Commencé le	Monday 15 April 2024, 09:00
État	Terminé
Terminé le	Monday 15 April 2024, 09:24
Temps mis	24 min 15 s
Note	**12,00** sur 20,00 (**60**%)

Question **1**

Incorrect

Note de 0,00 sur 2,00

According to the ars technica write-up on https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/ (https://moodle-exam.cri.epita.fr/pluginfile.php/21021/question/questiontext/152016/1/70240/What%20we%20know%20about%20the%20xz%20Utils%20ba which of the following assertions are correct ?

D'après l'article d'ars-technica au sujet de la faille
lesquelles des affirmations sont correctes ?

- [x] a. The entirety of the backdoor was never committed to github, the final piece of the puzzle was only ever added to the distribution source tarballs.

  La totalité de la porte dérobée n'a jamais été commit sur github, le dernier bout du puzzle a seulement été ajouté aux tarballs sources de distribution de xz.

- [ ] b. The backdoor only targets some 64 bit linux distributions: precisely those that use rpm or deb packages, and that link openssh with libsystemd.

  La porte dérobée ne cible que les distributions linux 64 bits qui utilisent les packages rpm ou deb, et qui ont une liaison dynamique entre openssh et libsystemd.

- [ ] c. The backdoor exploits a bug in the systemd / openssh combo

  La porte dérobée utilise un bug dans la combinaison systemd / openssh

- [ ] d. By default, openssh is always vulnerable to this kind of exploit, since it requires liblzma to function properly.

  Par défaut, openssh est vulnérable à ce type d'exploit car il a besoin de liblzma pour certaines fonctionnalités.

Votre réponse est incorrecte.

Les réponses correctes sont :
The backdoor only targets some 64 bit linux distributions: precisely those that use rpm or deb packages, and that link openssh with libsystemd.

La porte dérobée ne cible que les distributions linux 64 bits qui utilisent les packages rpm ou deb, et qui ont une liaison dynamique entre openssh et libsystemd.,

The entirety of the backdoor was never committed to github, the final piece of the puzzle was only ever added to the distribution source tarballs.

La totalité de la porte dérobée n'a jamais été commit sur github, le dernier bout du puzzle a seulement été ajouté aux tarballs sources de distribution de xz.

Question **2**

Correct

Note de 2,00 sur 2,00

CVE:

a. A database of known attacks for penetration testers.

Une base de données d'attaques pour les consultants qui font des tests de pénétration.

b. A centralized database of security issues, independently of the operating system, that associates a unique identifier to each problem. ✔

Une base de donnée centralisée de problèmes de sécurité, indépendante de l'OS, qui associe un identifiant unique à chaque problème.

c. A specialized CI (continuous integration) system based on cvs (an ancestor of git) oriented towards bug fixes: every known security issue corresponds to a number test in that system.

Un système d'intégration continu basé sur cvs (un vieil équivalent de git) spécialisé dans les tests de correction de bug: chaque problème de sécurité connu fait l'objet d'un test numéroté dans ce système.

Votre réponse est correcte.

La réponse correcte est :
A centralized database of security issues, independently of the operating system, that associates a unique identifier to each problem.

Une base de donnée centralisée de problèmes de sécurité, indépendante de l'OS, qui associe un identifiant unique à chaque problème.

Question **3**

Correct

Note de 2,00 sur 2,00

Contrary to one may think, an attacker doesn't need full detailed knowledge of a system to craft an exploit. Understanding the weak links is enough.

Contrairement aux idées reçues, un attaquant n'a pas besoin de connaître en totalité un système pour le pirater. Il suffit de connaitre les bonnes failles.

Veuillez choisir une réponse.

- Vrai ✔
- Faux

La réponse correcte est « Vrai ».

Question **4**

Incorrect

Note de 0,00 sur 2,00

never use **printf(3)** with a format string provided by an attacker.

Ne jamais utiliser **printf(3)** avec une chaine de format fournie par un assaillant.

- [ ] a. That format string can even be used to write to memory, using the correct format specifier.

  Cette chaine de format permet même d'écrire en mémoire, en utilisant le bon spécificateur de format.

- [x] b. That format string can be used to display "extra" parameters, which will often be sensitive information on the stack, including memory addresses (leading to address leaks that can be the missing ingredient to exploiting a buffer overflow, for instance) ✔

  Cette chaine de format pourra être utilisée pour afficher des paramètres en plus, qui peuvent être des informations confidentielles sur la piste, en particulier des adresses mémoire (la fuite d'adresse étant souvent un prérequis à l'utilisation d'un débordement de tampon par exemple).

Votre réponse est incorrecte.

Les réponses correctes sont :
That format string can be used to display "extra" parameters, which will often be sensitive information on the stack, including memory addresses (leading to address leaks that can be the missing ingredient to exploiting a buffer overflow, for instance)

Cette chaine de format pourra être utilisée pour afficher des paramètres en plus, qui peuvent être des informations confidentielles sur la piste, en particulier des adresses mémoire (la fuite d'adresse étant souvent un prérequis à l'utilisation d'un débordement de tampon par exemple).,

That format string can even be used to write to memory, using the correct format specifier.

Cette chaine de format permet même d'écrire en mémoire, en utilisant le bon spécificateur de format.

Question **5**

Incorrect

Note de 0,00 sur 2,00

One weakness of **fork(2)** is that it duplicates a process's address space, thus minimizing the inherent advantages of address space randomization and making guessing at canary values easier.
Thus leading to the common practice of re-**exec(2)**: recreating a daemon from scratch every 5 minutes or so by **exec(2)**'ing the binary again, thus leading to a new randomized space.

Une faiblesse de **fork(2)**, c'est que le processus enfant est une exacte copie du processus parent: même espace mémoire, mêmes valeurs de canaris, ce qui réduit fortement les avantages de la *randomisation.*
D'où la pratique fréquente du re-**exec(2)**: relancer un démon *ab nihilo* toutes les 5 minutes en **exec(2)**utant à nouveau le binaire, et donc avec de nouvelles valeurs aléatoires.

Veuillez choisir une réponse.

- ○ Vrai
- ◉ Faux ✖

La réponse correcte est « Vrai ».

Question **6**

Incorrect

Note de 0,00 sur 2,00

One simple way to protect against SQL injections is to protect your tables against writing for non administrative users

Un moyen simple de prévenir les injections SQL, c'est de protéger vos tables en écriture pour les non administrateurs.

Veuillez choisir une réponse.

- Vrai ✖
- Faux

La réponse correcte est « Faux ».

Question **7**

Correct

Note de 2,00 sur 2,00

Consider the following function on a modern system / Considérez la fonction suivante sur un système moderne

```
1  #define LINESIZE 70
2
3
4  char *
5  user_input()
6  {
7          char buffer[LINESIZE];
8
9          gets(buffer);
10
11         return strdup(buffer);
12 }
13
```

By default, the compiler will insert a canari on the stack, so that even a 1 byte buffer overflow will be caught at run-time

Par défaut, le compilateur va insérer un canari sur la pile, de sorte que tout débordement de tampon, ne serait-ce que d'un octet, sera détecté à l'exécution.

Veuillez choisir une réponse.

- Vrai
- Faux ✔

La réponse correcte est « Faux ».

Question **8**

Correct

Note de 2,00 sur 2,00

Buffer overflows on the heap can be detected using canaris, but this is expensive, unless you only check the canaris some of the time, like when freeing memory. Another technique is the so-called "guard pages" which does leave empty spaces in the virtual memory mapping.
But this won't catch every buffer overflow, as allocations won't always match the page boundary, this is more a debugging technique that, coupled with randomization, will help developers weed out buffer overflows.

Les débordements de tampon sur le tas peuvent également être détectés en utilisant des canaris, mais ça coûte cher, sauf si on ne vérifie les canari qu'au moment de libérer la mémoire. Une autre technique est celle des «pages de garde»: le fait de laisser des adresses vides dans la cartographie de la mémoire virtuelle. Mais cette technique n'est pas 100% efficace non plus, vu que toutes les allocations ne seront pas alignées sur des frontières de page. C'est plus une technique de debug qui, alliée à de la randomisation, aide les développeurs à trouver les débordements.

Veuillez choisir une réponse.

- Vrai ✔
- Faux

La réponse correcte est « Vrai ».

Question **9**

Correct

Note de 2,00 sur 2,00

The following snippet could be an excerpt from a service program starting as root.
L'extrait suivant pourrait provenir d'un programme de service tournant en tant que root.

```
1  ...
2  /* this starts as root */
3  /* open and read the config file */
4  int fd = open_config_file();
5  if (fd != -1)
6          err(1, "Couldn't open config");
7  if (!parse_config_file(fd))
8          err(1, "Couldn't read config");
9  close(fd);
10
11 /* find the service login creds */
12 struct passwd *creds = getpwnam(config->name);
13 if (!creds)
14         err(1, "Daemon %s doesn't exist", config->name);
15 /* change identity */
16 setuid(creds->pw_uid);
17 setgid(creds->pw_gid);
18 ...
```

One the lines is very wrong !
Une des lignes est très très fausse.

Réponse : 16 ✔

La réponse correcte est : 16

Question **10**

Correct

Note de 2,00 sur 2,00

Some bugs may seem very unlikely to lead to an exploit, but it's mostly a question of time: some attackers have developed incredibly sophisticated techniques that leads to some bugs that were once considered «impossible» to exploit to be reclassified as «critical, will soon lead to an exploit».

Certains bugs semblent impossibles a exploiter, mais c'est surtout une question de temps: certains attaquants ont développé des techniques incroyablement sophistiquées qui ont conduit à reclasser certains bugs qu'on pensait comme «impossibles à exploiter» en «critiques, exploit en vue».

Veuillez choisir une réponse.

- (●) Vrai ✔
- ( ) Faux

La réponse correcte est « Vrai ».

◄ Annonces

Aller à…