**NTNU**
**Norges teknisk-naturvitenskapelige universitet**

**Fakultet for informasjonsteknologi, matematikk og elektroteknikk**

**Institutt for datateknikk og informasjonsvitenskap**

**BOKMÅL//NYNORSK/ENGLISH**

# AVSLUTTENDE EKSAMEN I/FINAL EXAM

## TDT4237
## Programvaresikkerhet/Software Security

## Onsdag/Wednesday 19.12.2007

## Kl. 09.00 – 13.00

**Faglig kontakt under eksamen:**

Lillian Røstad, tlf. 994 00 628

**Hjelpemidler (supporting materials):**

D: Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.
*D: No written or printed supporting materials allowed. Basic calculator allowed.*

**Sensurdato (date for examination results):**

18. januar 2008. Resultater gjøres kjent på http://studweb.ntnu.no/ og sensurtelefon 81548014.

*January 18th 2008. Results available at http://studweb.ntnu.no/ and phone 81548014.*

**Det er angitt i prosent hvor mye hver deloppgave teller ved sensur.**

***Each task is labeled with maximum obtainable score in percentage of the total score.***

# BOKMÅL

## Oppgave 1 (20%) – Code Quiz – Spot the bug

I denne pseudo-Java-kodesnutten er det feil som kan forårsake sikkerhetsproblemer. Identifiser, klassifiser og beskriv feilene. Forklar også hvordan du ville ha gått frem for å endre på koden slik at samme funksjonalitet ble beholdt, men uten sårbarheter. Skriv gjerne pseudokode for å forklare.

```
public class RiskyClass {

 public static void main (String args[] ) {

  HandyClass handy = new HandyClass();
  //A secure logId is calculated to prevent tampering and targeted overwriting of the log
  int logId = handy.calculate(args[0]);

  handy.writeToLog(logId, args[1], args[2]){

 }
}

public class HandyClass {

   Random randy = new Random();
   String loglocation = ”C:/logg/logger/”;

   //This method calculates the logId.
   public int calculate (int val) {
       int result = val + randy.next();
       return result;
   }


   public void writeToLog(int id, String loggfile, String entry) throws java.lang.Exception {

      writeToFile(loglocation, loggfile, id, entry);

   }

}
```

# Oppgave 2 (25%)

a) Forklar kort begrepene Protection Profile, Security Target, TOE, Security Functionality og AEL slik de brukes i Common Criteria.
b) Input validering er et av de viktigste tiltak for økt sikkerhet i webapplikasjoner. List 4 ulike angrep som input validering kan fungere som beskyttelse mot, og forklar kort hvordan for hver av de.
c) Hva er et security pattern? Gi ett eksempel på et security pattern og forklar kort hvordan det kan brukes.
d) Hvilke faser inngår i Risk Management Framework (RMF)? Forklar kort hver fase i RMF.
e) Forklar begrepet race condition. Gi ett kort eksempel på hvordan race conditions kan føre til sikkerhetsproblemer.

# Oppgave 3 (40%) - Case

Du har laget et system for innsamling av epidemiologiske data. Dette systemet henter inn informasjon fra flere sykehus og bruker det til å forutsi utbrudd av epidemier – for eksempel influensa. Hvert sykehus har forskjellig format på dataene sine, og hittil har du derfor måttet skrive spesifikke filter for å tolke dataene fra hvert enkelt sykehus.
Nå ønsker du å utvide til mange flere datakilder (sykehus) og du ønsker da at IT-folkene ved sykehuset skal skrive disse filtrene selv. Kjernen i systemet ditt er skrevet i programmeringsspråket C, og filtrene skal derfor skrives i samme språk. Du ønsker å tilby et webgrensesnitt til systemet som lar de lokale IT-folkene ved sykehusene laste opp koden for filtrene de har laget inn i ditt system. Filtre og rådata lastes altså inn i ditt system og selve filtreringen av data foregår der.

Å tillate opplasting av kode er en klar sikkerhetsrisiko, men å tillate dette er en nødvendighet for systemet ditt. Forklar og demonstrer hvordan du vil gå frem for å innarbeide denne funksjonaliteten i systemet, slik at sikkerhetsrisikoen blir minst mulig. Gjør rede for hele prosessen fra planlegging/design til implementasjon og testing. Du trenger *ikke* å skrive noe kode, men forklar hvordan du ville tenkt.

Gjør antagelser der du må, men marker og forklar de tydelig.

# Oppgave 4 (15%) – Trusselmodellering

Du har blitt ansatt på et prosjekt for et sykehus der hovedansvaret ditt er risikovurdering. I prosjektet skal det utvikles et nytt system for automatisk håndtering av logger fra pasientjournalen. Systemet skal brukes til å analysere loggene for å identifisere mulige angrep og misbruk. I denne første delen av prosjektet er fokus insidere – dvs. trusselen fra autoriserte brukere som misbruker sine rettigheter.
I denne oppgaven skal det gjøres en overordnet modellering av mulige trusler som man kan forvente at insidere utgjør. Mao handlingsmønstre man kan forvente å finne igjen i loggen. Modelleringen skal gjøres i et møte med helsepersonell. Velg en notasjon og fremgangsmåte som du mener er hensiktsmessig for trusselmodellering på dette nivået.

Du skal *ikke* fullføre en fullstendig risikoanalyse – fokus i denne oppgaven er på trusselmodellering.

# NYNORSK

## Oppgåve 1 (20%) – Code Quiz – Spot the bug

I denne pseudo-Java-kodesnutten er det feil som kan forårsaka sikkerheitsproblema. Identifiser, klassifiser og beskriv feilane. Forklar også korleis du ville ha gått frem for å endre på koden slik at same funksjonalitet vert behald, men utan sårbarhetar. Skriv gjerne pseudokode for å forklåra.

```
public class RiskyClass {

 public static void main (String args[] ) {

  HandyClass handy = new HandyClass();
  //A secure logId is calculated to prevent tampering and targeted overwriting of the log
  int logId = handy.calculate(args[0]);

  handy.writeToLog(logId, args[1], args[2]){

 }
}

public class HandyClass {

   Random randy = new Random();
   String loglocation = ”C:/logg/logger/”;

   //This method calculates the logId.
   public int calculate (int val) {
       int result = val + randy.next();
       return result;
   }


   public void writeToLog(int id, String loggfile, String entry) throws java.lang.Exception {

      writeToFile(loglocation, loggfile, id, entry);

   }

}
```

# Oppgåve 2 (25%)

a) Gjer kort greie for uttrykka Protection Profile, Security Target, TOE, Security Functionality og AEL slik dei vart brukt i Common Criteria.
b) Input validering er eit av de viktigaste tiltaka for økt sikkerheit i webapplikasjonar. List 4 ulike angrep som input validering kan fungere som beskyttelse mot, og forklar kort korleis for kvar av de.
c) Kva er et security pattern? Gi eitt eksempel på et security pattern og forklar kort korleis det kan brukast.
d) Kva fasar inngår i Risk Management Framework (RMF)? Forklar kort kvar fase i RMF.
e) Forklar uttrykket race condition. Gi eitt kort eksempel på korleis race conditions kan føre til sikkerheitsproblema.

# Oppgåve 3 (40%) - Case

Du har laga eit system for innsamling av epidemiologiske data. Dette systemet hentar inn informasjon frå fleire sjukehus og bruker det til å føresjå utbrot av epidemiar – for eksempel influensa. Kvart sjukehus har forskjellig format på dataene sine, og hittil har du derfor måtta skrive spesifikke filter for å tolke dataene frå kvart enkelt sjukehus.
Nå ynskje du å utvide til mange fleire datakildar (sjukehus) og du ynskjer da at IT-folka ved sjukehuset skal skrive disse filtrene sjølv. Kjernen i systemet ditt er skrevet i programmeringsspråket C, og filtrene skal derfor skrives i same språk. Du ynskjer å tilby et webgrensesnitt til systemet som lar de lokale IT-folka ved sjukehusa laste opp koden for filtrene de har laget inn i ditt system. Filtre og rådata lastas altså inn i ditt system og sjølve filtreringen av data skjer der.

Å tillata opplasting av kode er ein klar sikkerheitsrisiko, men å tillate dette er ein nødvendigheit for systemet ditt. Forklar og demonstrer korleis du vil gå frem for å innarbeide denne funksjonaliteten i systemet, slik at sikkerheitsrisikoen blir minst mulig. Grei ut om heile prosessen frå planlegging/design til implementasjon og testing. Du treng *ikkje* skrive kode, men forklår korleis du ville ha tenkt.

Gjør antakingar der du må, men marker og forklar dei tydeleg.

# Oppgåve 4 (15%) – Trusselmodellering

Du har vorte ansatt på eit prosjekt for et sjukehus der hovudansvaret ditt er risikovurdering. I prosjektet skal det utviklast et nytt system for automatisk handtering av logger frå pasientjournalen. Systemet skal brukast til å analysere loggene for å identifisere mulige angrep og misbruk. I denne første delen av prosjektet er fokus insidere – dvs. trusselen frå autoriserte brukarar som misbruker sine rettigheitar.
I denne oppgåva skal det gjørast ein overordna modellering av mulige truslar som man kan forvente at insidere utgjør. Mao handlingsmønster man kan forvente å finne igjen i loggen. Modelleringa skal gjørast i et møte med helsepersonell. Vel ein notasjon og framgangsmåte som du meiner er hensiktsmessig for trusselmodellering på dette nivået.

Du skal *ikkje* fullføre ein fullstendig risikoanalyse – fokus i denne oppgåva er på trusselmodellering.

# ENGLISH

## Task 1 (20%) – Code Quiz – Spot the bug

In this pseudo-Java-code excerpt there are hidden security vulnerabilities. Identify, classify and describe the vulnerabilities. Additionally, explain how you would change the code so that the functionality remains the same but without the vulnerabilities. You may use pseudo code to elaborate your answers.

```
public class RiskyClass {

 public static void main (String args[] ) {

  HandyClass handy = new HandyClass();
  //A secure logId is calculated to prevent tampering and targeted overwriting of the log
  int logId = handy.calculate(args[0]);

  handy.writeToLog(logId, args[1], args[2]){

 }
}

public class HandyClass {

   Random randy = new Random();
   String loglocation = ”C:/logg/logger/”;

   //This method calculates the logId.
   public int calculate (int val) {
       int result = val + randy.next();
       return result;
   }


   public void writeToLog(int id, String loggfile, String entry) throws java.lang.Exception {

      writeToFile(loglocation, loggfile, id, entry);

   }


}
```

## Task 2 (25%)

a) Explain the terms Protection Profile, Security Target, TOE, Security Functionality and AEL as defined and used in the Common Criteria.
b) Input validation is one of the most important and common security measures for web applications. List 4 different attacks where input validation is a sensible countermeasure. For each of the attacks, explain how input validation may be used as a countermeasure.
c) What is a security pattern? Provide one example of a security pattern and explain how it may be used.
d) What subprocesses are included in the Risk Management Framework (RMF)? Explain each sub process briefly.
e) Explain the term race condition. Provide a brief example of how a race condition may constitute a security issue in a system.

## Task 3 (40%) - Case

You have developed a system for collecting epidemiological data from several hospitals. The systems groups and compares the data to predict epidemic outbreaks – for instance the flu. The data format is specific to each hospital. So far you have had to write filters for each hospital in order to transform the data to a common format usable in your system.
In the next step your goals is to expand to several more hospitals. To decrease your burden, you now want the local IT-people at each hospital to provide the filters. Your system has been written in the programming language C, which means that the filters have to be written in C as well. To facilitate this you plan to provide a web interface through which the local IT-people can upload their code and data to your system. To be specific, both the filters and the data from the hospitals are loaded into your system, and the processing taken care of inside your system.
Allowing outsiders to upload code into your system clearly constitutes a security risk, but it is necessary for the system to be usable. Explain and demonstrate how you would go about integrating this functionality into your system, such that the resulting security risk is minimized. Explain the entire process from design to implementation and testing. You are *not* supposed to write any code, but explain what your considerations are.

You may make assumptions where needed, but make sure to clearly indicate what assumptions are made and explain why.

## Task 4 (15%) – Threat modeling

You have been hired to work on a project for a hospital where your main responsibility is risk analysis. The aim of the project is to develop a system for automatic handling of audit logs. The system will be used to identify potential attacks and misuse. In this first phase of the project the focus is on insiders – i.e. the threat constituted by authorized users misusing their privileges.
In this task you should perform a high-level threat modeling of potential threats from insiders. In other words usage patterns one may locate in the audit log that constitutes misuse.

The threat modeling is to be performed in a meeting with health care personnel. Choose a modeling language and approach that you consider suitable for such a setting.

You shall *not* perform a complete risk analysis. Focus in this task is on threat modeling.