KTH Datavetenskap
och kommunikation

2D1395 Datasäkerhet
Tentamen 2007-01-15 kl 9.00–13.00

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

1a Vad är en referensmonitor och vad har den för uppgift(er)? Ge exempel på hur en OS-kärna kan agera referensmonitor. [2p]

1b När används en ”stackwalk” i Javas säkerhetslösning? Ge ett förklarande exempel. [2p]

1c Vad innebär det att ge ett ”proof-of-concept” (PoC) för ett säkerhetshål? Varför skulle man vilja konstruera ett PoC (bortsett från rent egoistiska skäl)? Vilka risker finns med att publicera ett PoC? [2p]

1d Vad innebär *non-repudiation* (”oförnekbarhet”)? Vad skulle det kunna innebära konkret i samband med handel med en internetbutik? [2p]

2 Lösenordsinloggning är vanlig, och genom att lagra saltade och hashade lösenord så slipper man en del typer av attacker, men denna procedur är fortfarande för osäker för t.ex. banktransaktioner över Internet eller access till datasystem med höga säkerhetskrav. Ge två exempel på hur man kan uppnå en högre grad av säkerhet. Det ena exemplet ska vara lämpligt för en internetbank och det andra för en militär datacentral. [2p]

3 Låt $n = 33, e = 3$ vara ett publikt RSA-nyckelpar (där n är på tok för litet för att kunna användas i praktiken). Låt oss anta att du uppsnappar kryptotexten (talet) 13.

Vilken är klartexten (svara med ett heltal), och vilken privat nyckel svarar mot den publika? [3p]

4 Det finns olika typer av brandväggar: tillståndslösa paketfilter, paketfilter med tillstånd och applikationsproxies. Förklara översiktligt vad respektive metod innebär. [3p]

5 Företaget ”Fast and Loose” har designat en egen hashfunktion som de tycker bör ersätta SHA-1 eftersom deras funktion går mycket fortare att beräkna. Du har kommit över specifikationen till deras hashfunktion: De delar upp meddelandet i 1024-bitsblock och bildar bitvis xor av alla block. Sedan körs SHA-1 på det 1024-bitsblock som är resultat av dessa xor, och det är hashvärdet. Det går mycket fortare att beräkna SHA-1 på 1024 bitar än på en fil som innehåller många MB data.

Visa att deras hashfunktion inte är bra att använda som hashfunktion för digitala signaturer. Mer exakt, visa att om du har ett signerat meddelande från en person (om du vill kan du anta att det är en RSA-signatur) så kan du konstruera ett nytt meddelande med giltig signatur (dvs som ser ut att vara signerat av samma person trots att hon inte bidragit till skapandet av signaturen). [3p]

6 I kursen har vi tagit upp flera representationer för accesskontroll: en accessmatris, accesskontrollistor och s.k. ”capabilities”. Nedan ges en accessmatris som representerar användares rättigheter över filer.

användare \ fil	`foo`	`bar`	`baz`
`ann`	`orw`	`orx`	`z`
`bob`			`orz`
`jan`		`r`	`x`
`zed`	`rx`	`rw`	`z`

Det finns fem olika rättigheter: `o`, `r`, `w`,`x`, `z`. De har följande innebörd:

`o`	Own: kan ändra rättigher för filen, se nedan.
`r`	Read: får läsa filen.
`w`	Write: får modifiera filen.
`x`	Execute: får exekvera filen, se nedan.
`z`	Execute as owner: får exekvera filen, se nedan.

Varje fil har exakt en ägare. Ägaren kan lägga till och ta bort rättigheter till filen för sig själv och för andra. Rättigheten ”`o`” kan dock inte läggas till eller tas bort. Däremot kan användare *u* ”ge bort” en fil hon äger till användare *v*. Om *u* ger filen `foo` till *v* så förlorar *u* rättigheten `o` till `foo`, medan *v* samtidigt får rättigheten `o` till `foo`. På så vis är man säker på att varje fil alltid har precis en ägare.

Skillnaden på `x` och `z` är vilka rättigheter programmet har under exekveringen. I det första fallet har det samma rättigheter som användaren som startar det och i det andra fallet har det samma rättigheter som den som äger programmet. En användare kan inte ha både `x` och `z` till ett visst program. Om *u* har `x` och sedan får `z` så förlorar *u* `x` för den filen (och om *u* har `z` och sedan får `x` så förloras `z`). Det är därför väldefinierat vilka rättigheter som gäller när en användare exekverar ett program.

6a Visa hur matrisen förändras om `ann` ger `bar` till `jan`. [1p]

6b Hur skulle rättigheterna i matrisen representeras om man använder accesslistor? Hur skulle rättigheterna i matrisen representeras om man använder ”capabilities”? [2p]

6c Det finns ett alvarligt säkerhetsproblem förknippat med den här sättet att hantera rättigheter. Vari består problemet? [2p]

7 Följande fragment C-kod har flera säkerhetsproblem.

```
/* Konverterar unixradslut till windowsradslut genom att
 * ersätta \n med \r\n
 */
void convert_eol(char s[]) {
  char t[10000];
  int i=0, n=0;
  while(s[i] != '\0') { /* tills strängen är slut */
    if (s[i] == '\n')
      t[n++] = '\r';
    t[n++] = s[i++];
  }
  for(i=0; i<n; ++i) s[i]=t[i]; /* kopiera tillbaka konverterad sträng */
  s[n] = '\0'; /* markera att strängen slutar här */
}
```

Vilka problem finns med koden? Vilket problem bedömer du som allvarligast (motivera)? Hur bör man modifiera koden (och ev funktionens gränssnitt) för att den ska fungera på avsett sätt och för att undvika odefinierat beteende? [3p]

8 Låt oss tänka oss att vi för datasäkerhetskursen skulle ha en webbsida där man kan få tag på gamla tentor genom att fylla i ett formulär och där ange datum för tentan på formen ”2007-01-15”. Tentalydelserna ligger i en katalog `tentor` där webbservern har läsrättigheter och varje tenta har en underkatalog. Till exempel skulle dagens tentallydelse ligga i `tentor/2007-01-15/ten.pdf`.

Det ifyllda webbformuläret hanteras av webbservern på följande sätt (pseudokod):

```
/* date är den sträng som skrivits i formuläret
   today är dagens datum på formen ÅÅÅÅ-MM-DD
*/

/* Kolla att tentan som efterfrågas redan ägt rum */
if (date < today) {  // strängjämförelse
  filename = "tentor/" + date + "/ten.pdf";
  ..... /* kod som skickar filen filename till webbklienten */
}
```

8a Visa att denna kontroll är otillräcklig för att hindra att någon får tag på dagens, eller kommande, tenta om dess lydelse redan lagts in. [1p]

8b En tänkbar åtgärd skulle vara att inte ge webbservern läsrättigheter i en tentas katalog förrän efter att tentan gått. Varför är den åtgärden otillräcklig? [1p]

8c Föreslå lämplig förändring sådan att formuläret kan användas men missbruk förhindras eller i varje fall försvåras väsentligt. [1p]