

Adatbiztonság ZH

2011. április 28.

1. Tegyük fel, hogy DES rejtjelezést használtunk. 64 bites üzenetblokkon belül a blokk végén 4 bites hibadetekciós ellenőrzőösszeget alkalmazunk. Kimerítő kulcskereséses támadást végzünk.

12 rejtjeles blokk megfigyelése elegendő lenne-e a gyakorlatilag egyértelmű kulcsazonosításhoz? (azaz, hogy téves kulcsok száma várható értéke a kulcstér teljes végigkeresése után 1 körüli legyen) (8 pont)

2. Tekintsük egy blokk méretű m üzenet RSA aláírását, azaz $Sig(m) = m^d \pmod{n}$ transzformációt, ahol d a titkos kitevő és n a modulus. Megengedjük, hogy egy Z támadó kéréseket küldhessen az aláíró orákulumnak, majd generálnia kell egy aláírást egy új üzenetre.

Konstruáljon támadást. (Segítség: $(ab)^i \equiv a^i b^i$) (10 pont)

3. Az AWP (Another Weak Protocol) célja a kommunikáció védelme két távoli fél között, pontosabban az átküldött üzenetek titkosítása, integritásvédelme, és a visszajátszás elleni védelem. Feltesszük, hogy a felek között már van egy megosztott 128 bites K szimmetrikus kapcsolatkulcs. Egy M üzenet integritásvédő ellenőrzőösszegét úgy számoljuk ki, hogy M -et 128 bites M_i ($i = 1, 2, \dots$) blokkokra osztjuk, ha az utolsó blokk, M_{last} rövidebb 128 bitnél, akkor 0 bitekkel 128 bitre egészítjük ki, majd az így kapott blokkok (bitenkénti) XOR összegét számolva kapjuk az ellenőrzőösszeget: $ICV = M_1 + M_2 + \dots + M_{last}$ (ahol $+$ jelöli az XOR-t). Ezután az ICV-t az eredeti üzenet végére csatoljuk, és az $(M \parallel ICV)$ bitsorozatot rejtjelezzük AES rejtjelezővel CTR módban. A rejtjelezésnél használt számláló kezdeti értéke C , és a számlálót minden blokk rejtjelezése után eggyel növeljük. A kezdő C értéket a csomag fejlécében küldjük át a vevőnek, így az átküldött csomag formátuma a következő: $C \parallel AES-CTR_{K,C}(M \parallel ICV)$, ahol $AES-CTR_{K,C}()$ jelöli az AES-sel történő CTR módú rejtjelezést K kulccsal és C kezdeti számláló értékkel. A kapcsolat kezdetén (az első üzenet küldésekor) C -t 0-ról indítjuk, majd minden üzenet küldésekor eggyel növeljük. Így C üzenetsorszámként is funkcionál, s ez biztosítja a visszajátszott üzenetek detektálhatóságát. Új kapcsolat esetén új K kapcsolatkulcsot használunk, s C -t ismét 0-ról indítjuk.

Konstruáljon támadást az AWP protokoll integritásvédelmi mechanizmusa ellen és titkosítási eljárása ellen (15 pont)!

4. *Melyik biztonságosabb? Számítással indokoljon!* (6 pont)

i) (a) egy 6 számjegyből álló véletlen PIN kód vagy (b) egy 4 alfanumerikus (case sensitive) karakterből álló véletlen jelszó?

ii) (a) egy 6 számjegyből álló véletlen PIN kód vagy (b) egy 8 karakterből álló felhasználó által választott jelszó?

5. Unix/Linux hozzáférésvédelem

Tekintsük az alábbi /etc/passwd file részletet:

```
u1:x:1003:1004:,,,:/home/u1:/bin/bash
u2:x:1004:1005:,,,:/home/u2:/bin/bash
u3:x:1005:1006:,,,:/home/u3:/bin/bash
u4:x:1006:1007:,,,:/home/u4:/bin/bash
```

Az /etc/group file releváns része:

```
u1:x:1004:
u2:x:1005:
u3:x:1006:
u4:x:1007:
g1:x:1008:u1,u2
g2:x:1009:u2,u3,u4
g3:x:1010:u2,u3
```

A fájl hozzáférési jogosultságok az alábbiak:

```
root@gotcha:/adatbizt# ls -la
total 16
drwxr-xr-x  4 root root 4096 2011-04-22 10:49 .
drwxr-xr-x 25 root root 4096 2011-04-22 10:51 ..
drwxrwsr-x  2 u1  g1  4096 2011-04-22 10:50 d1
drwxr-xr--  2 u2  g1  4096 2011-04-22 10:50 d2
root@gotcha:/adatbizt# ls -la d1
total 20
drwxrwsr-x 2 u1  g1  4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw----- 1 u1  root    4 2011-04-22 10:50 f1
-rw-rw-r-- 1 u1  g1    16 2011-04-22 10:50 f2
-rwxrwxrwx 1 u1  g2    8 2011-04-22 10:50 f3
root@gotcha:/adatbizt# ls -la d2
total 16
drwxr-xr-- 2 u2  g1  4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw-r--r-- 1 root g1    7 2011-04-22 10:50 f4
--w----- 1 root g1    6 2011-04-22 10:50 f5
```

- mely felhasználók tudják kitörölni a d1/f1 fájlt és miért? (rm d1/f1) (2 pont)*
- mely felhasználóknál fut le sikeresen a cp d2/f4 d1/f6 parancs? (2 pont)*
- az u1 felhasználó (u1 aktív csoporttal) készít egy új fájlt d1-ben (touch d1/fu1), milyen csoport lesz a tulajdonosa a létrejövő fájlnak (2 pont)*
- A root felhasználó mely fájlokat tudja törölni az d1 alkönyvtárban (2 pont)*
- Ki tudja végrehajtani sikeresen az f2 fájl olvasási jogának teljes törlését? (chmod a-r d1/f2) (2 pont)*

6. Adott az alábbi tűzfal szabályhalmaz:

Keressen példát a következő inkonzisztenciátípusokra, és válaszát röviden indokolja!

No.	Proto	Src	Dst	Decision
1	tcp	10.1.1.0/25	any	deny
2	udp	any	192.168.1.0/24	accept
3	tcp	10.1.1.128/25	any	deny
4	udp	172.16.1.0/24	192.168.1.0/24	deny
5	tcp	10.1.1.0/24	any	accept
6	udp	10.1.1.0/24	192.168.0.0/16	deny
7	udp	172.16.1.0/24	any	accept

a.) *Shadowing* (2 pont)

b.) *Generalization* (2 pont)

c.) *Correlation* (2 pont)

Pontozás: 1: 0-21, 2: 22-29, 3: 30-38, 4: 39-46, 5: 47-55

Ez az oldal direkt üres.

Adatbiztonság ZH megoldások
2011. április 28.

Név:

Neptun kód:

1. igen nem
magyarázat:

2.

3. Támadás:

4. Melyik biztonságosabb?

i.) a b ii.) a b

magyarázat:

i.)

ii.)

5.

a.)

b.)

c.)

d.)

e.)

6.

a.)

b.)

c.)

Adatbiztonság ZH megoldások

2011. április 28.

1. Nem.

Annak a valószínűsége, hogy egy téves kulccsal helyes paritásúra dekódolunk egy rejtett szöveg blokkot, 2^{-4} . Annak a valószínűsége, hogy 12 rejtjeles blokk mindegyikét helyes paritásúra dekódoljuk téves kulcs mellett 2^{-48} , tehát a nem kiszűrt téves kulcsok száma várható értéke a kulcstér teljes végigkeresése után $2^{56} \cdot 2^{-48} = 2^8 = 256$ lenne.

2. Z kettő kérést küld az aláíró orákulumnak: $m_1=2$, $m_2=2m$, ahol m egy új üzenet.

$$\text{Sig}(2m) = \text{Sig}(2) \cdot \text{Sig}(m) \pmod{n}$$

egyenlőség alapján

$$\text{Sig}(m) = \text{Sig}(2)^{-1} \cdot \text{Sig}(2m) \pmod{n}$$

($\text{Sig}(2)$ invertálható \pmod{n}), mivel 2 invertálható \pmod{n} , ugyanis $\text{l.n.k.o}(2,n)=1$)

3. Páros számú blokk azonos módosítását nem detektálja az integritásvédelmi mechanizmus ($M_1+X+M_2+X+M_3+\dots+M_{\text{last}} = M_1+M_2+M_3+\dots+M_{\text{last}}$).

Továbbá a blokkokon könnyű célzott módosítást végezni a kulcsfolyam-rejtjelezés miatt ($M+Q+M' = (M+M')+Q$, ahol Q az AES-CTR által generált kulcsfolyam).

Két egymást követő üzenet kódolásához használt két kulcsfolyam nagymértékben átfed (azonos). Pl. a C . üzenethez használt kulcsfolyam: $\text{AES}_K(C)$, $\text{AES}_K(C+1)$, $\text{AES}_K(C+2)$, ... és a $C+1$. üzenethez használt kulcsfolyam: $\text{AES}_K(C+1)$, $\text{AES}_K(C+2)$, ... Ezért az első kódolt üzenet $i+1$. és a második kódolt üzenet i . blokkját XOR-olva, a nyílt üzenetek megfelelő blokkjainak XOR összegét kapjuk: $Y_{i+1} + Y'_i = (M_{i+1} + \text{AES}_K(C+i)) + (M'_i + \text{AES}_K(C+i)) = M_{i+1} + M'_i$. Ha ismerjük az egyik üzenetet, akkor ebből ki tudjuk számítani a másikat.

4.

i) (a) $6 \cdot \log_{10} 6 = 6 \cdot 3.322 = 19.932$ bit (b) $4 \cdot \log_2 62 = 4 \cdot 5.954 = 23.816$ bit

ii) (a) 19.932 bit (b) $4 + 7 \cdot 2 = 18$ bit

5.

- a.) u_1 és u_2 , mert u_1 és g_1 csoport tagjai írhatják az alkönyvtárat
- b.) d_1 -be u_1 és a g_1 írhat, tehát csak u_1 és u_2 jön szóba, ők mindketten hozzáférnek a f_4 fájlhoz, tehát u_1 és u_2 .
- c.) g_1 lesz, mert csoport setgid jel van az alkönyvtáron
- d.) Az összeset, mert a root felhasználó speciális jogú
- e.) u_1 , ő a tulajdonosa (és a root)

6.

- a.) pl. 4-es szabályt árnyékolja a 2-es
- b.) pl. 7-es a 4-est
- c.) pl. 2-es a 6-ossal