KTH Datavetenskap
och kommunikation

2D1395 Datasäkerhet
Tentamen 2006-10-18 kl 14.00–18.00

Inga hjälpmedel är tillåtna. Betygsgränserna är preliminärt 3 – 15p, 4 – 20p, 5 – 25p för KTH-studenter och G – 15p, VG – 23p för SU-studenter.

Enligt nya bestämmelser så erbjuds studenter som varit mycket nära gränsen för godkänt att komplettera till betyg 3/G på tentan. Detta sker i så fall genom muntlig eller skriftlig komplettering motsvarande ett område som man inte vid tentatillfället behärskat. Vid komplettering kan tentamensbetyget inte bli högre än 3/G.

1 Förklara innebörden av nedanstående begrepp och ge exempel på mekanismer och tekniker som används för att åstadkomma dem. [6p]

Konfidentialitet (confidentiality)
Integritet (integrity)
Spårbarhet (accountability)

2 En lösenordsfil i exempelvis Unix innehåller bland annat användarnamn och hashade lösenord istället för lösenorden i klartext. Om någon får tag i lösenordsfilen så ska det inte ge tillgång till lösenorden i klartext.

2a Vilka krav bör man ha på den hashfunktion som används? [1p]

2b Ibland används *saltning* av lösenorden. Vilken sorts attacker skyddar saltning mot, hur går en sådan attack till (utan salt och med salt), och hur mycket mer tidskrävande blir en sådan attack om man har en lösenordsfil med 1000 användare och använder 8 bytes med salt? [2p]

3 Min prenumeration på ett antivirusprogram gick ut för några månader sedan, men nu har jag förnyat den. Jag har nu kört den uppdaterade versionen och den har rensat bort ett antal spionprogram och virus. Vad bör jag oroa mig för när det gäller säkerheten på datorn efter uppgraderingen? Vilka ytterligare åtgärder kan jag vidta och vad har de för fördelar och nackdelar? [3p]

4 Foo AB har kommit på ett jättefiffigt system för att hantera inloggning på Unix-liknande system. Istället för en lösenordsfil så har de en katalog med en fil per användare där filnamnet har formatet *användarnamn.lösenord* (exempel följer). Detta gör att de slipper implementera hashfunktion och annat jobbigt. Katalogen kan bara listas av administratören (som då förvisso kan se allas lösenord, men så kan det gå med fiffiga lösningar).

Användarnamn består enbart av små bokstäver. Lösenord kan innehålla bokstäver (stora och små), siffror, samt ett 20-tal övriga tecken.

T.ex. kan katalogen se ut så här (men man måste vara administratör för att lista filerna):

```
host# ls /secret/accounts
alice.Am3hc94-"# bob.password532
```

För att autentisera användarnamn och lösenord används följande pythonkod:

```
def goodpass(username, password):
        upass = username + '.' + password
        # No whitespace allowed in username or password
        for c in string.whitespace:
                if c in upass:
                        return False
        # Verify username and password
        if(len(glob.glob("/secret/accounts/" + upass)) == 1):
                return True
        else:
                return False
```

Det du behöver veta om python är att `glob.glob` fungerar likadant som om du gör '`ls`' med samma argument och att `len(...) == 1` kontrollerar att man får tillbaka exakt ett resultat. Du kan anta att den första for-slingan som kontrollerar att det inte finns whitespace (mellanslag m.m) i strängarna fungerar.

4a Om vi känner till att alice är ett användarnamn på systemet, beskriv en effektiv attack för att lyckas logga in som alice. [1p]

4b Vi misstänker att bob använder samma lösenord på ett annat system. Beskriv en effektiv attack för att ta reda på hans lösenord. Det finns inget kommando när man väl är inloggad för att visa lösenord, så det räcker inte med att man lyckas att logga in som bob. [1p]

4c Vi känner inte till några användarnamn på systemet. Beskriv en rimligt effektiv attack för att lyckas logga in. [1p]

4d Vi är intresserade av hurvida en viss fil, `/etc/uucp/passwd`, finns. Som användare har man inte listrättighet till katalogen `/etc/uucp/`. Beskriv en attack som tar reda på om den filen finns på systemet. [1p]

5 En brandvägg kan användas för att förhindra intrång. Det finns även ett antal tekniker som används för att upptäcka en pågående attack eller för att man i ett senare skede ska kunna anlysera misstänkta aktiviteter.

5a Bland annat använder man intrångsdetekteringssystem (IDS). Beskriv ett par lämpliga datakällor för ett IDS att jobba med. [1p]

5b Vilka två huvudsakliga tekniker arbetar IDS med för att analysera insamlade data? [1p]

5c Ofta vill man ha två (eller flera) IDS uppe parallelt. Varför? [1p]

5d ”Honeypot” är ett kompletterande verktyg. Vad är en honeypot och varför har man en? [1p]

6 I det här problemet använder vi asymmetrisk kryptering och digitala signaturer som vi tänker oss fungerar ungefär som i GnuPG. Vi antar att man inte använder samma nycklar för kryptering och signering. Antag att *A* har *B*s publika nycklar och vice versa.

6a *A* vill skicka ett krypterat och signerat meddelande till *B*. Beskriv *A*s protokoll för att göra detta (dvs vilka moment som ingår och hur nycklar används). Beskriv på motsvarande sätt vad *B* ska göra för att kunna kontrollera integritet och läsa klartexten. [1p]

6b Elaka *E* har fått tag på ett antal meddelanden som *B* fått och som är krypterade och signerade genom att stjäla ett USB-minne där *B* förvarade dem. Dessa meddelanden är från ett antal olika parter som alla kommunicerat på samma sätt som *A* i uppgiften ovan. Vilka möjligheter har *E* att undersöka vem som skickat meddelanden till *B*? [1p]

6c Om *A* och *B* inte har varandras nycklar kan de börja med att e-posta dem till varandra för att sedan kommunicera krypterat via e-post. Förklara principerna för en attack som låter *E* läsa de krypterade breven utan att *A* och *B* upptäcker det. Vilka mekanismer och tekniker finns för att undvika detta problem? [2p]

7 Hur går en *buffer overrun*-attack mot stacken till? Ett motmedel är att använda en ”kanariefågel” (canary). Vad innebär det? Datorsystem erbjuder ofta olika former av minnesskydd. Ge exempel på ett sådant som skyddar mot denna attack. [3p]

8 Java är ett språk som stöder kodbaserad åtkomstkontroll (CBAC), vilket innebär att olika klasser kan exekvera med olika rättigheter. Låt oss anta att vi har två klasser, `A` och `B`. Klassen `A` har metoden `log(String m)` som lägger till en rad med strängen `m` till filen `events.log` (exempelvis genom att först läsa in hela filen `events.log`, 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 m) throws Exception
    {   /* gör det som beskrivs i texten ovan */    }

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

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

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

Med hjälp av en policyfil har `A` läs- och skrivrättigheter till `events.log`. Rättigheterna för `B` 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("1");  }  catch (Exception e){ };
    try {  b.alog("2"); }  catch (Exception e){ };
    try {  a.alog("3"); }  catch (Exception e){ };
    try {  a.blog("4"); }  catch (Exception e){ };
  }
}
```

Användaren som startar programmet har rätt att läsa och skriva `events.log` som från början är tom. Vad kommer det att stå i `events.log` efter att programmet körts en gång (motivera genom att förklara vilka för frågan relevanta kontroller som sker vid körningen)? [3p]