Short Paper: TLS Ecosystems in Networked Devices vs. Web Servers*

Nayanamana Samarasinghe and Mohammad Mannan

Concordia Institute for Information Systems Engineering
Concordia University, Montreal, Canada
{n_samara,mmannan}@ciise.concordia.ca

Abstract. Recently, high-speed IPv4 scanners, such as ZMap, have enabled rapid and timely collection of TLS certificates and other security-sensitive parameters. Such large datasets led to the development of the Censys search interface, facilitating comprehensive analysis of TLS deployments in the wild. Several recent studies analyzed TLS certificates as deployed in web servers. Beyond public web servers, TLS is deployed in many other Internet-connected devices, at home and enterprise environments, and at network backbones. In this paper, we report the results of a preliminary analysis using Censys on TLS deployments in such devices (e.g., routers, modems, NAS, printers, SCADA, and IoT devices in general). We compare certificates and TLS connection parameters from a security perspective, as found in common devices with Alexa 1M sites. Our results highlight significant weaknesses, and may serve as a catalyst to improve TLS security for these devices.

1 Introduction

Beyond user-level computing devices and back-end servers, there are many other Internet-connected devices that serve important roles in everyday IT operations. Such devices include routers, modems, printers, cameras, SCADA (supervisory control and data acquisition) controllers, DVR (digital video recorders), HVAC (heating, ventilating and air conditioning technology), CPS (cyber physical systems), and NAS (network-attached storage) devices. Several past studies have identified critical security issues in these devices, including authentication bypass, hard-coded passwords and keys, misconfiguration, serious flaws in their firmware and web interfaces; example studies include: [26], [10], [9], [7], [8], [24]. The recent massive DDoS attack on DynDNS as attributed to the Mirai botnet (e.g., [25]), populated by DVRs, IP cameras and other IoT devices, shows the clear danger of security flaws and weaknesses in these devices.

Over the years, manufacturers of networked devices have implemented some security mechanisms, notably, the adoption of SSL/TLS for communication with

* An extended version of this paper is available as a technical report [27], which additionally includes: analysis of certificate issuers, certificate reuse, DH prime number reuse, stronger cipher suites, and device type ranking.

other devices. With the help of the ZMap [16] high-speed IPv4 scanner, some recent projects analyzed the TLS ecosystem for web, email and SSH servers, and identified and measured significant security issues in TLS deployments in the wild; see e.g., [15], [21], [14], [1].

Heninger et al. [20] highlighted faulty random number generators in networked devices (see also the recent follow-up work [19]). Chung et al. [6] analyzed over 80 million invalid TLS certificates, and attribute most of them to network devices, including modems/home routers, VPNs, NAS, firewalls, IP cameras and IPTVs. However, we are unaware of any comprehensive study on the overall TLS ecosystem for networked devices. In this paper, we report our results on analyzing certificates and TLS parameters from 299,858 devices (out of 1,018,911), collected from the Censys (`censys.io`) service on October 8, 2016. Unsurprisingly, many devices still use crypto primitives that are currently being phased out from modern browsers and web servers.

Specifically, we found a significant number of devices use unsafe RSA 512-bit keys (4100 certificates) and 768-bit keys (8919 certificates). The vulnerable/deprecated RC4 stream cipher is still widely used in devices (113,186, 37.7%). A large number of devices (66,540, 22.2%; 19,063) also use (deprecated) SSLv3 and SSLv2, respectively. We also compare TLS security parameters between devices and Alexa Top 1M sites, which clearly highlight the differences in these two domains. In all security aspects that we consider (SSL/TLS version, signature, encryption and hashing algorithms, and RSA key length), all device types are significantly more vulnerable than Alexa 1M sites (see [27]). Our analysis focuses on TLS security weaknesses, but we also summarize the use of stronger security primitives in devices and Alexa 1M sites. We hope our results, albeit preliminary, to serve as a catalyst to quick fixing of TLS issues in devices, so that these devices do not remain less secure than the HTTPS/web ecosystem in the long run.

2 Related work

We briefly discuss measurement studies on real-world TLS deployments.

To allow researchers to analyze SSL certificates, the EFF SSL Observatory project [17] offered the first large-scale, open certificate repository containing SSL certificates for the IPv4 address space in 2010. Later, in 2013, Durumeric et al. [15] analyzed the ZMap collected data over a period of 14 months to uncover all public certificate authorities (CAs) and the certificates they issued. Censys [13] is a search engine used to query information of hosts and networks stored in daily ZMap scans. As an example application for Censys, the prevalence of the unauthenticated Modbus protocol among SCADA systems has been studied. Numerous such systems have been found across the globe. However, non-SCADA devices, specifically, the TLS ecosystem for those devices have not been studied. We extend existing work to understand the TLS ecosystem for networked devices, mostly used at home, enterprise, and industrial environments, and physical/network infrastructures.

Heninger et al. [20] reported in 2012 that RSA/DSA algorithms as used specifically in embedded network devices are vulnerable due to faulty random number generators. They found that 0.75% of TLS certificates share keys, and RSA private keys can be easily calculated for 0.50% of TLS hosts (also reported similar results for RSA/DSA keys as used in the SSH protocol). However, other TLS/certificate parameters were not analyzed in this study.

Pa et al. [24] propose the IoT honeypot (IoTPOT) to analyze malware attacks against devices such as home routers, smart fridges, and other IoT devices. Their honeypot data also shows significant increase in Telnet-based attacks, including DDoS, against IoT devices. Costin et al. [7] devise a platform to find possible reuse of fingerprints of SSL certificates, public/private keys of devices in ZMap datasets; many devices were found with reused keys.

Shodan.io is a search engine similar to Censys, targeted towards IoT devices (full access requires paid subscriptions). In addition to IPv4 devices, Shodan claimed to have scanned millions of IPv6 addresses, reportedly by exploiting a loophole in the NTP Pool Project [3]. Arnaert et al. [2] highlight challenges in aggregating search results from Shodan and Censys, and propose an ontology to make these engines more usable and effective for finding vulnerable IoT devices.

3 Methodology and device info

We rely on the Censys [13] search engine for our analysis. In this section, we provide a brief overview of Censys, and detail our methodology.

Censys enables querying data from the Internet-wide scan repository (`scans.io`), a data repository hosting the periodic scan results as collected by the ZMap scanner [16]. Censys tags the collected data with security-related properties and device types, allowing easy but powerful search queries through its online search interface and REST API. Censys also tags TLS and certificate data of Alexa Top 1M web sites. Tagging is done by annotating the raw scan data with additional metadata, e.g., type and manufacturer for devices, and Alexa ranking for sites. The output from the application scanners is used to identify device-specific metadata. The annotation process involves ZTag (paired with ZMap and ZGrab), allowing researchers to add logic to define metadata for currently untagged devices [13]. Apparently, search capabilities in Censys is still evolving (not all device metadata is defined in ZTag, although ZTag can be

Device type	Non-TLS %	Non-TLS Count	TLS %	TLS Count
Infra. router	23.31	237,540	11.61	118,259
Modem	15.56	158,558	2.53	25,724
Camera	14.11	143,721	0.69	6809
NAS	7.07	71,997	5.45	55,503
Home router	5.04	51,347	2.52	25,667
Network	0.00	3	3.91	39,857
Printer	1.00	10,148	2.19	22,296
Scada	2.45	24,909	0.37	3773
CPS	1.26	12,820	0.09	868
Media	0.79	8000	0.11	1102
Total	70.57	719,043	29.43	299,858

Table 1: Type-wise device distribution

extended by other researchers); thus, TLS/certificate data and tag information for all device types are still not comprehensively reflected in Censys.

Table 1 lists available device types extracted from Censys, divided by TLS support. We further group some device types from Censys for easier presentation as follows: modem (cable/DSL), printer (all printer models, print servers), network (generic network devices, network analyzers), SCADA (scada controller, router, gateway, server, frontend), media (set-top box, digital video recorders, VoIP, cinema), CPS (PLC, HVAC, industrial control system, water flow controller, light controller, power distribution unit, power monitor, power controller). Certain device types (e.g., CPS) appear to be small in numbers. This may be due to the fact that the tagging process in Censys is still not very comprehensive. We do not consider some devices that are very low in number (e.g., 10 USB devices). The devices appear to come from all around the world (75 countries with >1000 devices); the top 10 countries host about 56% of all devices, including: Germany 17.9%, USA 15.0%, India 4.9%, and China 4.4%.

For comparison, we chose the Alexa Top 1M sites. Data extracted from Censys was transformed to an intermediary format that requires a resource-intensive post-processing phase. Search queries can be executed on Censys in two ways: a RESTful web API or an SQL interface engine. We used the latter option (with the help of a Censys author), as it is more efficient for large-scale search results. After the TLS parameters and certificates are extracted for devices and Alexa 1M sites, we first analyze our selected security parameters and algorithms in devices. We then compare the security parameters from devices with those from Alexa 1M sites, to highlight any important differences between them. Similar to past work (e.g., [15], [22]), we choose the following certificate/TLS parameters: cipher suite (algorithms used for hashing, key encryption, key exchange and authentication, signature), SSL/TLS protocol version, and RSA key length.

4 Analysis and results: Weak security practices

On October 8, 2016, we extracted certificates and TLS parameters (contained in a daily dump) from 299,858 TLS-supporting devices (out of a total of 1,018,911 devices), and from 598,888 HTTPS sites in Alexa Top 1M. The client used to extract TLS certificates are ZMap along with ZGrab (i.e., not following any popular browser), which is later queried from Censys. In this section, we provide the results of our analysis and compare the use of TLS/certificate parameters. For each cryptographic primitive in a device certificate and TLS/SSL protocol banner, we compute the percentage to compare the parameters between devices; see Figures 1- 5 for a comparison of the weak cryptographic primitives (for exact data, see [27]). We also compare average values from devices with Alexa sites (the last two bars). For brevity, we highlight results for algorithms and parameters that are most vulnerable.

Hash functions in message authentication. The use of SHA1 is prominent in all device types (67.4%), most notably in infrastructure routers (117,550, 99.4%) and network devices (35,918, 90.1%). In contrast, SHA1 usage in Alexa

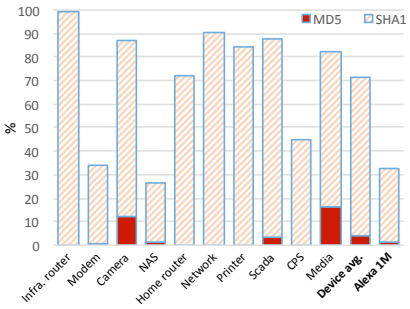

Fig. 1: Hashing algorithms

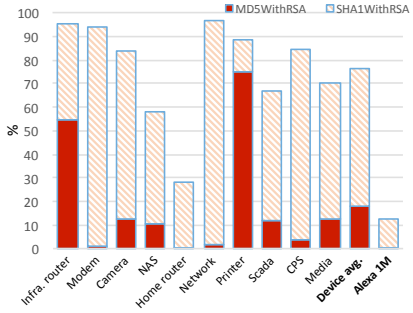

Fig. 2: Signature algorithms

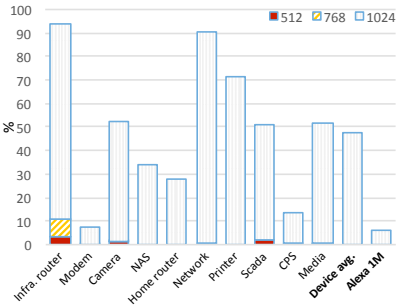

Fig. 3: Key lengths (RSA)

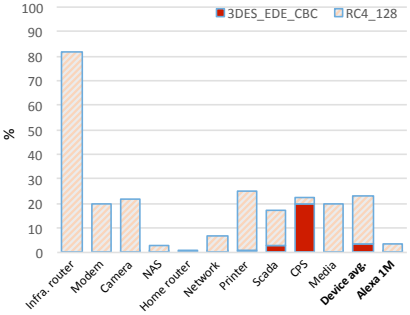

Fig. 4: Encryption algorithms

1M sites is far less (31.2%); see Figure 1. Some devices still use MD5, e.g., cameras (817, 12%) and media devices (176, 16%). MD5 is broken for more than a decade now [30], and SHA1 is also becoming subject to feasible collision attacks [28] (being phased out as of writing).

Hash functions in signature schemes. The MD5-RSA signature scheme is predominantly used in devices, notably in printers (16,993, 74.9%) and infra. routers (64,879, 54.9%); see Figure 2. These devices are vulnerable to certificate collision attacks, where attackers create certificates that collide with arbitrary prefixes/suffixes [29]. SHA1-RSA is also used more in modems (24,025, 93.4%), network (37,836, 94.9%) and CPS (703, 81%). A few devices (102) use "unknown" algorithms; according to a Censys author (email correspondence), these algorithms are not parseable.

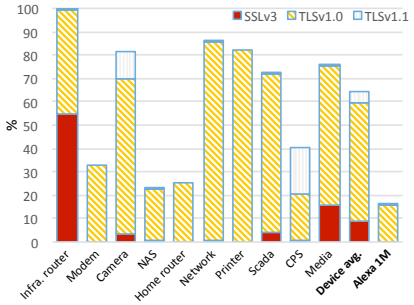

Fig. 5: SSL/TLS protocol versions

RSA key lengths. Certificates with 1024-bit RSA keys are deemed to be insecure as of early 2016; see NIST SP 800-131A (at least 2048 bits should be used). However, many devices still use 1024-bit keys (Figure 3); most notably infra. routers (98,432, 83.2%) and network devices (35,886, 90%). More seriously,

we found many devices with factorable 512-bit keys, e.g., infra. routers (3810, 3.2%), cameras (77, 1.1%) and scada devices (76, 2%).

Encryption algorithms. We check the use of vulnerable ciphers such as RC4 (see e.g., [18], RFC 7465), and 3DES (the recent Sweet32 attack [5]). Except infra. routers (96,433, 81.5%), the use of RC4 is relatively low in other devices (Figure 4). Some Alexa sites still use RC4 (3.1%). Note that the ZGrab application scanner as used with ZMap includes RC4 as a supported cipher (in addition to ciphers included in the Chrome browser), to support older TLS servers. The use of 3DES is very limited overall, except in CPS devices (171, 19.7%). The use of ChaCha20-Poly1305 (currently being standardized, RFC 7905) as a replacement of RC4 is still negligible in devices.

TLS/SSL version. TLS 1.0 is mostly used in network devices (33,637, 84.4%) and printers (18,367, 82.4%), and TLS 1.1 in CPS (168, 19.4%); see Figure 5. TLS 1.0 is vulnerable to the BEAST attack [12]. More seriously, many infra. routers (65,061, 55%) and media devices (175, 15.9%) use SSL 3.0 (vulnerable to the POODLE attack [23]). Surprisingly, 19,063 devices still support SSL 2.0 (deprecated in 2011, see RFC 6176). Top-5 such device types include: NAS (manufacturers: QNAP, NetGear, Synology; count: 5517), network (Cisco: 2006), printer (Lexmark, Sharp: 1812), camera (HikVision: 1324), and infra. router (Cisco: 1046). We do not include SSL 2.0 in Figure 5 or other comparisons, as SSL 2.0 dataset is separately maintained by Censys.

Manufacturer	MD5	RC4	SSLv3	<RSA1024	Device types
Cisco	347	98,904	65,413	12,731	Network, infra. router
Hewlett-Packard	1	5214	1	13	Network, printer, scada, home router
AVM	78	5062	33	2	Modem
Hikvision	664	1085	214	75	Camera
QNAP	383	889	286	51	NAS

Table 2: Top 5 manufactures with vulnerable devices

5 Disclosure

The vulnerable devices we found are manufactured by hundreds of different companies; see Table 2. We have contacted the ones with many vulnerable devices, where we could locate contact emails, explaining our findings (Oct. 2016). As of writing, we got responses from Cisco, Honeywell, Hikvision, and Hewlett Packard – most claiming to have released software/firmware upgrades in the past, but apparently, users did not follow. Example responses include: [Honeywell] "This helps a lot and as we have looked almost all of the systems you identified are "out of date" systems. Tridium/Honeywell released the patches to address your findings almost three years ago with follow on updates each year. The end users are not updating their systems to make them less vulnerable."

6 Limitations and future work

Certain statistics as extracted from Censys appear to be unusual. For example, there is only one infrastructure router from manufacturers, e.g., DrayTek and

LinkSys; Hewlett-Packard appears to have only one device with MD5 and SSLv3. We communicated such observations to a Censys author, who attributed them to be possible limitations of the current Censys logic, or device misconfiguration. Also, the SQL engine in Censys is still evolving. Currently, it does not allow querying all device-related information in a flexible structural format from the data available in ZMap. We plan to extend the comparison including all IPv4 web servers, when data hygiene and structure of data improve in Censys.

Some TLS vulnerabilities may have no effect if the services are accessed within a local network (e.g., inside a private home network), or via a modern browser—e.g., no current browser would accept the RC4 cipher or SSL 2.0, even if offered by a server. As these devices are varied (unlike regular web servers), actual exploitation of their weaknesses will depend on how they are used/accessed. These seemingly obsolete attack vectors can also be revived in the presence of a vulnerable TLS proxy between a modern browser and the vulnerable server, such as an anti-virus proxy [11]; simply supporting SSL 2.0 can be exploited as well [4]. We hope our findings to raise awareness of this issue and positively influence the manufactures to push appropriate firmware upgrades (possibly with auto-update).

Acknowledgements

We thank anonymous FC 2017 and IMC 2016 reviewers for their insightful comments and suggestions, and Zakir Durumeric for helping us with Censys. We also appreciate the feedback we received from the members of Concordia's Madiba Security Research Group, especially, Xavier de Carné de Carnavalet. The second author is supported in part by an NSERC Discovery Grant.

References

1. D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelink, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *CCS'15*, Denver, CO, USA, Oct. 2015.
2. M. Arnaert, Y. Bertrand, and K. Boudaoud. Modeling vulnerable Internet of Things on SHODAN and CENSYS: An ontology for cyber security. In *SECURE-WARE'16*, Nice, France, July 2016.
3. ArsTechnica.com. Using IPv6 with Linux? you've likely been visited by Shodan and other scanners. News article (Feb. 1, 2016).
4. N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, E. Käsper, S. Cohney, S. Engels, C. Paar, and Y. Shavitt. DROWN: Breaking TLS using SSLv2. In *USENIX Security*, Austin, TX, USA, Aug. 2016.
5. K. Bhargavan and G. Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In *CCS'16*, Oct. 2016.
6. T. Chung, Y. Liu, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Measuring and applying invalid SSL certificates: The silent majority. In *IMC'16*.
7. A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A large-scale analysis of the security of embedded firmwares. In *USENIX Security*, Aug. 2014.

8. A. Costin, A. Zarras, and A. Francillon. Automated dynamic firmware analysis at scale: A case study on embedded web interfaces. In *ASIACCS'16*, 2016.
9. A. Cui, M. Costello, and S. J. Stolfo. When firmware modifications attack: A case study of embedded exploitation. In *NDSS'13*, San Diego, CA, USA, Feb. 2013.
10. A. Cui and S. J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In *ACSAC'10*, Dec. 2010.
11. X. de Carnavalet and M. Mannan. Killed by proxy: Analyzing client-end TLS interception software. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
12. T. Duong and J. Rizzo. Here come the ⊕ ninjas. Technical report (May 2011).
13. Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. Halderman. A search engine backed by Internet-wide scanning. In *CCS'15*, Denver, CO, USA, Oct. 2015.
14. Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson. The matter of Heartbleed. In *IMC'14*, Vancouver, Canada, Nov. 2014.
15. Z. Durumeric, J. Kasten, and M. Bailey. Analysis of the HTTPS certificate ecosystem. In *IMC'13*, Oct. 2013.
16. Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast internet-wide scanning and its security applications. In *USENIX Security*, Aug. 2013.
17. Electronic Frontier Foundation. The EFF SSL observatory. `https://www.eff.org/observatory`.
18. C. Garman, K. G. Paterson, and T. Van der Merwe. Attacks only get better: Password recovery attacks against RC4 in TLS. In *USENIX Security*, Aug. 2015.
19. M. Hastings, J. Fried, and N. Heninger. Weak keys remain widespread in network devices. In *IMC'16*, Santa Monica, CA, USA, Nov. 2016.
20. N. Heninger, Z. Durumeric, E. Wustrow, and J. Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *USENIX Sec.*, 2012.
21. R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
22. H. Lee, T. Malkin, and E. Nahum. Cryptographic strength of SSL/TLS servers. In *IMC'07*, San Diego, CA, USA, Oct. 2007.
23. B. Möller, T. Duong, and K. Kotowicz. This POODLE bites: Exploiting the SSL 3.0 fallback. Techical report (Sept. 2014). `https://www.openssl.org/~bodo/ssl-poodle.pdf`.
24. Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow. IoTPOT: Analysing the rise of IoT compromises. In *USENIX Security*, 2015.
25. ReadWrite.com. Dyn DDoS attack sheds new light on the growing IoT problem. News article (Oct. 24, 2016).
26. E. Ronen, C. O'Flynn, A. Shamir, and A.-O. Weingarten. IoT goes nuclear: Creating a ZigBee chain reaction. Cryptology ePrint Archive, Report 2016/1047, 2016.
27. N. Samarasinghe and M. Mannan. Short paper: TLS ecosystems in networked devices vs. web servers. Technical Report 982186, Concordia University, Feb. 2017. `http://spectrum.library.concordia.ca/982186/`.
28. M. Stevens, P. Karpman, and T. Peyrin. Freestart collision for full SHA-1. In *Eurocrypt'16*, Vienna, Austria, May 2016.
29. M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. Osvik, and B. de Weger. Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In *CRYPTO'09*, Santa Barbara, CA, USA, Aug. 2009.
30. X. Wang and H. Yu. How to break MD5 and other hash functions. In *Eurocrypt'05*, Aarhus, Denmark, May 2005.