

MURPHY ÉS A BIZTONSÁG

MI AZ, AMI ELROMOLHAT, ÉS EL IS ROMLIK?



Veres-Szentkirályi András 2019-04-24



Veres-Szentkirályi András

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

Menetrend

1 Bevezetés

- Horizontális privilégiumszint-emelés
- Saját kriptográfia használata
- Entrópia
- Üzleti logikai szabály/sorrend megkerülése
- Forráskód/resource menedzsment

Egy fontos gondolat előszó helyett



“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>

Grand Brighton Hotel, 1984



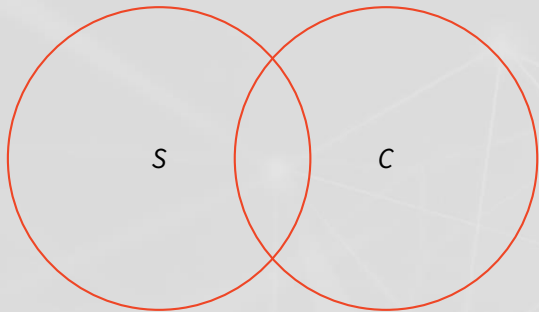
“Today we were unlucky, but remember **we only have to be lucky once. You will have to be lucky always.**”

– Provisional IRA (1984. október 13.)

Taylor, Peter (2001). Brits: The War Against the IRA.
Bloomsbury Publishing. p. 265. ISBN 0-7475-5806-X.

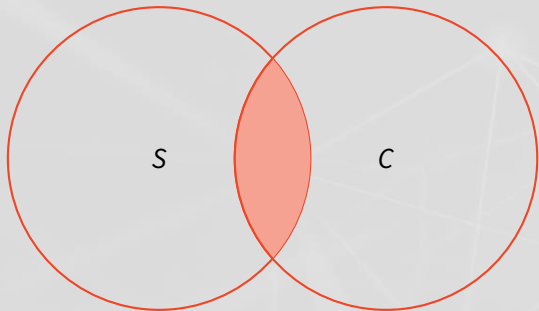
https://en.wikipedia.org/wiki/Brighton_hotel_bombing

Offenzív vs. defenzív megközelítés



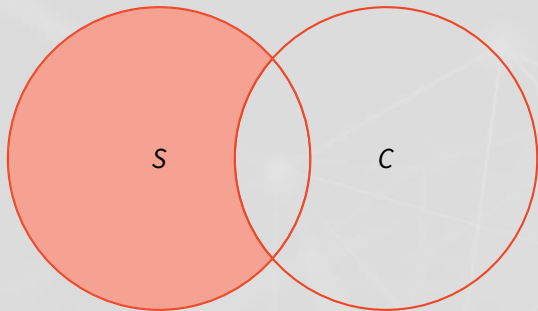
- ▶ S specifikáció
- ▶ C kódrendszer

Offenzív vs. defenzív megközelítés



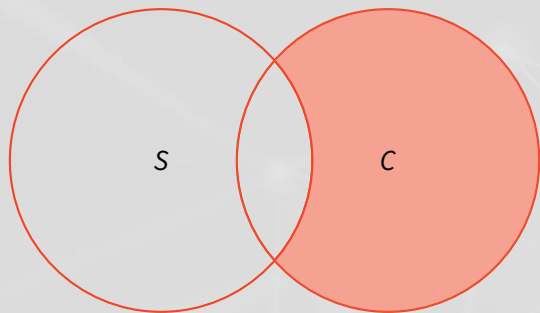
- ▶ S specifikáció
- ▶ C kódbázis
- ▶ $S \cap C$ specifikációnak megfelelően működő kód (ideálisan $S = C$)

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

- Bevezetés
- **2 Horizontális privilégiumszint-emelés**
- Saját kriptográfia használata
- Entrópia
- Üzleti logikai szabály/sorrend megkerülése
- Forráskód/resource menedzsment

Számlaegyenleg lekérdezése



```
GET /app/szamlaegyenleg?szamla=20190424
```

```
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



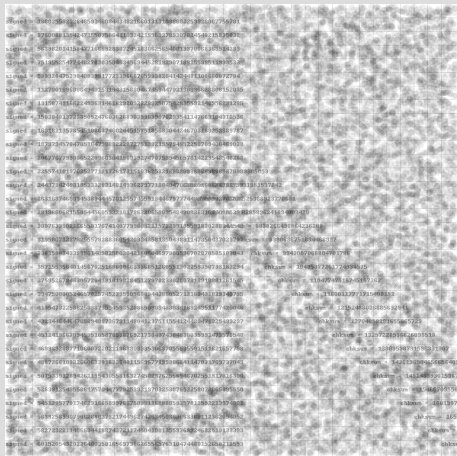
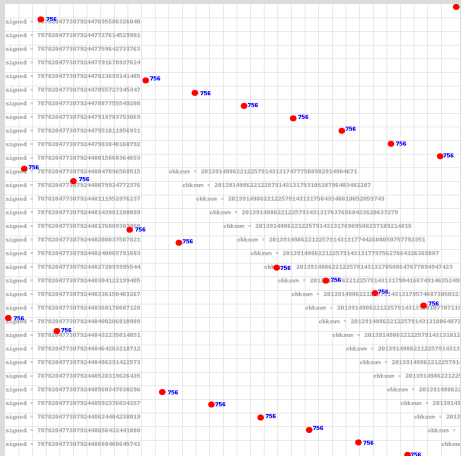
- ▶ fehérlista OK
- ▶ feketelista OK
- ▶ számlaegyenleg?
- ▶ fail-open?

Menetrend

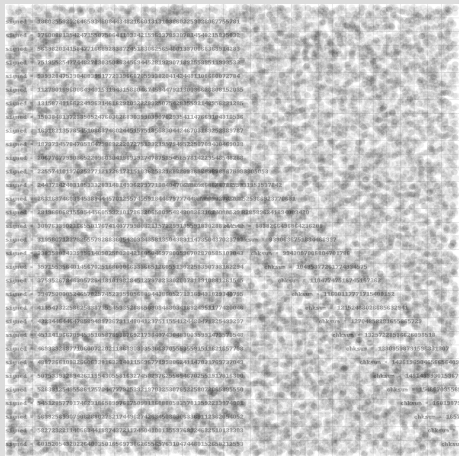
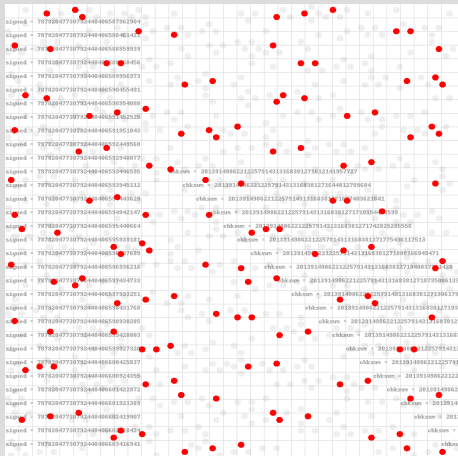
- Bevezetés
- Horizontális privilégiumszint-emelés
- 3 Saját kriptográfia használata
- Entrópia
- Üzleti logikai szabály/sorrend megkerülése
- Forráskód/resource menedzsment

- ▶ Előírás: 3DES, AES, HMAC, RSA, HMAC, PBKDF2 ...
- ▶ Szakma: ECDH, ChaCha20, Poly1305, Ed25519, scrypt ...
- ▶ <https://latacora.micro.blog/2018/04/03/cryptographic-right-answers.html>
- ▶ NaCl/Libsodium
 - ▶ <https://download.libsodium.org/doc/> (natív)
 - ▶ <https://github.com/joshjdevl/libsodium-jni> (Java)
 - ▶ <https://github.com/adamcaudill/libsodium-net> (.Net)

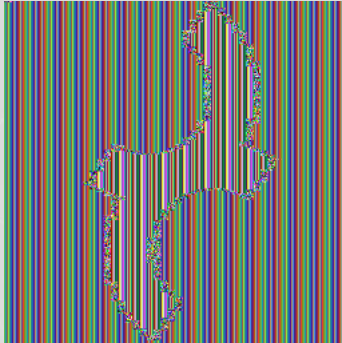
Saját hash vs. SHA-2 (20k tran)



Saját hash vs. SHA-2 (20k tran)



ECB vs. CBC



- ▶ `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



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

- Bevezetés
- Horizontális privilégiumszint-emelés
- Saját kriptográfia használata
- 4 ● Entrópia
- Üzleti logikai szabály/sorrend megkerülése
- Forráskód/resource menedzsment

- ▶ „információtartalom”
- ▶ Példa: <https://linusakesson.net/scene/a-mind-is-born/>
 - ▶ PCM/WAV soundtrack: 26 MB
 - ▶ MP3 soundtrack: 2,2 MB
 - ▶ C64 executable: 256 bájt
- ▶ relevancia hash-elésnél: hányat kell próbálkozzak?
- ▶ relevancia kulcsszármaztatásnál: hányféle kulcs létezhet?

Hash mint „entrópiaforrás”

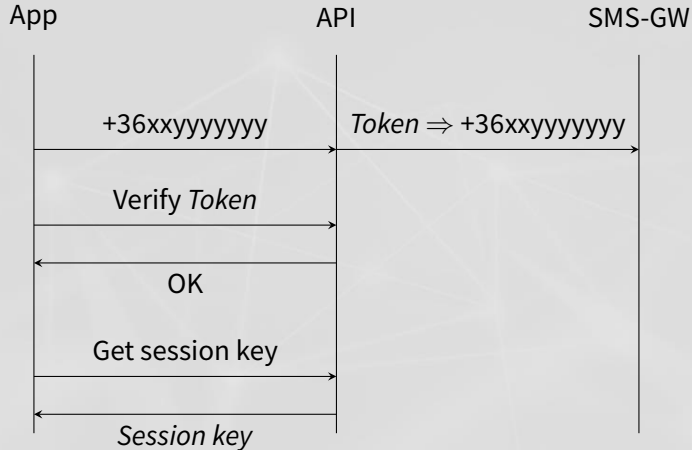


- ▶ `https://example.com/pwreset?token=xxx`
- ▶ csak alacsony entrópia bizonyítható
- ▶ black box: security by obscurity
- ▶ $RND() = H(P())$ ahol $P()$ kimenete megjósolható kiváló backdoor
 - ▶ gyakori tapasztalat: $P = \text{time}(2)$
 - ▶ hardveres CSPRNG megbízhatósága?

Menetrend

- Bevezetés
- Horizontális privilégiumszint-emelés
- Saját kriptográfia használata
- Entrópia
- 5 ● Üzleti logikai szabály/sorrend megkerülése
- Forráskód/resource menedzsment

2FA, 1FA, 0FA



Ami nem látható, az nem is létezik?



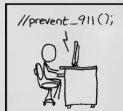
```
if (request.user.hasRightTo("foobar")) {  
    foobar.visible = true;  
}
```

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

Menetrend

- Bevezetés
- Horizontális privilégiumszint-emelés
- Saját kriptográfia használata
- Entrópia
- Üzleti logikai szabály/sorrend megkerülése
- 6 Forráskód/resource menedzsment

Kommentár nélkül (<https://xkcd.com/424/>)



IN THE RUSH TO CLEAN UP THE DEBIAN-OPENSSL FIASCO, A NUMBER OF OTHER MAJOR SECURITY HOLES HAVE BEEN UNCOVERED:

AFFECTED SYSTEM	SECURITY PROBLEM
FEDORA CORE	VULNERABLE TO CERTAIN DECODER RINGS
XANDROS (EEE PC)	GIVES ROOT ACCESS IF ASKED IN STERN VOICE
GENTOO	VULNERABLE TO FLATTERY
OLPC OS	VULNERABLE TO JEFF GOLDBLUM'S POWERBOOK
SLACKWARE	GIVES ROOT ACCESS IF USER SAYS ELVISH WORD FOR "FRIEND"
UBUNTU	TURNS OUT DISTRO IS ACTUALLY JUST WINDOWS VISTA WITH A FEW CUSTOM THEMES

```
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-----
```

- ▶ 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
- ▶ 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

