NTNU
Institutt for datateknikk
og informasjonsvitenskap

Eksamensoppgave i/Exam

TDT4237 – Programvaresikkerhet/Software Security

Torsdag/Thursday 8. desember/December 2011, 09:00 - 13:00

Oppgaven er utarbeidet av faglærer Lillian Røstad og kvalitetssikret av Torbjørn Skramstad.
Kontaktperson under eksamen er Lillian Røstad (mobil 994 00 628)

Språkform: Engelsk/Bokmål/Nynorsk
Tillatte hjelpemidler: D
Ingen trykte eller håndskrevne hjelpemidler tillatt. No printed or hand-written materials allowed.
Bestemt, enkel kalkulator tillatt. Approved calculator allowed.

Sensurfrist (results available): 9. januar 2011 (January 9th 2011)

Les oppgaveteksten nøye. Finn ut hva det spørres etter i hver oppgave.

Dersom du mener at opplysninger mangler i en oppgaveformulering gjør kort rede for de antagelser og forutsetninger som du finner det nødvendig å gjøre.

Read each task carefully. Identify what the task asks for.

If you find that information is missing in a task you are free to make the assumptions you consider necessary, but remember to explain and clearly mark your assumptions.

ENGLISH

Task 1 (25% - each task counts 5%)

a) Explain how a CSRF/XSRF attack is executed. Please provide an example to illustrate.
b) One of the main principles of security is *the principle of least privilege*. Explain how this principle should be used in a coding context.
c) What is the *attack surface* of a system? How do you identify a system's attack surface?
d) What is a security requirement? What are the criteria for writing a good security requirement according to Firesmith? Please explain using examples.
e) What is a zero-day vulnerability?

Task 2 - From threat modelling to penetration testing (20%)

Gary McGraw once said: "If you fail a penetration test you know you have a very bad problem indeed. If you pass a penetration test you do not know that you don't have a very bad problem".

While this is true, penetration testing is often the only type of security testing performed on a system as it can be done fairly easy and with limited resources. However, as with all testing, the quality of the test relies on the time and effort that goes into planning the test.

You have been given the task of performing a penetration test on a website for a bookstore. The website includes an open section that allows customers to browse for books by topic, title or author. It also includes a section for ordering and paying for books that requires users to be authenticated.

Your goal is to provide the best possible test coverage with limited resources, and also to be able to communicate your test plan and the results to the customer.

How would you plan the penetration test for this site?

What techniques would you use, and why?

Your answer should include a demonstration/partial test plan specific to this case to illustrate your choices.

Task 3 – Risk Management – Smart Grid (35 %)

According to an article in IEEE Security and privacy by Patrick McDaniel and Stephen McLaughlin, the Smart Grid is:

"(…) a network of computers and power infrastructure that monitor and manage energy usage. Each energy producer—for example, a regional electrical company— maintains operational centers that receive usage information from collector devices placed throughout the served area. In a typical configuration, a neighborhood contains a single collector device that will receive periodic updates from each customer in the neighborhood via a wireless mesh network. The collector device reports usage readings to the operational centers using a long-haul communication media such as a dial-up line or the Internet. The utilities manage transmission and perform billing based on these readings.

The usage-reporting device at each customer site is called the smart meter. It's a computerized replacement of the electrical meter attached to the exterior of many of our homes today. Each smart meter contains a processor, nonvolatile storage, and communication facilities. Although in many respects, the smart meter's look and function is the same as its unsophisticated predecessor, its additional features make it more useful. Smart meters can track usage as a function of time of day, disconnect a customer via software, or send out alarms in case of problems. The smart meter can also interface directly with "smart" appliances to control them—for example, turn down the air conditioner during peak periods."

Your company has won a contract to create smart meters – devices to be deployed in every customer's home that allows the customers to manage their energy usage. This facilitates smart energy management to reduce costs, for instance if energy is more expensive at certain time periods, but it can also allow customers to act as energy producers. An example would be a household that is both connected to the power infrastructure and has their own solar energy equipment. What energy they use depends on the current energy balance: they rely on the energy from the power infrastructure only when not enough solar energy is generated. If they generate more solar energy than they need, they may sell their surplus energy using the smart meter and the smart grid infrastructure.

While Smart Grid technology has many potential benefits, there are also risks involved, both related to security and privacy. For this task, you should focus on the smart meter. Using the Risk Management Framework, complete an entire iteration working from business to technical risks for the smart meter. Remember to rank and prioritize the risks.

Task 4 – Code Quiz – Spot the bug (20%)

Each of the following code fragments contains a security issue. For each case you should: identify the issue and explain why it is a problem, classify the error according to McGraw's taxonomy of coding errors, and provide a solution illustrating how you would fix the code.

Assume that all the code fragments are written in Java.

Case 1:

```
class Login {
  public Connection getConnection() throws SQLException {
    DriverManager.registerDriver(new
            com.microsoft.sqlserver.jdbc.SQLServerDriver());
    String dbConnection =
      PropertyManager.getProperty("db.connection");
    // can hold some value like
    // "jdbc:microsoft:sqlserver://<HOST>:1433,<UID>,<PWD>"
    return DriverManager.getConnection(dbConnection);
  }

  String hashPassword(char[] password) {
    // create hash of password
  }

  public void doPrivilegedAction(String username, char[] password)
                                  throws SQLException {
    Connection connection = getConnection();
    if (connection == null) {
      // handle error
    }
    try {
      String pwd = hashPassword(password);

      String sqlString = "SELECT * FROM db_user WHERE username = '"
                         + username +
                         "' AND password = '" + pwd + "'";
      Statement stmt = connection.createStatement();
      ResultSet rs = stmt.executeQuery(sqlString);

      if (!rs.next()) {
        throw new SecurityException(
          "User name or password incorrect"
        );
      }

      // Authenticated; proceed
    } finally {
      try {
        connection.close();
      } catch (SQLException x) {
        // forward to handler
      }
    }
  }
}
```

Case 2

```
class IPaddress {
  String ipAddress = new String("172.16.254.1");
  public static void main(String[] args) {
    //..
  }
}
```

Case 3

```
try {
  FileInputStream fis =
      new FileInputStream(System.getenv("APPDATA") + args[0]);
} catch (FileNotFoundException e) {
  // Log the exception
  throw new IOException("Unable to retrieve file", e);
}
```

Case 4

```
FileOutputStream fis = new FileOutputStream(new File("/img/" + args[0]));
// ...
```

BOKMÅL

Oppgave 1 (25% - hver oppgave teller 5%)

a) Forklar hvordan et CSRF/XSRF angrep utføres. Illustrer med et eksempel.
b) Et av de viktigste prinsippene innen sikkerhet er *the principle of least privilege*. Forklar hvordan dette prinsippet skal brukes i en programmeringskontekst.
c) Hva er angrepsflaten til et system? Hvordan identifiserer du et systems angrepsflate?
d) Hva er et sikkerhetskrav? Hva er kriteriene for å skrive gode sikkerhetskrav, i følge Firesmith? Bruk eksempler for å forklare.
e) Hva er en null-dags sårbarhet (zero-day vulnerability)?

Oppgave 2 - Fra trusselmodellering til penetrasjonstesting (20%)

Gary McGraw har uttalt: "If you fail a penetration test you know you have a very bad problem indeed. If you pass a penetration test you do not know that you don't have a very bad problem".

Selv om dette er sant, er penetrasjonstesting ofte den eneste formen for sikkerhetstesting som utføres på et system siden det kan gjøres relativ enkelt og med begrensede ressurser. Men, som med all testing, avhenger kvaliteten på testen av tid og innsats som brukes på testplanlegging.

Din oppgave er å utføre penetrasjonstesting på websiden til en bokhandel. Websiden inkluderer både en åpen del der man kan se etter bøker basert på tema, tittel eller forfatter og en de der man må være autentisert for å kunne bestille og betale for bøker.

Målet ditt er å oppnå best mulig testdekning med begrensede ressurser, i tillegg til å kunne kommunisere testplanen og resultatene til kunden.

Hvordan ville du planlegge penetrasjonstesten?

Hvilke teknikker ville du bruke og hvorfor?

Svaret dit bør inkludere en demonstrasjon/delvis testplan for denne oppgaven for å illustrere valgene dine.

Oppgave 3 – Risk Management – Smart Grid (35 %)

I følge en artikkel i IEEE Security and privacy av Patrick McDaniel og Stephen McLaughlin er Smart Grid:
"(…) a network of computers and power infrastructure that monitor and manage energy usage. Each energy producer—for example, a regional electrical company— maintains operational centers that receive usage information from collector devices placed throughout the served area. In a typical configuration, a neighborhood contains a single collector device that will receive periodic updates from each customer in the neighborhood via a wireless mesh network. The collector device reports usage readings to the operational centers using a long-haul communication media such as a dial-up line or the Internet. The utilities manage transmission and perform billing based on these readings.
The usage-reporting device at each customer site is called the smart meter. It's a computerized replacement of the electrical meter attached to the exterior of many of our homes today. Each smart meter contains a processor, nonvolatile storage, and communication facilities. Although in many respects, the smart meter's look and function is the same as its unsophisticated predecessor, its additional features make it more useful. Smart meters can track usage as a function of time of day, disconnect a customer via software, or send out alarms in case of problems. The smart meter can also interface directly with "smart" appliances to control them—for example, turn down the air conditioner during peak periods."

Ditt firma har vunnet et oppdrag for å lage *smart meter* – enheter som skal utplasseres hos hver kunde som gjør kunden i stand til selv å kontrollere sin energibruk. Dette muliggjør smart energibruk for å redusere kostnader, for eksempel dersom energi er mer kostbart i bestemte tidsperioder, men kan også gjøre det mulig for kunder å være energiprodusenter. Et eksempel er en husholdning som både er tilknyttet energinettet og har sitt eget solenergiutstyr. Hvilken energi de velger å bruke avhenger av energibalansen til enhver tid: de bruker energinettet kun når de ikke selv produserer tilstrekkelig solenergi. Dersom de produserer mer solenergi enn de selv har bruk for, kan de også velge å selge overskuddsenergien ved hjelp av smart meter og smart grid infrastrukturen.

Smart Grid teknologi har mange mulige fordeler, men også utfordringer relatert til sikkerhet og personvern. I denne oppgaven skal du fokusere på smart meter. Bruk Risk Management Framework til å fullføre en komplett syklus der du går fra forretningsrisiko til tekniske risiko for smart meter. Husk å rangere og prioritere risiko.

Oppgave 4 – Code Quiz – Spot the bug (20%)

Hver av de følgende kodefragmenter inneholder sikkerhetsproblemer. For hvert tilfelle skal du: identifisere problemet og forklare hvorfor det er et problem, klassifisere feilen i henhold til McGraw sin taksonomi for kodefeil og foreslå en løsning som forklarer hvordan du vil rette problemet.

Anta at alle kodefragmentene er skrevet i Java.

Case 1:

```
class Login {
  public Connection getConnection() throws SQLException {
    DriverManager.registerDriver(new
            com.microsoft.sqlserver.jdbc.SQLServerDriver());
    String dbConnection =
      PropertyManager.getProperty("db.connection");
    // can hold some value like
    // "jdbc:microsoft:sqlserver://<HOST>:1433,<UID>,<PWD>"
    return DriverManager.getConnection(dbConnection);
  }

  String hashPassword(char[] password) {
    // create hash of password
  }

  public void doPrivilegedAction(String username, char[] password)
                                 throws SQLException {
    Connection connection = getConnection();
    if (connection == null) {
      // handle error
    }
    try {
      String pwd = hashPassword(password);

      String sqlString = "SELECT * FROM db_user WHERE username = '"
                       + username +
                       "' AND password = '" + pwd + "'";
      Statement stmt = connection.createStatement();
      ResultSet rs = stmt.executeQuery(sqlString);

      if (!rs.next()) {
        throw new SecurityException(
          "User name or password incorrect"
        );
      }

      // Authenticated; proceed
    } finally {
      try {
        connection.close();
      } catch (SQLException x) {
        // forward to handler
      }
    }
  }
}
```

Case 2

```
class IPaddress {
  String ipAddress = new String("172.16.254.1");
  public static void main(String[] args) {
    //..
  }
}
```

Case 3

```
try {
  FileInputStream fis =
      new FileInputStream(System.getenv("APPDATA") + args[0]);
} catch (FileNotFoundException e) {
  // Log the exception
  throw new IOException("Unable to retrieve file", e);
}
```

Case 4

```
FileOutputStream fis = new FileOutputStream(new File("/img/" + args[0]));
// ...
```

NYNORSK

Oppgåve 1 (25% - kvar oppgåve teller 5%)

a) Forklar korleis eit CSRF/XSRF angrep vert utført. Klargjer med eit eksempel.
b) Eit av de viktigaste prinsippa innom sikkerheit er *the principle of least privilege*. Forklår korleis dette prinsippet skal brukast i ein programmeringskontekst.
c) Kva er angrepsflata til eit system? Korleis finn du angrepsflata til eit system?
d) Kva er eit sikkerheitskrav? Kva er kriteria for å skrive gode sikkerheitskrav, i følgje Firesmith? Bruk eksempla for å forklåre.
e) Kva er en null-dags sårbarheit (zero-day vulnerability)?

Oppgåve 2 - Frå trusselmodellering til penetrasjonstesting (20%)

Gary McGraw har uttala: “If you fail a penetration test you know you have a very bad problem indeed. If you pass a penetration test you do not know that you don’t have a very bad problem”.

Sjølv om dette er sant, er penetrasjonstesting ofte den einaste forma for sikkerheitstesting som vert utført på eit system ettersom det kan gjerast relativ enkelt og med begrensa ressurser. Men, som med all testing, avheng kvaliteten på testen av tid og innsats som vert brukt på testplanlegging.

Di oppgåve er å utføre penetrasjonstesting på websida til ein bokhandel. Websida inneheld både ein åpen del der ein kan sjå etter bøker sortert på tema, tittel eller forfattar og ein del der ein må være autentisert for å kunne bestille og betale for bøker.

Målet ditt er å oppnå best mogeleg testdekning med begrensa ressurser, i tillegg må du kunne kommunisere testplanen og resultata til kunden.

Korleis vil du planlegge penetrasjonstesten?

Kva for teknikkr ville du bruke og kvifor?

Svaret dit bør innehalde ein demonstrasjon/delvis testplan for denne oppgåva for å vise prioriteringane dine.

Oppåve 3 – Risk Management – Smart Grid (35 %)

Etter ein artikkel i IEEE Security and privacy av Patrick McDaniel og Stephen McLaughlin er Smart Grid:
"(…) a network of computers and power infrastructure that monitor and manage energy usage. Each energy producer—for example, a regional electrical company— maintains operational centers that receive usage information from collector devices placed throughout the served area. In a typical configuration, a neighborhood contains a single collector device that will receive periodic updates from each customer in the neighborhood via a wireless mesh network. The collector device reports usage readings to the operational centers using a long-haul communication media such as a dial-up line or the Internet. The utilities manage transmission and perform billing based on these readings.
The usage-reporting device at each customer site is called the smart meter. It's a computerized replacement of the electrical meter attached to the exterior of many of our homes today. Each smart meter contains a processor, nonvolatile storage, and communication facilities. Although in many respects, the smart meter's look and function is the same as its unsophisticated predecessor, its additional features make it more useful. Smart meters can track usage as a function of time of day, disconnect a customer via software, or send out alarms in case of problems. The smart meter can also interface directly with "smart" appliances to control them—for example, turn down the air conditioner during peak periods."

Ditt firma har vunne eit oppdrag for å lage *smart meter* – enheiter som skal utplasserast hos alle kunder og som gjer kunden i stand til sjølv å kontrollere energibruken sin. Dette muliggjer smart energibruk for å redusere kostnader, til dømes om energi er meir kostbart i bestemte tidsperiodar, men kan og gjere det mogeleg for kundar å være energiprodusentar. Eit eksempel er ein husholdning som både er tilknyttet energinettet og som har sitt eige solenergiutstyr. Kva for energi de velger å bruke avheng av energibalansen til kvar tid: dei nyttar energinettet berre når dei ikkje sjølv produserer nok solenergi. Om dei produserer meir solenergi enn dei sjølv har bruk for, kan dei og velje å selja overskuddsenergien ved bruk av smart meter og smart grid infrastrukturen.

Smart Grid teknologi har mange moglege fordeler, men og utfordringer knytta til sikkerheit og personvern. I denne oppgåva skal du fokusere på smart meter. Bruk Risk Management Framework til å gjennomfør ein komplett syklus der du går frå forretningsrisiko til teknisk risiko for smart meter. Husk å rangere og prioritere risiko.

Oppgåve 4 – Code Quiz – Spot the bug (20%)

Kvar av de følgende kodefragmenter inneheld sikkerheitsproblemer. For kvart tilfelle skal du: identifisere problemet og forklare kvifor det er eit problem, klassifisere feilen i etter McGraw sin taksonomi for kodefeil og foreslå e løysing som viser korleis du vil rette på problemet.

Anta at alle kodefragmentene er skrevet i Java.

Case 1:

```
class Login {
  public Connection getConnection() throws SQLException {
    DriverManager.registerDriver(new
            com.microsoft.sqlserver.jdbc.SQLServerDriver());
    String dbConnection =
      PropertyManager.getProperty("db.connection");
    // can hold some value like
    // "jdbc:microsoft:sqlserver://<HOST>:1433,<UID>,<PWD>"
    return DriverManager.getConnection(dbConnection);
  }

  String hashPassword(char[] password) {
    // create hash of password
  }

  public void doPrivilegedAction(String username, char[] password)
                                 throws SQLException {
    Connection connection = getConnection();
    if (connection == null) {
      // handle error
    }
    try {
      String pwd = hashPassword(password);

      String sqlString = "SELECT * FROM db_user WHERE username = '"
                       + username +
                       "' AND password = '" + pwd + "'";
      Statement stmt = connection.createStatement();
      ResultSet rs = stmt.executeQuery(sqlString);

      if (!rs.next()) {
        throw new SecurityException(
          "User name or password incorrect"
        );
      }

      // Authenticated; proceed
    } finally {
      try {
        connection.close();
      } catch (SQLException x) {
        // forward to handler
      }
    }
  }
}
```

Case 2

```
class IPaddress {
  String ipAddress = new String("172.16.254.1");
  public static void main(String[] args) {
    //..
  }
}
```

Case 3

```
try {
  FileInputStream fis =
      new FileInputStream(System.getenv("APPDATA") + args[0]);
} catch (FileNotFoundException e) {
  // Log the exception
  throw new IOException("Unable to retrieve file", e);
}
```

Case 4

```
FileOutputStream fis = new FileOutputStream(new File("/img/" + args[0]));
// ...
```