MURPHY ÉS A SECDEV

NÉHÁNY TANULSÁGOS SECURITY FAIL TÖRTÉNET

Veres-Szentkirályi András **2020-11-16**

$ whoami

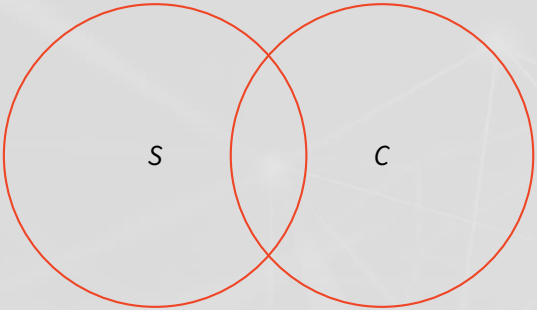
Veres-Szentkirályi András

- OSCP, GWAPT, SISE
- Silent Signal alapító
- pentester, toolmaker

Menetrend

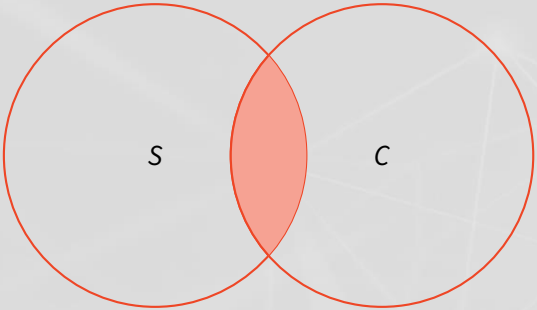
1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

Offenzív vs. defenzív megközelítés



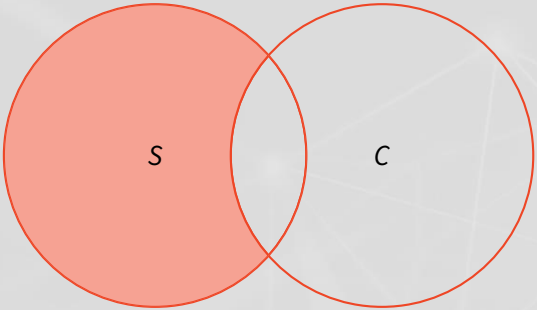
- *S* specifikáció
- *C* kódbázis

Offenzív vs. defenzív megközelítés



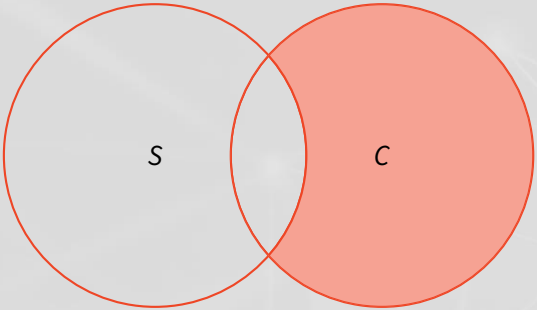
- S specifikáció
- C kódbázis
- $S \cap C$ specifikációnak megfelelően működő kód (ideálisan $S = C$)

Offenzív vs. defenzív megközelítés



- S specifikáció
- C kódbázis
- $S \cap C$ specifikációnak megfelelően működő kód (ideálisan $S = C$)
- $S - C$ specifikációban szereplő, de kódban **nem** implementált funkció
  - bug, funkcionális hiba

Offenzív vs. defenzív megközelítés



- S specifikáció
- C kódbázis
- $S \cap C$ specifikációnak megfelelően működő kód (ideálisan $S = C$)
- $S - C$ specifikációban szereplő, de kódban **nem** implementált funkció
  - bug, funkcionális hiba
- $C - S$ specifikációban **nem** szereplő, de kódban implementált funkció
  - feature, potenciális biztonsági hiba

Menetrend

1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

Számlaegyenleg lekérdezése

silent signal

```
GET /app/szamlaegyenleg?szamla=20201116

SELECT egyenleg FROM szamla WHERE id = ?;
Szamla.objects.get(id=request.args['szamla'])
```

https://www.theregister.co.uk/2011/06/14/citigroup_website_hack_simple/

Web Application Firewall & co

silent signal

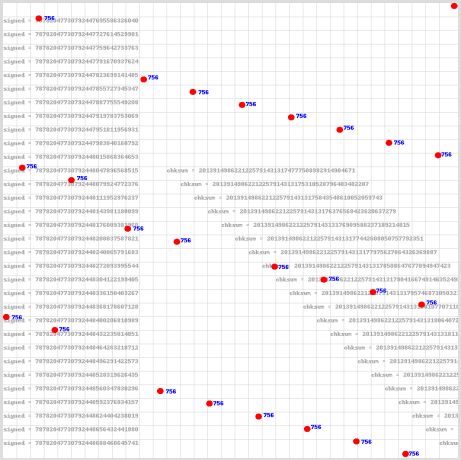
- fehérlista OK
- feketelista OK
- számlaegyenleg?
- fail-open?

Menetrend

1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

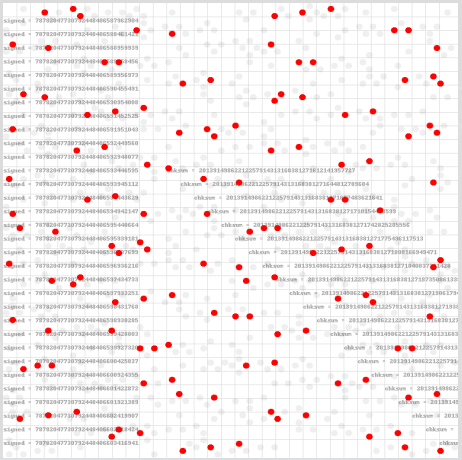
Saját hash vs. SHA-2 (20k tran)

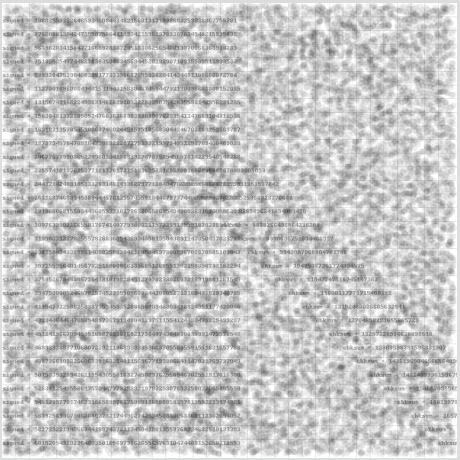
silent signal



Saját hash vs. SHA-2 (20k tran)

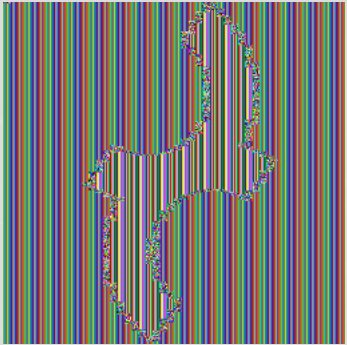
silent signal

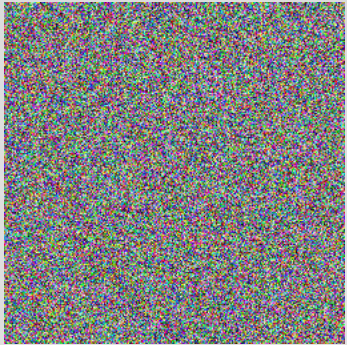




ECB vs. CBC

silent signal





- `openssl enc -aes-128-ecb -e -k demo -in ecbdemo-plain.data -out ecbdemo-ecb.data`
- `openssl enc -aes-128-cbc -e -k demo -in ecbdemo-plain.data -out ecbdemo-cbc.data`

Jó példa: cryptography.io

silent signal

`cryptography` is broadly divided into two levels. One with safe cryptographic recipes that require little to no configuration choices. These are safe and easy to use and don't require developers to make many decisions.

The other level is low-level cryptographic primitives. These are often dangerous and can be used incorrectly. They require making decisions and having an in-depth knowledge of the cryptographic concepts at work. Because of the potential danger in working at this level, this is referred to as the "hazardous materials" or "hazmat" layer. These live in the `cryptography.hazmat` package, and their documentation will always contain an admonition at the top.

`https://cryptography.io/en/latest/`

Menetrend

1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

Hash mint „entrópiaforrás”

silent signal

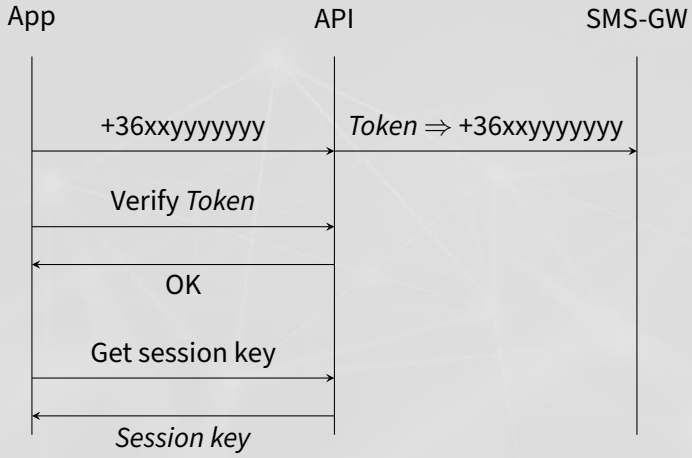
- `https://example.com/pwreset?token=xxx`
- csak alacsony entrópia bizonyítható
- black box: security by obscurity
- $RND() = H(P())$ ahol $P()$ kimenete megjósolható kiváló backdoor
  - gyakori tapasztalat: $P =$ `time(2)`
  - hardveres CSPRNG megbízhatósága?

Menetrend

1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

2FA, 1FA, 0FA

silent signal



Ami nem látható, az nem is létezik?

silent signal

```
if (request.user.hasRightTo("foobar")) {
  foobar.visible = true;
}
```

```
<input name="email" value="vsza@silentsignal.hu">
<input type="submit" value="Update user info">
```

Menetrend

1. Bevezetés
2. Horizontális privilégiumszint-emelés
3. Saját kriptográfia használata
4. Entrópia
5. Üzleti logikai szabály/sorrend megkerülése
6. Forráskód/resource menedzsment

Kulcskérdés

silent signal

```
Payload/Censored.app/***certificate $ grep -r 'BEGIN RSA PRIVATE KEY'
***T10/server.pem:-----BEGIN RSA PRIVATE KEY-----
***T5/server.pem:-----BEGIN RSA PRIVATE KEY-----
**********/server.pem:-----BEGIN RSA PRIVATE KEY-----
**********TESZT/server.pem:-----BEGIN RSA PRIVATE KEY-----
***INT/server.pem:-----BEGIN RSA PRIVATE KEY-----
********PUBLIC***/server.pem:-----BEGIN RSA PRIVATE KEY-----
***DEV/server.pem:-----BEGIN RSA PRIVATE KEY-----
***INT/server.pem:-----BEGIN RSA PRIVATE KEY-----
***ELES/server.pem:-----BEGIN RSA PRIVATE KEY-----
***FIT/server.pem:-----BEGIN RSA PRIVATE KEY-----
```

Összefoglalás

silent signal

- Ne csak „magunk ellen” védjük a rendszert!
- Gondoljunk arra, hogy mindig lesz olyan, aki nem tartja be a játékszabályokat!
  - sőt, tőlük kellene igazán védenie a rendszernek
  - „make invalid state unrepresentable”
- Szervezzük ki a szaktudást igénylő döntéseket!
  - de nem mindegy, hogy kinek
- Nyessük le a plusz „feature”-öket!
- A biztonság is legyen része a formalizált-automatizált minőségbiztosítási rendszernek!

KÖSZÖNÖM!

VERES-SZENTKIRÁLYI ANDRÁS

vsza@silentsignal.hu

facebook.com/silentsignal.hu

@SilentSignalHU

@dn3t

silent signal