KTH Datavetenskap
och kommunikation

DD2395 Datasäkerhet
Tentamen 2009-01-07 kl 09.00–13.00

Inga hjälpmedel är tillåtna. För att få godkänt på tentan måste man få godkänt på del I (G på minst 6 frågor). Tentabetyget ges av poängen på del II. Betygsgränserna är preliminärt D – 12p, C – 17p, B–22, A – 27p.

Del I (besvaras på separat blad, inlämnas efter två timmar)

1 Vad är en logisk bomb?

2 Vad innebär MAC och DAC (accesskontroll) och vilken typ är Bell LaPadula exempel på?

3 Vad gör en referensmonitor? Ge exempel på hur en OS-kärna kan agera referensmonitor.

4 Är (51,4), (51,13) ett giltigt RSA-nyckelpar? Verifiera eller motbevisa.

5 ”Public key infrastructure” (PKI) bygger på certifikat. Hur är ett certifikat uppbyggt, och hur verifieras det?

6 Förklara signaturbaserad intrångsdetektering. Vilken annan metod används ibland vid intrångsdetektering och vad innebär den?

7 Vad innebär ”non-repudiation”?

8 Beskriv accessrättigheterna i nedanstående matris som accesskontrollistor (ACL:er).

	foo.txt	bar.exe	messages.log
migo	rw	rx	r
pehrs	r		r
alba		rwx	rw

9 Vad kontrollerar ett spoofing-filter (en sorts brandvägg)?

Del II

10

10a Hur går en buffer overrun-attack mot stacken till? [3p]

10b Varför är det vanligare med denna typ av sårbarhet hos C-program än hos Java-program? [1p]

10c Ge exempel på något som minskar (C-)programs sårbarhet mot denna attack (inte genom att modifiera källkoden utan snarare egenskaper hos OS och/eller kompilator)? [1p]

11 Tänk dig att vi använder ett databassystem för att hantera resultaten på CSC-kurser och att du som student kan komma åt information om dina resultat via anropet
`showres dasak08`
Kommandot `showres` exekverar SUID db som är en speciell användare som har rätt att läsa i och ändra i databasen, vilket vanliga användare inte får göra. I programmet finns koden

```
query = "SELECT grade FROM exam WHERE user='"
         + uname + "' and course='" + cname + "';" ;
result = db_query(query); /* Utför frågan mot databasen */
display(result);
```

där `query`, `cname` och `uname` är strängvariabler. Värdet på `uname` får programmet från environment-variabeln USER och värdet på `cname` är kommandoradsargumentet (dasak08 i exemplet).

Vilka säkerhethål kan man misstänka finns här? Vad kan man göra för att täppa till dem? [4p]

12 Brandväggar baserade på iptables har så kallade kedjor (chains). Man kan definiera egna kedjor men tre finns fördefinierade. Vad heter de? Gör en enkel skiss över ett lokalt nätverk som har en brandvägg mot Internet. Förklara vilka paket som berörs av vilken kedja. [4p]

13 Hur går ”cross-site request forgery” till? Ge exempel på vad attacken kan användas till. [3p]

14 MAC (message authentication code) och digitala signaturer är två olika sätt att uppnå dataintegritet. Förklara hur respektive metod fungerar och vilka likheter och skillnader som finns. Antag att du har skrivit ett program som folk kan ladda ner. Om du vill att folk ska kunna kontrollera integriteten för den nerladdade filen, vilken av metoderna är då lämpligast och varför? [4p]

15 Anna och Bertil jobbar som jurister och använder modellen ”Kinesiska muren”.

- Byrån har sex företag som klienter och vi numrerar med siffrorna 1 till 6.
- För varje företag finns tre objekt: en affärsplan P och en intern budget B, samt en offentlig (”sanitized”) årsrapport R. Vi använder P_1, B_1 och R_1 för att beteckna klient 1s objekt, och så vidare.
- Vi har tre *intresse-konflikt-grupper*: $COI_1 = \{1, 2\}$, $COI_2 = \{3, 4\}$ och $COI_3 = \{5, 6\}$.

15a Anna är specialiserad på COI_1 och har inte läsrättighet till hemliga (unsanitized) objekt som tillhör företag i övriga COI. Bertil har dock från början läsrättigheter till allt. Anna och Bertil försöker nu utföra, i tur och ordning, följande operationer. Vilka tillåts och vilka tillåts inte? Förklara varför. [1p för 5 rätt, 2p för 6 rätt, 3p för 7 rätt, 4p för 8 rätt]

1. Anna försöker läsa från P_2
2. Anna försöker läsa från B_5
3. Bertil försöker läsa från P_3
4. Bertil försöker skriva till B_1
5. Anna försöker skriva till B_2
6. Bertil försöker läsa från B_4
7. Anna försöker läsa från R_5
8. Bertil försöker läsa från B_2

15b Kan man realisera denna modell med hjälp av en kombination av Biba och Bell LaPadula? Förklara hur man gör det eller förklara varför det är problematiskt. [1p]

16 Java är ett språk som stöder kodbaserad åtkomstkontroll (CBAC), vilket innebär att olika kod kan exekvera med olika rättigheter. Låt oss anta att vi har två klasser, `A` och `B`. Klassen `A` har metoden `log(String filename, String m)` som lägger till en rad med strängen `m` till filen `filename` (exempelvis genom att först läsa in hela filen, sedan öppna den för skrivning och skriva tillbaks allt det inlästa och lägga till strängen `m` och till sist stänga den).

Dessutom har `A` och `B` metoder som följer:

```
import java.io.*;

class A
{
    void log(String filename, String m) throws Exception
    {   /* gör det som beskrivs i texten ovan */    }

    void blog(Strin fn, String m) throws Exception
    { B b = new B(); b.alog(fn, m); }

    void alog(String fn, String m) throws Exception
    { A a = new A(); a.log(fn, m); }
}

class B extends A
{
    void alog(String fn, String m) throws Exception
    { A a = new A(); a.log(m);  }
}
```

Klasserna `A` och `B` är definierade i olika ”code bases”, som vi kallar `ABase` och `BBase` nedan. Med hjälp av en policyfil har `ABase` läs- och skrivrättigheter till filerna `log.txt` och `sec.txt`. Rättigheterna för `BBase` tillåter inte skrivning av filer alls.

En användare skriver ett program som har fulla rättigheter:

```
public class Test {
  public static void main(String[] args) {
    A a = new A();
    B b = new B();
    try {  a.log("log.txt", "log 1");  }  catch (Exception e){ };
    try {  b.alog("log.txt", "log 2"); }  catch (Exception e){ };
    try {  b.log("log.txt", "log 3");  }  catch (Exception e){ };
    try {  a.blog("log.txt", "log 4"); }  catch (Exception e){ };
    try {  a.log("sec.txt", "sec 5");  }  catch (Exception e){ };
    try {  b.alog("sec.txt", "sec 6"); }  catch (Exception e){ };
    try {  b.log("sec.txt", "sec 7");  }  catch (Exception e){ };
    try {  a.blog("sec.txt", "sec 8"); }  catch (Exception e){ };
  }
}
```

Användaren som startar programmet har rätt att läsa och skriva `log.txt`, men inte att läsa eller skriva `sec.txt`. Vad kommer det att stå i respektive fil efter att programmet körts en gång (motivera genom att förklara vilka för frågan relevanta kontroller som sker vid körningen)? [5p]