TIW4 SÉCURITÉ DES SYSTÈMES D'INFORMATIONS
Contrôle Continu 1

Master 2 *Technologies de l'Information et Web (TIW)*
Promotion 2018 – 2019

Aucun document autorisé, sauf les dictionnaires multilingues. Les réponses doivent être données sur la feuille. Les réponses aux questions ouvertes doivent être claires et courtes.

Exercise 1 : Failles SSL du TP « HTTPS et authentification »

1. Décrire succinctement le problème de la faille Heartbleed (CVE-2014-0160).

2. Citer deux principales (selon vous) mesures mises en place pour contrer Heartbleed.

3. Le résumé de l'attaque POODLE (CVE-2014-3566) est donné en annexe A. Comment tester si votre serveur est vulnérable ?

4. Juger, en le motivant, si l'attaque POODLE est plus ou moins critique que Heartbleed.

5. On a donné plus haut les références CVE de Heartbleed et POODLE. Qu'est-ce-que une CVE ?

6. Expliquer ce qu'est l'outil `nmap`, ses principales fonctionnalités.

7. Expliquer ce qu'est l'outil Metasploit, ses principales fonctionnalités.

Exercise 2 : Sécurisation du serveur

1. Vous avez changé un certificat lors de la sécurisation de votre serveur. Pourquoi ?

2. Donner les principales étapes de la mise en place du nouveau certificat.

3. Dans les « Recommandations de sécurité relatives à TLS » de l'ANSSI, on lit *R6 : Échanger les clés en assurant toujours la PFS*. PFS est l'acronyme de *Perfect Forward Secrecy*. Expliquer.

4. Dans les suites cryptographiques il est question de « Diffie–Hellman éphémère » (acronyme DHE). Expliquer de quoi il s'agit.

5. Une des suites cryptographique supportée par Apache est la suivante, l'expliquer.
`TLS_DHE_RSA_WITH_AES_128_CBC_SHA256`.

6. Recommanderiez-vous *systématiquement* une mise-à-jour `apt-get dist-upgrade` d'un serveur de production ? Justifier.

7. Commenter « *When setting up any new system, Step 1 : Change default admin password.* » (*Security 101 : Security Basics in 140 Characters Or Less*).

Exercise 3 : Application PhP d'authentification

1. Donner trois mesures de sécurité que vous recommanderiez pour le service MySQL.

2. Un extrait de la documentation de `password_hash` est donné en annexe B. Vous exécutez `<?php echo password_hash("secret", PASSWORD_DEFAULT); ?>`. Le résultat produit est de la forme `$2y$1cdf2$QjSH496pcT5Cp3lRtee9HDxY3K`. Expliquer sa structure.

3. Expliquer ce qu'est le sel (*salt*) et son intérêt.

4. Justifier le choix de `bcrypt` pour `PASSWORD_DEFAULT`. Pourquoi ne pas prendre un algorithme comme SHA-2 ?

5. Le fonction jumelle de `password_hash` est `bool password_verify`. Donner son prototype et expliquer son fonctionnement.

6. On dit que `bcrypt` est résistant à l'attaque à « la seconde pré-image ». Expliquer ce qu'est cette attaque et les conséquences d'une faiblesse.

7. Un processus classique de récupération de mot de passe oublié se fait en plusieurs étapes. Proposer un tel processus.

A Extrait de la description de l'attaque POODLE

On donne un extrait de `https://www.openssl.org/~bodo/ssl-poodle.pdf`

SSL 3.0 [RFC6101] is an obsolete and insecure protocol. While for most practical purposes it has been replaced by its successors TLS 1.0 [RFC2246], TLS 1.1 [RFC4346], and TLS 1.2 [RFC5246], many TLS implementations remain backwardscompatible with SSL 3.0 to interoperate with legacy systems in the interest of a smooth user experience.

The protocol handshake provides for authenticated version negotiation, so normally the latest protocol version common to the client and the server will be used. However, even if a client and server both support a version of TLS, the security level offered by SSL 3.0 is still relevant since many clients implement a protocol downgrade dance to work around serverside interoperability bugs.

In this Security Advisory, we discuss how attackers can exploit the downgrade dance and break the cryptographic security of SSL 3.0. Our POODLE attack (Padding Oracle On Downgraded Legacy Encryption) will allow them, for example, to steal "secure" HTTP cookies (or other bearer tokens such as HTTP Authorization header contents).

We then give recommendations for both clients and servers on how to counter the attack : if disabling SSL 3.0 entirely is not acceptable out of interoperability concerns, TLS implementations should make use of `TLS_FALLBACK_SCSV`. CVE2014-3566 has been allocated for this protocol vulnerability.

B Extrait de la documentation `password_hash` de PhP

On donne un extrait de `http://php.net/manual/en/function.password-hash.php`

```
string password_hash (string $password, int $algo [, array $options ])
```

`password_hash()` creates a new password hash using a strong one-way hashing algorithm. `password_hash()` is compatible with `crypt()`. Therefore, password hashes created by `crypt()` can be used with `password_hash()`.

The following algorithms are currently supported :

`PASSWORD_DEFAULT` - Use the `bcrypt` algorithm (default as of PHP 5.5.0). [...]

On donne aussi l'extrait de `https://en.wikipedia.org/wiki/Bcrypt`

`bcrypt` is a password hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999.[1] Besides incorporating a salt [...], `bcrypt` is an adaptive function : over time, the iteration count can be increased to make it slower [...].