KRIPTOGRÁFIAI ALAPOK FEJLESZTŐKNEK ÉS ÉRDEKLŐDŐKNEK

BEVEZETÉS A KRIPTOGRÁFIAI AUTENTIKÁCIÓ BIRODALMÁBA

Veres-Szentkirályi András 2020-07-25

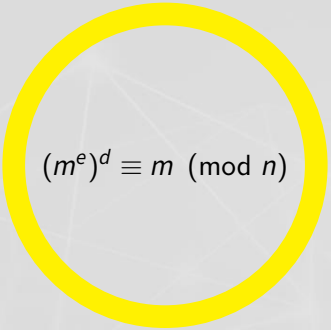
$ whoami

Veres-Szentkirályi András

- OSCP, GWAPT, SISE, HAREC
- Silent Signal társalapító (2009-)
- Etikus hackelés + oktatás
- 15 év tapasztalat

Disclaimer

silent signal



Az előadást kriptográfusok számára a szövegben előforduló általánosítások és egyszerűsítések miatt csak laikus társaságában ajánljuk!

Menetrend

1. Általános kriptográfiai bevezetés
2. Kriptográfiai kivonatok (digest)
3. Kriptográfiai aláírások (digital signature)

Egy fontos gondolat előszó helyett

silent signal

“Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can’t break.”

– Bruce Schneier (1998. október 15.)

https://www.schneier.com/crypto-gram-9810.html#cipherdesign

Különlegesek-e a biztonsági hibák?

silent signal

From: Linus Torvalds <torvalds@linux-foundation.org>
Date: Tue, 15 Jul 2008 02:28:23 UTC

On Tue, 15 Jul 2008, pageexec@freemail.hu wrote:
> so guys (meaning not only Greg but Andrew, Linus, et al.), when will you
> publicly explain why you're covering up security impact of bugs? and even
> more importantly, when will you change your policy or bring your process
> in line with what you declared?

We went through this discussion a couple of weeks ago, and I had
absolutely zero interest in explaining it again.

I personally don't like embargoes. I don't think they work. That means
that I want to fix things asap. But that also means that there is never a
time when you can "let people know", except when it's not an issue any
more, at which point there is no _point_ in letting people know any more.

So I personally consider security bugs to be just "normal bugs". I don't
cover them up, but I also don't have any reason what-so-ever to think it's
a good idea to track them and announce them as something special.

So there is no "policy". Nor is it likely to change.

Linus

Mi tartozik ide?

silent signal

- (adat) titkosítás: üzenetek, hálózati forgalom
- (adat) autentikáció: üzenetek, hálózati forgalom, futtatható kód
- kiegészítők: kulcsszármaztatás, kulcscsere
- protokollok mindezek köré

Adat autentikáció

silent signal

- integritásvédelem: sérült-e véletlenül vagy szándékosan átvitel közben
- feladóhoz kötés: attól jött-e, akinek mondja magát
- összetett építmény: sok szereplő, sok műszaki elem
- nem véd minden ellen: replay attack

Menetrend

1. Általános kriptográfiai bevezetés
2. Kriptográfiai kivonatok (digest)
3. Kriptográfiai aláírások (digital signature)

Hash innen!

silent signal

- bővebben lásd
  - `https://silentsignal.hu/docs/S2_VSzA_HWSW_hash_2017.pdf`
  - `https://silentsignal.hu/docs/S2_VSzA_HWSW_hash_sysadminday_2017.pdf`
- lényeg: tetszőleges bemenet $\Rightarrow$ fix hosszúságú kimenet
- asszociatív tömb, `dict`, Perl `hash`, Java `Hash{Map,Set}`: szórjon jól és gyorsan (DoS!)
- integritásvédelem vétlen módosítás ellen: elég(?) 32 bit
- integritásvédelem rosszindulatú manipuláció ellen: erről lesz szó

Digest #minekvan

silent signal

Legegyszerűbb: ha már van egy megbízható csatorna, amin csak hash megy át

```
<script src="//code.jquery.com/jquery-1.12.4.min.js" type="text/javascript"
  integrity="sha256-ZosEbRLbNQzLpnKIkEdrPv7lOy9C27hHQ+Xp8a4MxAQ="
  crossorigin="anonymous"></script>
```

Lásd még: PGP fingerprint névjegykártyán, SSL/TLS fingerprint levélfejlécben

Menetrend

1. Általános kriptográfiai bevezetés
2. Kriptográfiai kivonatok (digest)
3. Kriptográfiai aláírások (digital signature)

Kriptográfiai aláírás

silent signal

- titkos kulcs: aláírást készít
- publikus kulcs: ellenőrizhető vele az aláírás
- titkos ⇒ publikus: triviális
- publikus ⇒ titkos: ideális esetben nagyon nehéz
- elterjedt algoritmusok: EdDSA, RSA, ECDSA, DSA (ajánlott … kevésbé ajánlott)
  - RSA: footgun, RSA-PKCS#1 v1.5 vs. RSA-PSS
  - ECDSA: `CVE-2020-0601`
  - DSA: `https://buttondown.email/cryptography-dispatches/archive/557475c5-9781-47e0-a640-5734bc849bc7`

Mire jó egy aláírás?

silent signal

- Varázspénz: tárca = titkos kulcs, aláírja a tranzakciót
- TLS: HTTPS, IMAPS, SMTPS, WPA Enterprise
- SSH: szerver és opcionálisan kliens autentikáció
- S/MIME és PGP: e-mail aláírás
- kódaláírás: Windows, Android, Apple világ, Linux disztribúciók
- DKIM: e-mail hitelesítés
- VPN-ek: IPsec, OpenVPN, Wireguard
- WSSE, JWT: lol

Publikus kulcs logisztika

silent signal

- támadói modell tartalmazza a módosítást? ha nem, triviális (DKIM?)
- publikus kulcs (RSA: kivonat) névjegykártyán, fejlécben
- Wireguard, JWT, SSH kliens auth, varázspénz: publikus kulcs „hardcoding” (utóbbinál kivonat)
- SSH szerver auth: TOFU – meglepően jó!
- PGP: kulcsszerverek, web of trust
- OpenBSD: minden verzióban következő két major release publikus kulcsa
- PKI: a pokol külön bugyra (TLS, S/MIME, kódaláírás, OpenVPN, néha IPsec és JWT)

PKI

silent signal

- CA: aláír egy publikus kulcs + identitás párost
- Ki dönti el, melyik CA-(k)ban bízom meg?
  - `https://buhera.blog.hu/tags/pki`
  - `https://blog.mozilla.org/security/2018/03/12/distrust-symantec-tls-certificates/`
- Érvényességi idő: örök vita
- Visszavonás: CRL, OCSP, OCSP stapling
- Érvényességi időn kívüli használat: időbélyegzés

Mi rossz történhet? (©HBO)

silent signal

1. példa: Erlang HTTP kliens

“It never even occurred to me that an http client would be insecure by default when connecting over https.”

```
$ curl https://self-signed.badssl.com/
curl: (60) SSL certificate problem: self signed certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

3> httpc:request("https://self-signed.badssl.com/").
{ok,{{"HTTP/1.1",200,"OK"},
     [{"cache-control","no-store"},
      {"connection","keep-alive"},
      {"date","Fri, 24 Jul 2020 03:13:37 GMT"},
      ...
```

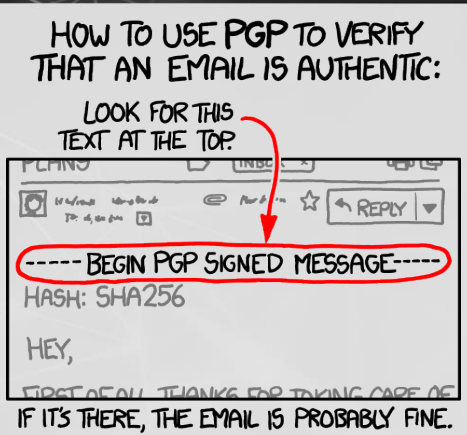
2. példa: Android (2013)

silent signal

- https://buhera.blog.hu/2013/07/08/andropokalipszis
- Java implementáció aláírás ellenőrzésére
  - “Later, the PackageParser goes through each entry in the zip file, verifying that the file was signed with a consistent signature. This code iterates over the LinkedHashMap. The result is that only the last entry with a given name is considered for signature verification: all previous duplicates are discarded.”
- C implementáció futtatásra
  - “This algorithm is an unchained hashtable with linear probing, without replacement. The result is that every entry in the original zip file is placed into the array. The algorithm for finding entries is the same as that for adding them: you scan forward looking for a match. This means earlier entries are used instead of later ones.”
  - “If you thereby take an existing APK file and add a new entry to the zip file for "classes.dex" that comes before the one that is already in the APK, it will continue to verify correctly (as the second instance of the entry will be used by the verifier) but the modified file will be loaded by the VM (as it comes first).”
- Kiváló writeup, fenti idézetek is innen vannak: http://www.saurik.com/id/17

Futtatható kód aláírása

- támadói modell: kompromittált mirror, véletlen hálózati/diszk hiba
- kivonat, aláírt kivonat, tanúsítvány, időbélyeg, saját CA, saját CRL/OCSP, …
- gyakori tévhit: ettől még lehet rosszindulatú a kód
  - Stuxnet (2010): lopott certek
  - Flame (2012): MD5 ütközés ⇒ MS cert
  - RobbinHood (2020): sebezhető Gigabyte drivert vitt, aminek még nem vonta vissza a Verisign a certjét



https://xkcd.com/1181/

Útravaló: NaCl/libsodium

silent signal

- `https://download.libsodium.org/doc/` (natív)
- `https://github.com/joshjdevl/libsodium-jni` (Java)
- `https://github.com/adamcaudill/libsodium-net` (.Net)

Összefoglalás

silent signal

- a biztonsági hibák nem „sima” bugok
- jól megválasztott algoritmussal kriptográfiai kivonat egyenértékű annak bemenetével
- aláírt kivonattal ellenőrizhető az üzenet hitelessége
- tanúsítvánnyal ellenőrizhető(?) az aláíró kiléte
- időbélyeg/OCSP/CRL segít ellenőrizni az aláírás megbízhatóságát
- mindez sorosan kapcsolva: bármi bugos, a támadó nyer
- mit sem ér az erős elem, ha a leggyengébb láncszem törhető
  - `https://buhera.blog.hu/2014/02/22/apple_vs_ssl_goto_fail`
  - `https://buhera.blog.hu/2014/03/04/gnutls_vs_ssl_goto_cleanup`

KÖSZÖNÖM!

VERES-SZENTKIRÁLYI ANDRÁS

vsza@silentsignal.hu

facebook.com/silentsignal.hu

@SilentSignalHU

@dn3t

silent signal