SRS - MI - 2021./2022. - Rješenja

UPUTE

Kod kratkih pitanja potrebno je sažeto odgovoriti na pitanje ili zaokružiti ispravna rješenja. Kod zadataka na zaokruživanje koji imaju više ispravnih rješenja potrebno je odabrati sva ispravna rješenja (i samo ispravna rješenja) kako bi dobili bodove. Kod problemskih zadataka potrebno je detaljno odgovoriti na postavljena pitanja. Zadatke rješavate isključivo u prostoru predviđenom za odgovore u samom ispitu koristeći poleđinu papira kada je potrebno. Neispravna rješenja ne nose negativne bodove.

KRATKA PITANJA

1. Osnovni pojmovi [6 bodova]

a) Narušavanje nekog od sigurnosnih zahtjeva nazivamo: _***incident***_

b) Koji sigurnosni zahtjev je narušen ako napadač snima komunikaciju između korisnika i Web sjedišta www.hr: _***tajnost***_

c) Koji sigurnosni zahtjev je narušen ako napadač snimi komunikaciju (bez da gleda njen sadržaj) te ju ponovi: _***autentičnost***_

d) Kako nazivamo očekivani gubitak koji je posljedica prijetnje, vjerojatnosti ostvarenja prijetnje, štete i ranjivosti: _***rizik***_

e) Kako se zove komunikacijski kanal kroz koji cure informacije koje napadač može upotrijebiti za neki svoj cilj: _***sporedni kanal (side channel)***_

f) Što iskorištava prijetnja: _***ranjivost***_

2. Kriptografija - hibridna enkripcija [4 boda]

a) Navedite jedan razlog za kombiniranje simetrične i asimetrične šifre.

***Asimetrična šifra nije siguran sustav kriptiranja pa ju je bolje kombinirati sa simetričnom šifrom (tad je sigurna).***

b) Opišite jedan siguran način kombiniranja kripto sustava „obični RSA“ i kripto sustava AES.

***Digitalna omotnica ili kriptiranje materijala za ključ (postupak za jedno od tog dvoje).***

3. Jednokratna bilježnica [2 boda]

Navedite jednu prednost i dva nedostatka jednokratne bilježnice.

***Prednost: savršeno povjerljiva***

***Nedostaci: ključ mora biti jednake duljine kao poruka i ključevi se moraju mijenjati***

By: CroMedo7, filzor, Svibanjka & 0ct0hus

4. Ranjivost i prijetnje [5 bodova]

Za svaku od sljedećih izjava navedite radi li se o prijetnji ili ranjivosti:

a) Napad pogađanja grubom silom: ***_prijetnja_***

b) Višestruka upotreba iste lozinke: ***_ranjivost_***

c) Krađa podataka s čvrstog diska: ***_prijetnja_***

d) Pogađanje lozinki: ***_prijetnja_***

e) Spajanje računala na Internet: ***_ranjivost_***

5. Sigurnost programske podrške 1 [1 bod]

Napisali ste sljedeći kod u nekom programskom jeziku:

```
x = 3;
y = 'SRS';
z = x + y;
```

Ako se radi o slabo tipiziranom programskom jeziku, što će se ispisati / koja vrijednost će biti pohranjena u varijablu z?

a) Greška prilikom prevođenja

b) Greška tijekom izvođenja

c) Samo brojevna vrijednost obzirom da se koristi operacija zbrajanja

d) Varijabla z imat će vrijednost 3SRS

6. Sigurnost operacijskih sustava [2 boda]

Izlistavanjem datoteka vidite da datoteka srs_ispit.txt ima postavljene ovlasti rw-------, pri čemu je vlasnik datoteke korisnik 'predavač' a grupa 'srs-studenti'. Htjeli bi definirati da studenti i predavač mogu pregledavati i pisati ispit. Što trebate upisati kao argument naredbe chmod?

a) 777 srs_ispit.txt

b) 640 srs_ispit.txt

c) 544 srs_ispit.txt

d) 660 srs_ispit.txt

7. Kontrola pristupa [2 boda]

a) Kada proces učitava datoteku, tijekom postupka autorizacije subjekt je ***_proces_***, objekt je ***_datoteka_*** i operacija je ***_učitavanje_***.

b) Na koja dva faktora autentifikacije se temelje pametne kartice: ***_ono što imamo i ono što znamo_***

8. Zloćudni kod [2 boda]

Anti-virusni program je spriječio zarazu računala zloćudnim kodom. Analizom zloćudnog koda utvrđeno je da zloćudni kod nakon zaraze skenira lokalnu mrežu te se pokušava proširiti i na druga računala u lokalnoj mreži. Također je utvrđeno da kad uspješno zarazi neko računalo onda šifrira sve dokumente na disku te započinje komunikaciju s adresom 169.259.14.15. Odgovorite na sljedeća pitanja:

a) Kojom vrstom zloćudnog koda s obzirom na širenje je računalo zaposlenika zaraženo: _***crv***_

b) Kojom vrstom zloćudnog koda s obzirom na zloćudni teret je računalo zaposlenika zaraženo: _***ransomware***_

c) Kako se zove računalo s IP adresom 169.259.14.15: _***C&C poslužitelj***_

d) Koja grupa napadača (izvora prijetnji) najvjerojatnije stoji iza ovog napada: _***kibernetički kriminalci***_

9. Lozinke [1 bod]

Navedite poruku o grešci koju bi ispisali korisniku kada upiše nepostojeće korisničko ime:

***Username or password incorrect***

10. Sigurnost programske podrške 2 [2 boda]

Napisali ste kod:

```
1: ispitProlaz = true;
2: try {
3:     bodovi = provjeriBodove();
4:     if(bodovi < 50) {
5:           ispitProlaz = false;
6:     }
7: }
8: catch (Exception ex)
9: {
10:  // write error
11: }
```

Ako metoda provjeriBodove() može baciti grešku, što bi trebalo popraviti u kodu da bi ispravno radio i da zadovoljimo princip sigurnog Ispadanja (Fall Securely):

a) Trebali bi dodati uvjet *else { ispitProlaz true; }* nakon linije 6

b) Trebali bi izraz na linlji 1 zamijeniti sa *ispitProlaz = false;*

c) Trebali bi zamijeniti izraz na liniji 1 sa ispitProlaz = false ; i dodati uvjet else ispitProlaz = true; } nakon linije 6

d) Ne bi trebalo mijenjati ništa, ovako će raditi ispravno i bez grešaka

11. Sigurnost programske podrške 3 [1 bod]

Što je princip najmanjih prava (least privilege) prilikom izrade programskih rješenja?

a) Princip koji definira kako svim korisnicima i procesima treba dozvoliti najmanje potrebne ovlasti

b) Princip koji definira kako prilikom dizajna sustava ili programa zbog sigurnosti treba implementirati samo najbitnije funkcionalnosti

c) Princip koji definira kako je potrebno smanjiti broj nepotrebnih uloga u sustavu ili programu

d) Princip koji definira koja uloga korisnika treba imati najmanje prava

Problemski zadaci

12. Autentificirana šifra [6 bodova]

Zadan je sljedeći pseudo kod kojim se šifriraju kratke poruka fiksne duljine od 128 bitova s ciljem da je osigurana i povjerljivost i integritet komunikacije. Za šifriranje se koristi blok sifra AES128 i kriptografska funkcija SHA256.

Postupak šifriranja: Ulaz je poruka m duljine 128 bitova koju je potrebno šifrirati, te ključ k duljine 128 bitova.

1. r je niz od 128 bitova generiran kriptografskim generatorom slučajnih brojeva

2. c = AES128(r, k) XOR m

3. t = SHA256(r || c) gdje I I označava spajanje dva niza bitova

4. šifrat je trojka (r, c, t)

Djelomični postupak dešifriranja i provjere integriteta: Ulaz je šifrat (r, c, t), te ključ k duljine 128 bitova.

1. Provjeri je li t = SHA256(r || c), ako nije prijavi grešku

2. ...

Pitanja:

a) Dovršite opis postupka dešifriranja.

Dobij m = AES128(r, k) XOR c

- ***Nakon što provjerimo integritet poruke koristeći SHA256, možemo dovršiti postupak dešifriranja:***
- ***Provjeri je li t = SHA256(r || c), ako nije prijavi grešku***
- ***m' = AES128_dec(r, k) XOR c***
- ***Izlaz je dešifrirana poruka m'***

b) Pruža li ovakav postupak šifriranja svojstvo povjerljivosti? Ako da, iznesite zašto, ako ne opišite jedan scenarij u kojem je svojstvo povjerljivosti narušeno.

Da, postupak lici na CTR ali samo s jednim blokom poruke: CTR je praktički OTP s ključem koji se koristi u algoritmu AES koji je povjerljiv ako se mijenjaju ulazi, sto je ovdje ostvareno jer je r uvijek drugačiji.

Ovaj postupak šifriranja pruža svojstvo povjerljivosti jer koristi simetričnu kriptografiju (AES128) za šifriranje poruke: AES128 je siguran algoritam za šifriranje koji osigurava tajnost podataka kada se koristi ispravno: Povjerljivost se održava jer bez ključa k, treće strane ne mogu dešifrirati poruku c.

c) Pruža li ovakav postupak šifriranja svojstvo integriteta? Ako da, iznesite zašto, ako ne, opišite jedan scenarij u kojem je svojstvo integriteta narušeno.

Ne, jer je podložan malleable encryptionu: Ako napadač zna originalnu poruku m1 i presretne šifrat c1, može originalnu poruku zamijeniti s novom m2 tako da pošalje šifrat c2 = m1 XOR m2 XOR c1: Tada će izmijeniti i hash, no budući da se koristi SHA256 za koji, osim kriptirane poruke, treba r kojeg iščita iz presretnute poruke, izmjena hasha ce biti moguća.

Ovaj postupak šifriranja ima nedostatak u pogledu osiguravanja integriteta: Budući da se ključ ne koristi prilikom generiranja sažetka (t) pomoću SHA256, postoje scenariji u kojima integritet može biti narušen npr.:

Ako napadač zna stari m, može izračunati novi c tako da izvede operaciju XOR između c, stare poruke (m) i nove poruke: Nakon toga, napadač može izračunati novi t koristeći SHA256 bez potrebe za ključem: Na taj način, napadač bi mogao promijeniti šifriranu poruku bez otkrivanja.

13. Prelijevanje međuspremnika [6 bodova]

Dolje je zadan isječak koda alata koji provjerava administratorsku zaporku te, ako je ona ispravna, omogućuje korisniku unos daljnjih administratorskih naredbi. Cilj napadača je da se u programu izvrši funkcija login_success, a bez da napadač pogodi administratorsku zaporku. Alat se koristi na Linux x86-64 sustavu koji ima uključenu samo "Write-XOR-Execute" zaštitu. Napadač zna da se login_success fja nalazi na adresi 0x0000000040253c u memorijskom prostoru procesa.

```
void check_admin_password() {
	// Lokalne varljable
	char entered_password[16];
	char admin_password[16];

	// Pitaj korisnika da upiše administratorsku zaporku
	printf("Enter password: ");
	scanf("%s, entered_password);

	// Pročitaj pravu administratorsku zaporku iz datotek
	load_admin_password(admin_password);

	// Usporedi zaporke
	if (!strcmp(admin_password, entered_password))

		login success();
```

Funkcijski stog raste prema nižim adresama te su kasnije deklarirane lokalne varijable smještene na nižim adresama. Primjer izgleda funkcijskog stoga u slučaju kada je administratorska zaporka "pass", a korisnik je upisao zaporku "mrkva" je dan niže. Primijetite da se koristi 'little-endian' zapis pa tako niz znakova "mrkva" s ASCII vrijednostima znakova redom Ox6d, 0x72, Oxb6, 0x76, 0x61 odgovara vrijednosti 0x61766b726d na odgovarajućem mjestu u memoriji.

Memorijska adresa	Vrijednost	Komentar
0x00007fffffffd900	0x0000000073736170	prvih 8 bajtova od admin_password
0x00007fffffffd908	0x0000000000000000	drugih 8 bajtova od admin_password
0x00007fffffffd910	0x00000061766b726d	prvih 8 bajtova od entered_password
0x00007fffffffd918	0x0000000000000000	drugih 8 bajtova od entered_password
0x00007fffffffd920	0x00007fffffffd930	spremljeni stari $rbp
0x00007fffffffd928	0x00000000004012bb	adresa za povratak

a) Navedite jedan točan niz bajtova koji napadač može poslati kao zaporku kako bi se izvršila funkcija login_success te skicirajte izgled funkcijskog stoga netom prije povratka iz funkcije check_admin_password.

Kratki odgovor: 0x 24 * smece bajt 3c 25 40

Dugi odgovor:

Recimo da nam je smeće slovo A i da je adresa funkcije login_success 0x000000000040253c

napadač onda upisuje

AAAAAAAAAAAAAAAAAAAAAAAA3c 25 40

(bez novih redova i razmaka, ovi A su kao smeće varijabla, može bit i random slovo nije bitno, ali ne smiju bit redom nule jer se to gleda kao terminator i scanf prestaje čitat dalje sta si pisao)

Nakon upisa, stog nam izgleda ovako:

stog:

--niže adrese

A

A

...

A

3c

25

40

prazno

prazno

...

prazno

--više adrese

Prikaz stoga s 8 bajtova u svakom redu:

stog:

--nize adrese

AAAAAAAA

AAAAAAAA

AAAAAAAA

3c 25 40 prazno prazno prazno prazno prazno

--vise adrese

(jer gore su nize adrese, dolje vise, a također su lijevo nize adrese a desno vise)

Onda kad će učitavati stvari sa stoga u memoriju (konkretno adresu metode login_success), kad uzima tih 8 bajtova on će gledat aha meni je 40 najbitniji jer je na najvišoj adresi (kod little endiana je najmanje bitan dio na najnižoj adresi sto odgovara vrhu stoga, tj. nižim adresama na stogu) pa je adresa 0x40253c i onda će ubacit još nule ispred pa će kao vrijednost za adresu za povratak bit upisano 0x000000000040253c.

b) Objasnite kako se od napada preljeva međuspremnika možemo obraniti koristeći tzv. kanarince te detaljno obrazložite je li još uvijek moguć napad iz prvog dijela zadatka.

Kanarinac je neki nasumičan broj koje compiler umetne neposredno prije povratne adrese: Kada naleti na povratnu adresu, proces ce provjeriti kanarinca i ako se taj broj promijenio, znat će da nešto nije u redu i prekinut će se.

Napad iz prvog dijela zadatka nije moguć jer bi se definitivno izmijenio broj prije povratne adrese.

c) Objasnite na koji način bi promijenili izvorni kod tako da napad više nije moguć čak i ako se ne koriste kanarinci niti druge zaštite.

Ne bih koristila scanf ili bi npr. upisivala char po char pa ograničila petlju na 16 znakova.