Australian Government Parliamentary Joint Committee on Intelligence and Security (PJCIS)

Letter Regarding the TELECOMMUNICATIONS AND OTHER LEGISLATION AMENDMENT (ASSISTANCE AND ACCESS) BILL 2018

14 November 2018

We appreciate the opportunity to provide comments concerning the Telecommunications Assistance and Access Bill 2018. We are engineers, technologists, and researchers who work to make devices, networks, and the Internet itself more secure. Our comments here focus on one particularly concerning aspect of the Bill: that secrecy provisions in the Bill could thwart efforts to implement new technical transparency systems necessary to the ongoing security of the Internet. We are concerned that the secrecy provisions would undermine user trust, threaten the ongoing security of the Internet, and could actually convert government demands into demands that would create systemic weaknesses.

The new secrecy requirements in the Bill come at a particularly dangerous time given a growing category of cybersecurity threats to the integrity of the software, firmware, and hardware on which global Internet users depend. The computer security community is responding to these threats by designing a new family of technical transparency mechanisms that will help users detect and thwart attacks on the systems we use today. But, the Bill as written could preclude a designated communications provider (DCP) from implementing the latest technical security transparency mechanisms for fear of violating rules regarding public disclosure of new surveillance capabilities mandated by a technical capability notice (TCN), as described in Section 317ZF. In this letter we describe the new cybersecurity risks emerging, explain how the technical community is working to address them, and show why the secrecy provisions of the Bill would impair those efforts at great cost to security for individuals, institutions, and governments.

Today's digital environment has evolved substantially from the time when the main threat was that an adversary might break into a confidential communication or compromise encrypted data by defeating the confidentially scheme itself. A new class of vulnerabilities has emerged in which attackers seek to undermine the authentication of security keys, application software, operating systems, and even hardware. A range of recent and damaging attacks illustrate this new class of risks and demonstrate the importance of new transparency mechanisms that can help assure users that they are using trustworthy systems. For example, a certificate authority (CA) managed by the Chinese Government was found to be issuing false certificates for Google.com and many other websites[1]; a popular Linux distribution has been found to be hacked

[1] https://security.googleblog.com/2015/03/maintaining-digital-certificate-security.html

and backdoored[2], putting millions of users at risk; and a provider of a VPN commonly used by the U.S. Government was found to be backdoored[3]. These vulnerabilities put all classes of users at risk, everyone from individuals to large corporations, critical infrastructure providers to the national security apparatus of governments.

In order to protect against these emerging security risks, the industry has developed "transparency" technologies. In a transparency system, the actions of a centralized provider that is at risk of being compromised are required to be logged to a public, cryptographically verifiable ledger. This allows the centralized providers' actions to be scrutinized by the broader community of Internet users –- in particular, by those who would be harmed by a compromised central authority — so that bad actions by the centralized provider can be identified and remediated. Here, we summarize three types of transparency systems that have been deployed or are under active development for all of the above types of providers.

1. Certificate Transparency is in use by web browsers today to detect the issuance of rogue website certificates. Valid certificates are published on an online log — when a browser sees a new certificate that doesn't appear on a trusted log, it warns the user that it may be fraudulent, even though the cryptographic signatures appear valid.

   Let's say that under the proposed law the government issued a TCN that required a CA to enable surreptitious government access to information protected by HTTPS. The CA could issue a misleading certificate to trick the user into believing that its communication was protected end-to-end, despite the fact that the government could actually read the plain text. No such proposal has been made but the broad powers proposed in the Bill could open the door to this possibility.

   The anti-transparency penalties in the Bill could create a situation in which certain implementations of this vital security protocol are compromised without users knowing it was happening. Either browsers would have to be forced via a TCN to ignore the warning that a certificate had not been publicly logged, which would make the log useless, or the browser would function as intended — and disallow the connection. This is the *exact* attack scenario that Certificate Transparency was meant to avoid. If implemented in even a narrow class of HTTPS services, this would leave users unsure about whether they could trust any TLS guarantees at all.

[2] *https://blog.linuxmint.com/?p=2994*
[3] *https://blog.cryptographyengineering.com/2015/12/22/on-juniper-backdoor/*

2. Message Key Transparency: In a similar environment to the HTTPS example above, one may consider a group chat where law enforcement has asked to be secretly added as a hidden party. This may work for some chat apps, but the addition of a cryptographic key transparency log (for example, coniks[4] or Google Key Transparency[5]), would allow any user to publicly verify the members' keys, even from another computer. In the exact same way as described for Certificate Transparency, key transparency solutions would either be made meaningless by a TCN regime, or make the eavesdropping TCN key trivially discoverable.

3. Binary Transparency: We know that there have been many publicly documented cases in which software has been maliciously misrepresented as trustworthy as a result of stolen access to code-signing keys. Binary Transparency is a method of proving that a software update, or other blob of code, including code embedded in firmware, has been seen and provisionally trusted by all users. Such transparency helps provide users with confidence that a "special" malicious piece of code has not been developed particularly for some user and targeted at them to steal data or exploit some other aspect of their system. Imagine if someone broke into your operating system vendor (e.g. Google) and managed to steal the keys responsible for authenticating software updates (e.g. for Android). That person could create a malicious update for that system, and present it to any phone he or she desired. A similar scenario might arise if a malicious actor stole access to keys that enabled firmware updates for widely used computers, whether desktop devices, rack-based CPUs for cloud services, or mobile devices.

   The growing threats to the integrity of software and firmware, whether from criminal syndicates or malicious state actors, will place increasing importance on binary and firmware transparency tools. So, we should be particularly wary of legal requirements that put such transparency systems at risk. Binary Transparency requires any update to be logged by a transparency server. In other words, the fact that your software vendor has signed a malicious update would easily be discovered by those monitoring the log. In the case of law enforcement access demands, if a TCN were to ask Google to provide a mandated "special" update, for example, it would be easily found and disclosed by such a transparency mechanism.

   Binary transparency becomes even more important when we consider the institutions that control firmware (the software that makes your hardware function). Operating system vendors often rely on numerous third parties for hardware-specific features, maintained using highly privileged firmware updates. A Huawei Android phone, for instance, relies on Google to develop the operating

[4] https://coniks.cs.princeton.edu/
[5] https://security.googleblog.com/2017/01/security-through-transparency.html

> system, and Google relies on Huawei, Qualcomm, and others to develop firmware for the hardware itself. If it were mandated that such updates were publicly logged, everyone could ensure that no such firmware were crafted with malicious intent.

Our concerns reflect an important shift in the technical debate about strategies to address the needs for law enforcement exceptional access. Going back several decades, discussions about enabling law enforcement access to encrypted communications on the Internet were focused on breaking the *confidentiality* guarantees provided by Internet protocols and applications (for example, via key escrow systems such as the Clipper Chip and more recent proposals such as CLEAR[6] and others[7]). Recently, there has been an emerging consensus among technical researchers and policymakers in debates –- industry, governments, and academics –- that compromising confidentiality is not a workable approach. We understand that the drafters of the Bill intended to reflect this shift insofar as the Bill makes clear that DCPs may not be required to implement or build a “systemic weakness” or vulnerability. The Bill’s proponents have publicly stated that DCPs could therefore not be requested or required to implement key escrow as part of a technical assistance notice (TAN) or TCN[8].

In response to this emerging caution about violating confidentiality controls, proponents of giving law enforcement access to encrypted communications have shifted their focus to approaches that break *authenticity* security guarantees instead. Under this type of approach providers could be requested or required to make a law enforcement authority appear as though it were an authorized participant. Technical details on these proposals are still not available, but allowing law enforcement to break the chain of trust in the authentication of a chat room, a piece of software, or a website risks that the transparency mechanisms now being designed to protect users from malicious attacks would have to be compromised such that users can no longer trust the systems they use.

New technical transparency technologies are emerging as critical tools in defending applications and the Internet at large from increasingly common attacks on the integrity of the software, firmware, and security infrastructure. These technical transparency techniques appear to be directly threatened by the proposed legal requirements that service providers hide the existence of surveillance capacity from the public. If DCPs are required to build or implement new law enforcement access capabilities without revealing the existence of those capabilities, the providers will be unable to use transparency technologies, thus undermining the trust of all users. We encourage the

[6] *https://github.com/rayozzie/clear/blob/master/clear-rozzie.pdf*
[7] *http://cseweb.ucsd.edu/~savage/papers/lawful.pdf*
[8] *https://minister.homeaffairs.gov.au/peterdutton/Pages/Crime-Stoppers-National-Conference,-Canberra.aspx*

legislators now debating the Bill to take these technical security concerns into account as the consideration of the legislation continues.

Signed (affiliations for identification purposes only)

Hal Abelson
Professor, Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Ross J. Anderson
Professor of Security Engineering, Computer Laboratory
University of Cambridge

Richard Barnes
Chief Security Architect for Collaboration
Cisco

Xavier Boyen
Associate Professor, Science and Engineering Faculty
Queensland University of Technology

Alissa Cooper
Fellow, Cisco

Chris Culnane
Lecturer, Computing and Information Systems
University of Melbourne

Rajeev Gore
Professor, Research School of Computer Science
The Australian National University

Ben Laurie
Project Lead, Certificate Transparency Project
Founder, OpenSSL Project
Founding Director, Apache Software Foundation

Peter G. Neumann
Chief Scientist, SRI International Computer Science Lab

Mark Nottingham
Member, Internet Architecture Board

Josef Pieprzyk
Professor & Senior Principal Research Scientist
CSIRO | Data61

Ron Rivest
Institute Professor
Massachusetts Institute of Technology

Bruce Schneier
Fellow, Berkman Klein Center for Internet and Society
Harvard University

Jeffrey Schiller
IETF Security Area Director 1993-2004

Michael Specter
EECS PhD Student and Security Researcher
Massachusetts Institute of Technology

Vanessa Teague
Chair, Cybersecurity and Democracy Network
University of Melbourne

Yuval Yarom
Senior Lecturer, School of Computer Science
University of Adelaide | Data61

Daniel J. Weitzner
Founding Director, Internet Policy Research Initiative
Massachusetts Institute of Technology