Secura

THE STATE OF DUTCH SSL/TLS CERTIFICATES

Ralph Moonen, Technical Director & Tom Tervoort, Security Specialist at Secura

August 2018

SECURA

Vestdijk 59
5611 CA Eindhoven
Netherlands

Karspeldreef 8
1101 CJ Amsterdam
Netherlands

T +31 (0)40 23 77 990
E info@secura.com
W www.secura.com

Follow us on

The State of Dutch SSL/TLS certificates

Introduction

Transport Layer Security (TLS)[1], also known as the Secure Sockets Layer (SSL), is the most important security mechanism in use on the internet at the moment. It is based on a set of protocols, each of which use a number of cryptographic algorithms to secure transport of data over untrusted networks for the websites we use every day including banks, Facebook, Twitter etc. It is to be expected that its use will grow further, as data protections laws evolve, and it is our expectation that the European Union at some point in the future will make the use of encryption technology mandatory for all web sites.

TLS is a technology that has evolved over the past few years and has known quite a few vulnerabilities. The most used implementation is OpenSSL, which has also seen quite a few rather bad vulnerabilities. TLS is also a technology that is quite finicky, in that you need to get everything perfectly configured for it to work properly: it supports weak encryption algorithms, so you must disable them; if you self-sign your certificate it isn't valid; if you did not change your private keys after Heartbleed, you have no security (a significant percentage of sites updated the software but forgot to change their compromised encryption keys. I suspect the NSA still makes jokes at the lunch table about this). It goes on: if your keys are not random or really prime, they can be cracked; and finally, let's not forget that issuers of certificates can be hacked (Diginotar) or simply have bad security practices (https://www.bleepingcomputer.com/news/security/google-outlines-ssl-apocalypse-for-symantec-certificates/). It's not that it is difficult to get it right, but it is a cumbersome process.

Authentication of the websites is performed using so-called X.509-certificates that contain public keys, which are used to prove that a communication partner knows a corresponding private key. If the public key is short or can be cracked, the private key can become known, and subsequent communication with that system can be decrypted. If the certificate is not valid, we can not know with certainty the identity of the communication partner. It is therefore imperative that TLS certificates are valid, and secure.

Periodically, we like to look at the state of the certificates used for TLS in use in the Netherlands. We especially look at the most common cryptographic algorithm, RSA[2].

[1] https://en.wikipedia.org/wiki/Transport_Layer_Security
[2] https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Subject and Approach

In this article, we research the validity and security parameters of TLS certificates in use in The Netherlands. In order to do this, we performed a scan of most[3] IPv4 addresses that are routed into The Netherlands, according to RIPE. The RIPE database is located at https://stat.ripe.net/NL#tabId=database and reflects the current known state of IP ranges routed into the Netherlands. These ranges contain many millions of IP addresses, a fraction of which actually connect to active hosts (a growing issue, because not all of the limited number of IPv4 addresses is in use, however are claimed and unused, while not being useful).

In order to make any statement on certificates in our country, we need to scan all Dutch IP addresses and extract the TLS certificates from them.

Method

We used the stateless port scanner zmap[4] to scan for TLS-enabled web services and some other services such as TLS-enabled email, FTP or LDAP services. Using OpenSSL we extracted the certificates. In total a little less that 500.000 certificates were downloaded.

Please note that there are many more TLS certificates for Dutch domain names (ending in .nl) or other Dutch entities that utilise mostly foreign hosting services that therefore are not included in this research. In order to stay within solely Dutch territory (and jurisdiction), we used the Dutch IP lists, and not the Dutch domain lists.

The ~500.000 certificates were analysed for various aspects, such as issuer, expiration dates, algorithms supported, key lengths, and weak keys. Besides the statistics, we have an interesting finding regarding a specific brand of commercial equipment, which we will elaborate on a little further on in this article.

[3] We excluded the large DSL and Fiber home ranges of KPN, Ziggo and others because they relate to private individuals.

[4] https://zmap.io/

Some Numbers!

The most used certificate issuers are:

1. *99194: C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Domain Validation Secure Server CA*
2. *66075: C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3*
3. *18047: C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Organization Validation Secure Server CA*
4. *16528: C=US, ST=Someprovince, L=Sometown, O=none, OU=none, CN=localhost/emailAddress=webmaster@localhost*
5. *16300: C=TW, ST=HsinChu, L=HuKou, O=DrayTek Corp., OU=DrayTek Support, CN=Vigor Router*
6. *12805: C=US, ST=Arizona, L=Scottsdale, O=GoDaddy.com, Inc., OU=http*
7. *12268: C=US, O=GeoTrust Inc., CN=RapidSSL SHA256 CA*
8. *8801: C=US, ST=TX, L=Houston, O=cPanel, Inc., CN=cPanel, Inc. Certification Authority*
9. *6540: C=US, O=GeoTrust Inc., CN=RapidSSL SHA256 CA - G3*
10. *6369: C=US, ST=Virginia, L=Herndon, O=Parallels, OU=Parallels Panel, CN=Parallels Panel/emailAddress=info@parallels.com*

"Let's Encrypt" has very quickly gained the number two spot after Comodo for most popular CA (not surprising since they give away free certificates). About half of all certificates given out by actual CA's, the rest is self-signed or issued by an unaccredited CA. This in itself means that half of the certificates can not be trusted. Luckily your browser will warn you for these and you have the option of not accepting the connection (which you should do). It also means that half of all Dutch businesses don't really care (or know) about providing a good level of security for the publicly accessible web interfaces. A pretty sad state! The Dutch Governmental CA, PKIOverheid, falls just outside the top 10 with 4986 issued public certificates.

The top 5 of signature algorithms used in The Netherlands is as follows:

1. *375686: sha256WithRSAEncryption*
2. *102735: sha1WithRSAEncryption*
3. *7049 md5WithRSAEncryption*
4. *1148 ecdsa-with-SHA256*
5. *1084 sha512WithRSAEncryption*

Apparently, RSA is very popular! However, in order to be able to set an unforgeable signature, it is also important to use a secure cryptographic hash function. We see the following distribution of hash functions:

- SHA-2-family: used by 77.7% of certificates
- SHA-1: 20.8%
- MD5: 1.5%

No attacks are known against any of the SHA-2 hash functions. However, SHA-1 is deprecated, and has been definitively broken at the start of 2017. Forging certificates signed with SHA-1 (and thus breaking TLS security) is expected to become practical in the short term. MD5, on the other hand, can already be broken for the cost of about 50 cents of computing time per certificate. The fact that weak hash functions are still regularly used shows that TLS is certainly no guarantee for security. Browsers might complain about them but the average user has no clue what is going on here, leading to widespread acceptation of bad certificates by the public.

We also examine the size of the RSA keys that are being used, and see the following results:

- 2048 bits or more: 88.76%
- 1024 bits, or in between 1024 and 2048 bits: 11.17%
- Less than 1024 bits: 0.07%

Keys of 2048 bits are nowadays considered to be sufficiently large. 1024-bit keys are no longer recommended, although apparently, they are still being used by about a one in ten sites we scanned. The largest known RSA public key that has been factored (i.e. broken) to date is 768 bits long.

Collisions

A largely overlooked but nevertheless fun bit of public key encryption is the fact that public RSA keys are used and published in TLS certificates and many other places. If one where to crack the public key and derive the private key from it, there would be a huge problem. Debit cards, web sites and many digital identities would be compromised. In fact, an algorithm for quantum computers was designed by Peter Shor that can crack public keys! Luckily no one has created a quantum computer powerful enough to do this, but this is just a matter of time. For now however, we must rely on other techniques if we want to crack RSA keys. Let's look at the math to understand this.

The public key contains a term that consists of the product of two prime numbers. So its only factors are those two primes. It is trivial to calculate the product of two primes. But given the result, it turns out to be very inefficient to find the factors of such a number. This is the basis of the RSA algorithm.

One could try all possible combinations of prime numbers multiplied with every possibility. However this is well beyond our current capacities. But what if you could predict one of those numbers? How do we even *make* prime numbers that are large enough to be impervious to a brute force attack?
It turns out that if your primes that are calculated during key generation are predictable or not really primes, bad things happen. A nice, recent example of this is the ROCA vulnerability that affected many smartcards (see https://www.secura.com/en/blog-a-bad-week-for-crypto-ROCA).

Every number can be factorised into its prime factors. But generating a number p*q that is hard to factorise relies on generation of random numbers. Computers are very bad at making random numbers: they are designed for deterministic behaviour. So they rely on sources of entropy to gather enough 'randomness' and then use that randomness to generate random primes.

It is common that these sources do not provide sufficient entropy, though. A common problem is that devices may generate their keys right after being booted for the first time, before the OS has been able to gather sufficient entropy. Under some circumstances, this may result in this entropy being predictable or identical among different devices.

Also, if the numbers are not generated randomly, some prime factors are more likey to occur than others. If two products p*q share a prime, you can even calculate their greatest common denominator (GCD) very easily using Euclid's algorithm: https://en.wikipedia.org/wiki/Euclidean_algorithm.

If numbers were really random, we would have no chance of findings such GCD collisions: keys range from 1024 to 4096 bits length, which means the prime numbers range from 512 to 2048 bits. That means that your prime numbers are on the order of 2^512: over 150 digits long.

We can calculate the number of primes of that length, see http://primes.utm.edu/howmany.shtml. There are 2.8^147 primes to choose from! If random number generator were perfect, we would have no chance….

Some years ago, the scientists behind https://factorable.net/ showed that many public keys in certificates share the same primes, due to bad random number generators, and can be cracked. Cracked keys means an attacker can impersonate this web site, and decrypt intercepted traffic. We thought it would be interesting to revisit this research and apply it to the Dutch IP space. Therefore we extracted the public keys from all Dutch certificates, and ran fastGCD. After an hour or crunching numbers, our laptop spat out no less than 113 broken certificates.

A Closer Look

A closer look at these certificates show something interesting: none of these certificates are issued by an accredited CA. It is likely that the keys used within these certificates are system-generated, or generated on network appliances. If that is indeed the case, it would imply that that these appliances have bad cryptographic random number generators (known as Pseudo Random Number Generators, or PRNGs) and suffer from weaknesses in the generation of cryptographic keys. If they had cryptographic random number generators that function correctly, we would certainly not find any collisions at all in a small data set of ~500.000. However we found 113, which is statistically very close to impossible unless the devices suffer from grave weaknesses in their PRNGs (or in the method that is used to initialize them).

The devices belong to a wide range of organisations: schools, small businesses, a Dutch province, a 'Veiligheidsregio' and a couple of IT companies. We will not disclosure their identities here, we provided the NCSC with the details.

Older Netgear SSL VPN appliances show several dozen certificates cracked. This could possibly mean that these devices have a deficient PRNG. If that were to be the case for the TLS certificates installed on the web interface, it would probably also be the case for the VPN keys, meaning that potentially all communication is at risk. Further research is required to determine whether VPN communications with these devices is vulnerable. Netgear is investigating the issue.

We also found a number of DLINK and Fortinet devices as well as a number of 'System generated' keys that we have not be able to reliable fingerprint as of yet. This brings to mind the recently discovered DUHK vulnerability (https://duhkattack.com/), which was caused by issues with random number generators on Fortinet devices. It is unlikely, however, that the duplicate prime factors are caused by this particular issue.

It is good to note that we compared the Dutch IP space to the global database of TLS certificates as published by Censys (www.scans.io). Many thousands more vulnerable TLS-certificates exist worldwide. Finding these is left as an exercise to the reader....

Conclusion

If you use TLS, you must use good practices. We have found that approximately half of the users of TLS do not use it properly. A very small, but very significant percentage of public RSA keys can be cracked, showing that especially certain older models of Fortinet, Dlink and Netgear appliances may have had their keys generated using deficient PRNGs.

About Secura

Secura has worked in information security and privacy for nearly two decades. This is why we uniquely understand the challenges that you face like no one else and would be delighted to help you address your information security matters efficiently and thoroughly. We work in the areas of people, processes and technology. For our customers we offer a range of security testing services varying in depth and scope.

Secura has the mission to support organizations with up-to-date knowledge to work toward a bright and safe future.

Keep updated with the latest insights on digital security and subscribe to our periodical newsletter.

Interested?

Contact us today at info@secura.com or visit secura.com for more information.

SUBSCRIBE

TO OUR NEWSLETTER

Secura