Introduction aux Padding Oracles

Résolution de l'épreuve Crypto 200 d'Insomni'hack 2011

Emilien Girault

e.girault@sysdream.com
@emiliengirault
www.segmentationfault.fr / www.ghostsinthestack.org

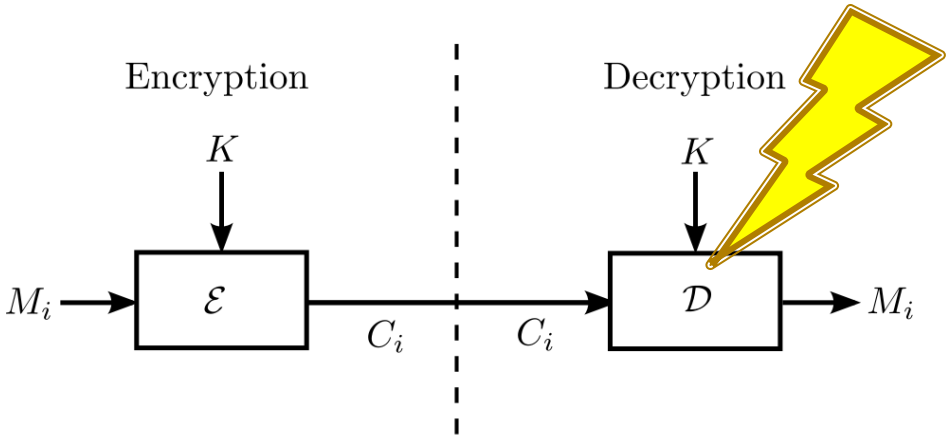
Sommaire

- Introduction
  - Chiffrement par bloc, CBC, padding PKCS#5
- Padding Oracle
  - Définition
  - Algorithme d'exploitation
- Insomni'hack 2011 Crypto 200
  - Epreuve
  - Résolution
- Références

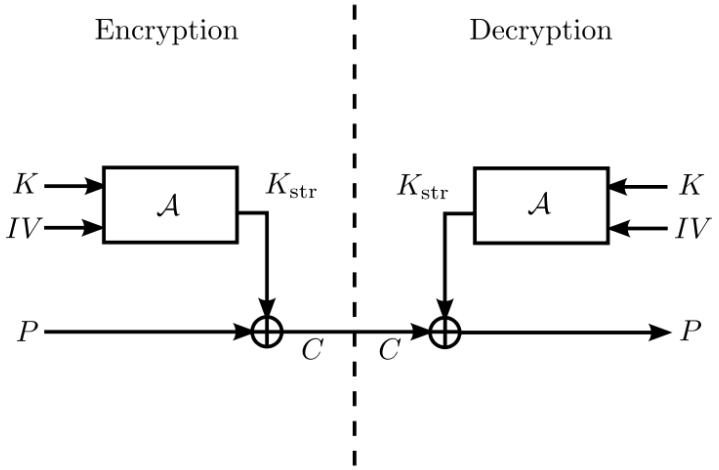
Introduction (1)

- Chiffrement symétrique
  - 1 seule clé K pour chiffrer et déchiffrer
  - Ex : (3-)DES, AES, Blowfish, RC4, A5/1…

Chiffrement par bloc

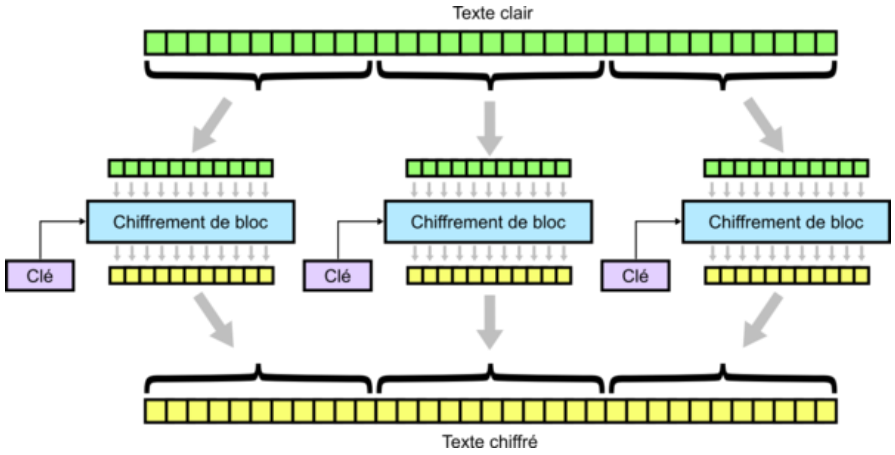


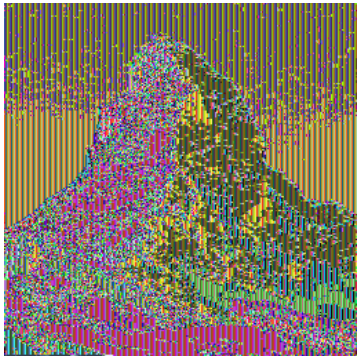
Chiffrement par flot



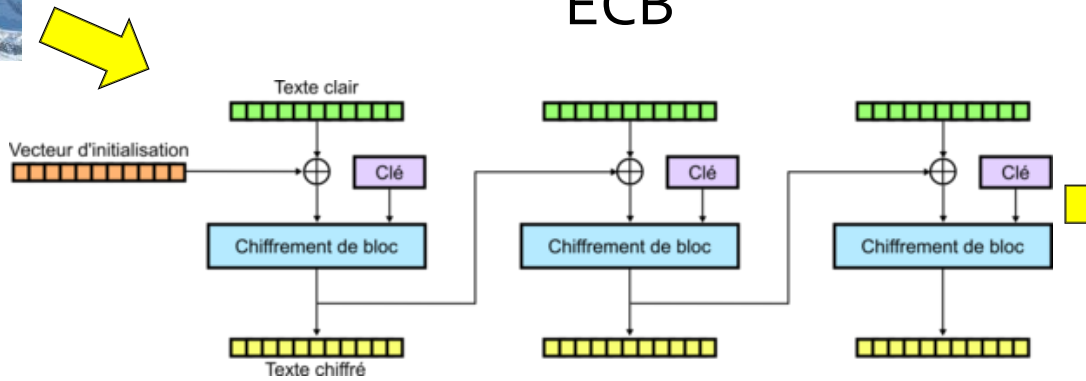
Introduction (2)

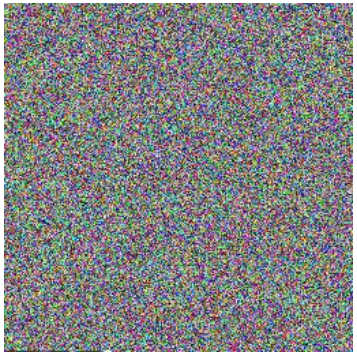
- Chiffrement par bloc : modes d'opération





ECB





CBC

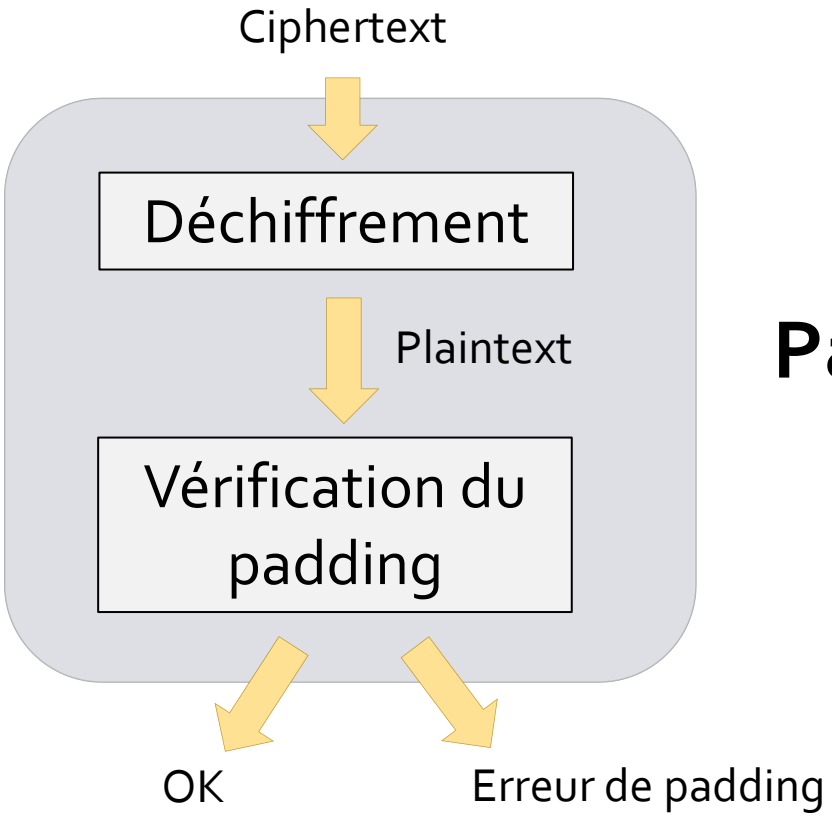
Introduction (3)

- Padding
  - Et si la taille du plaintext |M| est inférieure à un multiple de la taille B d'un bloc ?

  ➔ On rajoute une série d'octets de *padding*
  - Plusieurs façons de faire, dont **PKCS#5**
    - 01 s'il manque 1 octet
    - 02 02 s'il manque 2 octets
    - …
    - 08 08 08 08 08 08 08 08 si |M| est un multiple de B (ici B = 8)
  - Le padding est retiré après déchiffrement

Padding Oracle : Définition

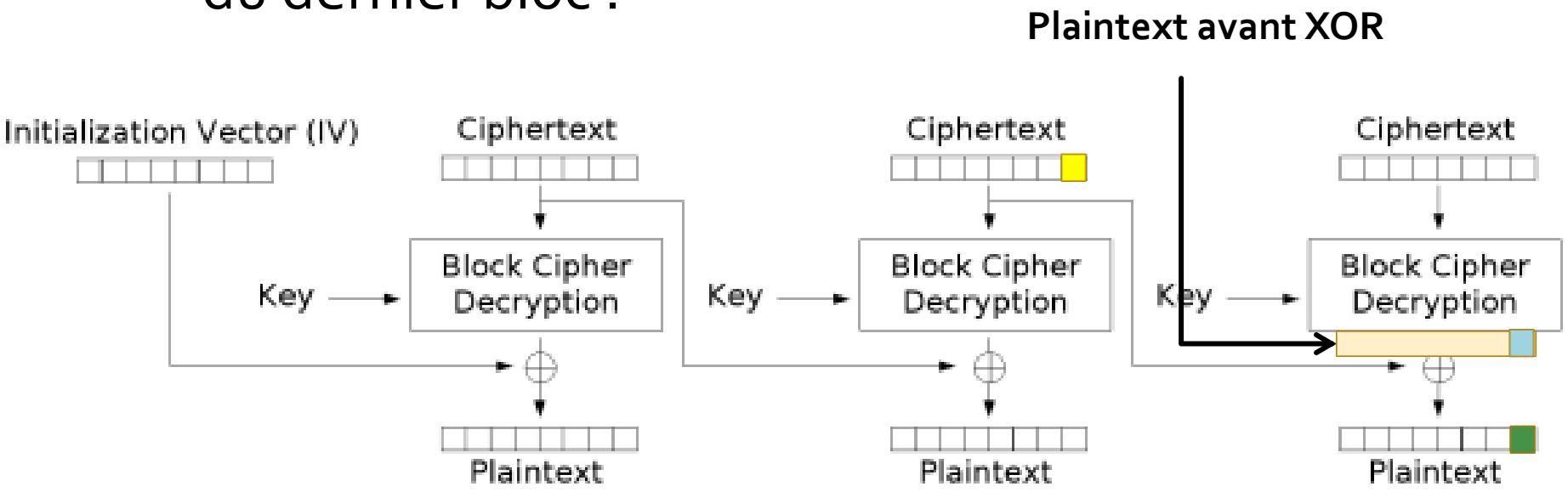
- Programme permettant de savoir si le padding après déchiffrement est correct



Padding Oracle

Padding Oracle : Idée

- Exploitation du mode CBC
  - L'avant-dernier bloc chiffré influe sur le padding du dernier bloc !



Cipher Block Chaining (CBC) mode decryption

Padding Oracle : Exploitation

- 3 étapes
  - Déchiffrement du dernier octet
  - Déchiffrement d'un bloc
  - Déchiffrement de la totalité du ciphertext

Security Flaws Induced by CBC Padding
Applications to SSL, IPSEC, WTLS...

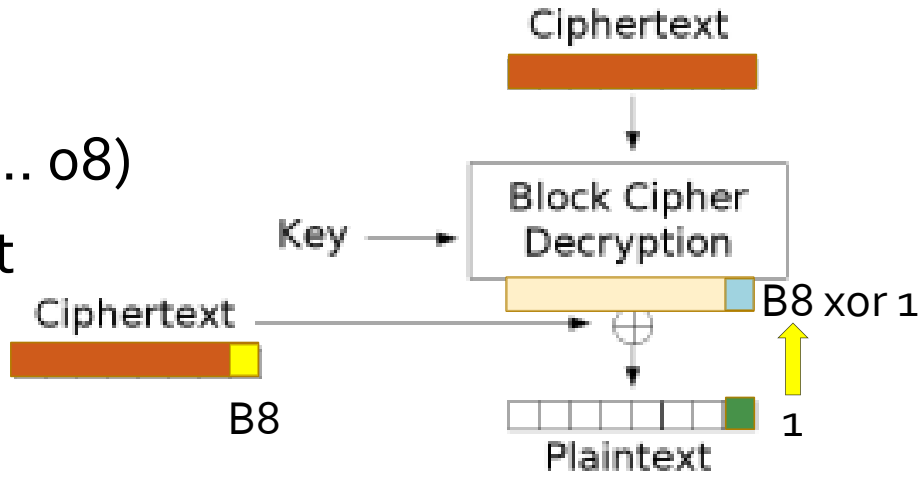
Serge Vaudenay

Swiss Federal Institute of Technology (EPFL)
Serge.Vaudenay@epfl.ch

Déchiffrement du dernier octet

- Forger un faux ciphertext composé de 2 blocs
  - 1er bloc aléatoire B
  - 2ème bloc issu d'un ciphertext C à déchiffrer
- Bruteforcer le dernier octet B8 du 1er bloc
- Réponse de l'oracle ?
  - Vrai → padding correct
    - 01 ou (02 02) ou … ou (08 … 08)

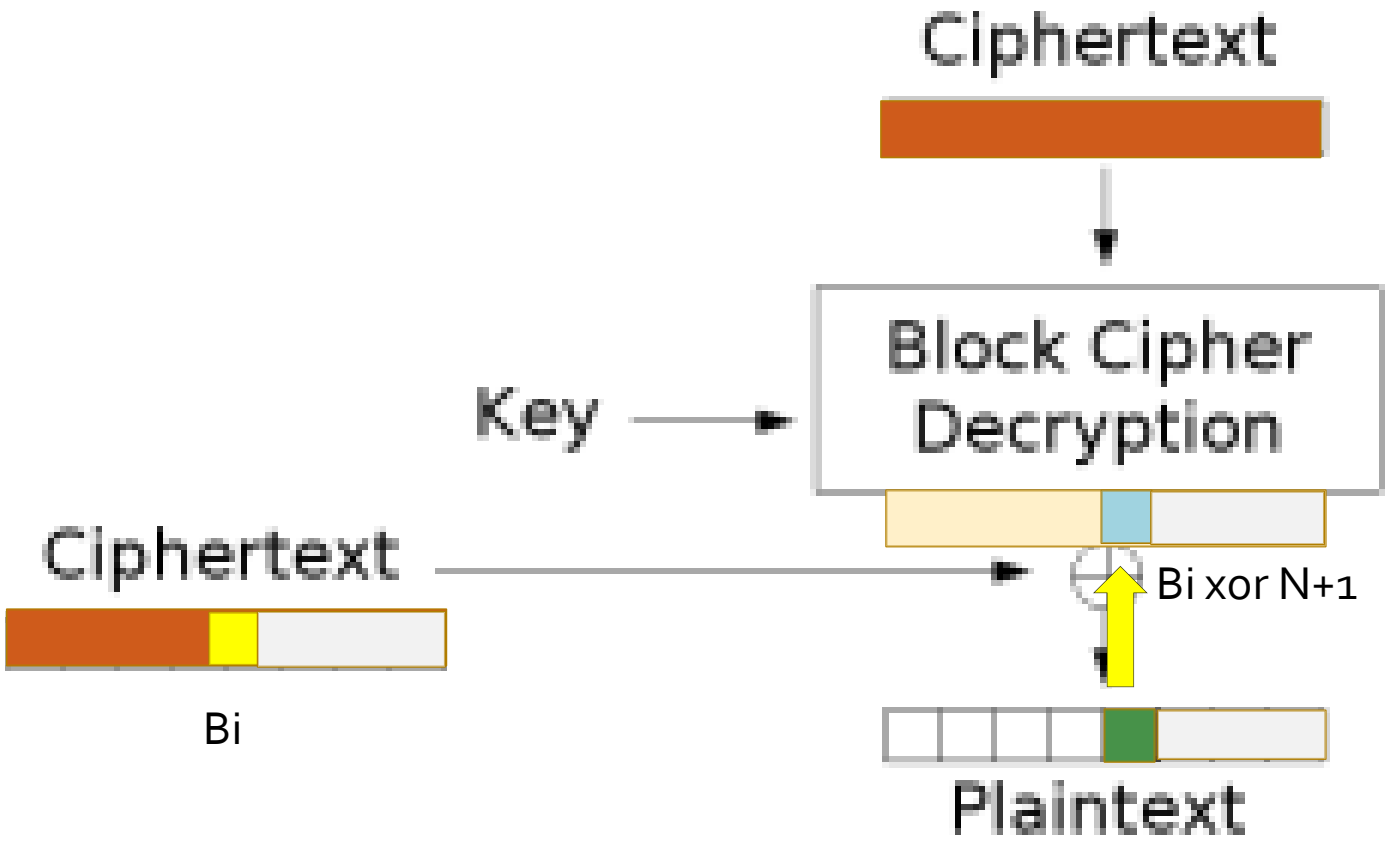
    → Déchiffrement d'un octet



Déchiffrement d'un bloc

- Algorithme récursif
  - Suppose que l'on a déjà les N derniers octets du plaintext (on part de la fin)
  - Comment déchiffrer l'octet précédent ?
- Forger un ciphertext composé de 2 blocs
  - 1er bloc basé sur le bloc aléatoire précédent
    - On souhaite que le padding du bloc déchiffré résultant corresponde à N+1 octets de valeur (N+1)
  - 2ème bloc issu du ciphertext à déchiffrer

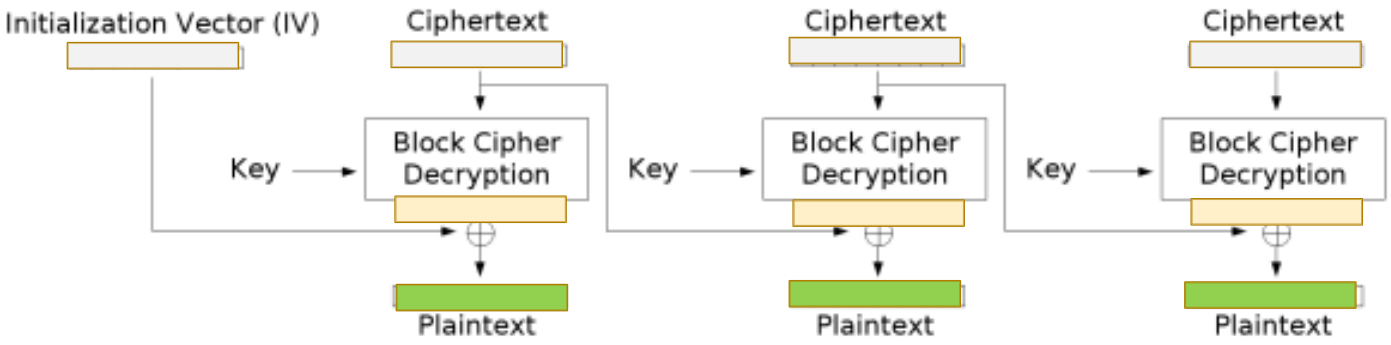
Déchiffrement d'un bloc



11

Déchiffrement du ciphertext

- On applique les algorithmes précédents pour chaque bloc
  - On obtient les plaintexts intermédiaires
- On utilise le mode CBC pour obtenir le plaintext original



Cipher Block Chaining (CBC) mode decryption

Padding Oracle : Conslusion

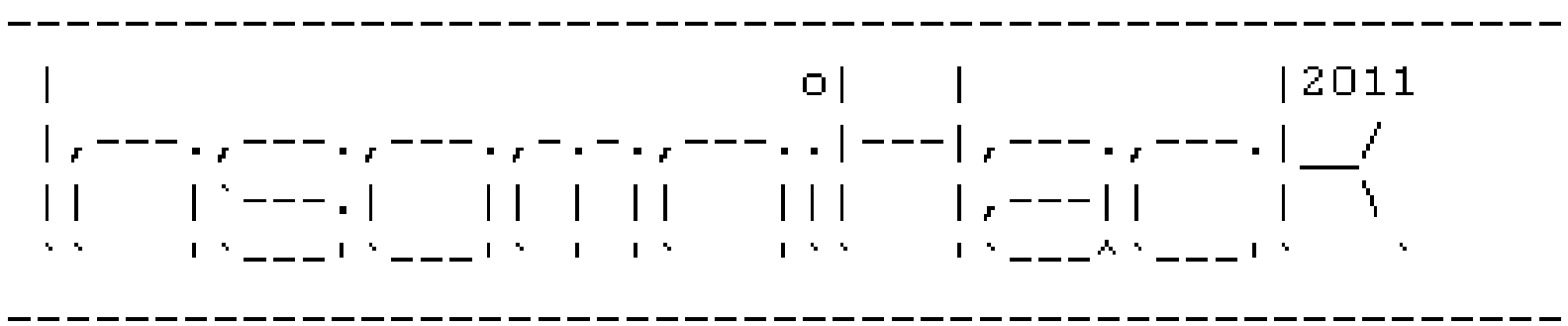
- On est en mesure de déchiffrer le ciphertext
  - Bruteforce intelligent, octet par octet
- Condition nécessaires
  - Connaître la taille du bloc, ou la deviner (8 ?)
  - Mode CBC
  - Affichage d'une erreur de padding
- Il n'est pas utile de connaître…
  - La clé
  - L'algorithme de chiffrement par bloc utilisé

Insomni'hack 2011 Crypto 200

- Challenge organisé à Genève
- Pas d'accès Internet, et pas de documentation…
- Epreuve Crypto 200 : 2 pages Web
  - L'une permet de chiffrer un plaintext quelconque
  - L'autre permet de vérifier le succès du déchiffrement d'un ciphertext quelconque
    - Sans pour autant afficher le résultat du déchiffrement ;-)
- Une indication : utilisation de 3DES-CBC
  - Clé inconnue, of course
- But : déchiffrer un ciphertext donné

Insomni'hack 2011 Crypto 200

- Démo



```
3DES(texte):     3F3FCA8CF9F6FF12A67635396 5A00ECECB2D3485
Résultat:        Déchiffrement échoué! [Padding Error]
```

Insomni'hack 2011 Crypto 200

- Résolution ?
  - La page 1 est inutile
  - Utiliser la page 2 en tant qu'oracle
  - Appliquer les 3 étapes précédentes
    - Automatiser en codant un exploit (en Python, of course!)

Insomni'hack 2011 Crypto 200

- Code pour interroger l'oracle :

```
URL_DEC="http://"+SERVEUR+"/dec/?t="
def oracle_dec(t):
    u = urllib2.urlopen(URL_DEC+str(t))
    r = u.read()
    u.close()
    return re.search("ussi", r) != None
```

- Quelques minutes de bruteforce…
- Résultat :

```
D:\Challenges\Insomnihack\2011\ctf\crypto\2>decrypt.py
107 100 14 189 34 129 105 16
45 183 29 222 187 198 199 73
186 149 253 197 187 249 250 132
122 234 55 178 99 117 171 142
C'est_bon_Peggy_je_gere!
```

Insomni'hack 2011 Crypto 200

- Démo et code de l'exploit

Crédit photo : http://bruno.kerouanton.net/blog/2011/03/05/insomnihack-2011-cest-fini/

Conclusion

- Padding Oracles
  - « verbosité » excessive de l'application
  - Attaque par texte chiffré choisi
  - Exploitation en 3 étapes
- Vulnérabilité de plus en plus fréquents
  - ASP.NET, JSP, …

Références

- *Security Flaws Induced by CBC Padding Applications to SSL, IPSEC, WTLS…*, S. Vaudenay (Eurocrypt 2002) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.3005&rep=rep1&type=pdf
- *Padding Oracles Everywhere*, T. Duong, J. Rizzo (Ekoparty 2010) http://www.ekoparty.org/archive/2010/ekoparty_2010-Duong_Rizzo-Padding_Oracles_Every_Where.pdf
- RFC 2898, Section 6.1.1 (Padding) http://tools.ietf.org/html/rfc2898#section-6.1.1
- Images : Wikipédia

Merci !

- Slides bientôt en ligne sur
  - www.segmentationfault.fr
  - www.hackerzvoice.net

Questions ?