.org/whatsapp-is-broken-really-broken/>, Sep 2012.
- [12] OWASP, "Open Web Application Security Project (OWASP): Top Ten Project", URL: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project, Oct 2013.
- [13] Wired, "NSA Laughs at PCs, Prefers Hacking Routers and Switches", URL: <http://www.wired.com/threatlevel/2013/09/nsa-router-hacking/>, Apr 2013.
- [14] Zeljka Zorz, "NSA impersonated Google in MitM attacks", URL: <https://www.net-security.org/secworld.php?id=15579>, Apr 2013.
- [15] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", [draft-ietf-avtcore-rtp-security-options-08](#) (work in progress), October 2013.
- [16] Wing, D., Fries, S., Tschofenig, H., and F. Audet, "Requirements and Analysis of Media Security Management Protocols", [RFC 5479](#), April 2009.
- [17] Perkins, C. and M. Westerlund, "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution", [draft-ietf-avt-srtp-not-mandatory-14](#) (work in progress), October 2013.
- [18] Sheffer, Y. and R. Holz, "Recommendations for Secure Use of TLS and DTLS", [draft-sheffer-tls-bcp-01](#) (work in progress), September 2013.
- [19] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", [RFC 6973](#), July 2013.
- [20] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.

- [21] Postel, J., "Instructions to RFC Authors", [RFC 1543](#), October 1993.
- [22] Williams, N., "Hypertext Transport Protocol (HTTP) Session Continuation: Problem Statement", [draft-ietf-websec-session-continue-prob-00](#) (work in progress), July 2013.
- [23] Tschofenig, H., Turner, S., and M. Hanson, "An Inquiry into the Nature and the Causes of Web Insecurity", [draft-tschofenig-secure-the-web-04](#) (work in progress), October 2012.

Author's Address

Hannes Tschofenig

Email: Hannes.Tschofenig@gmx.net

