

FUNKTIONALITET FÖR SPIONMJUKVARA

MINIPROJEKT I JAVA FÖR IT-FORENSIK PROGRAMMET

INDLEDNING

Företaget *ViktigaStyrelseMöten AB* har styrelsemöten där känsliga beslut om företagets framtid tas. Dessa styrelsemöten äger rum den 25:e klockan 10.00 – 11.00 varje månad. Investerare spekulerar om att företaget går dåligt och ryktet om att beslut skall ha nått konkurrenter är skadligt för företaget. Styrelsen vill behålla investerare som satsat på företagets aktier och vill såklart utreda hur en del beslut nått konkurrenter.

Därför blir företaget väldigt misstänksam när ett program som ingen känner till hittas på den stationära datorn för presentation som står i styrelserummet. Kan någon information läckt ut via denna datorn?

Följande filer hittades:

```
EncryptDecrypt.class
EncryptedListener.class
GenDirHelper.class
ListenScheduler.class
Listener.class
MListener.class
StartProgram.class
PasswordDialog.class
\144216133052740339898257427201239\sk3md8a22ss.d03
99ksnda185kd3.d02
```

Ni anlitas av företagets styrelse för att undersöka programmet och rapportera om programmet kan ha skickat iväg information angående styrelsens olika beslut. Då programmet är skrivit och kompilerat i Java så innehåller class-filerna enbart BYTE-kod istället för läsbar källkod. Innehållet i filerna *sk3md8a22ss.d03* och *99ksnda185kd3.d02* ser "kryptiskt" ut och kan vara en ledtråd till hur programmet fungerar.

UPPGIFT

Er uppgift som grupp blir att (lämplig gruppstorlek är 4-5 personer):

- Exekvera programmet och studera vad som händer.
- Använda ett verktyg för att konvertera Java BYTE-kod till Java källkod (sk. "reverse engineering"). Se länk i sista delen av denna beskrivning. Där finns en länk till ett program som möjliggör omvandling av BYTE-kod till java källkod.
- Ett lösenord krävs för att den mjukvara som hittats skall starta, lösenordet ligger krypterat i filen *99ksnda185kd3.d02*. Studera källkoden och filen för att förstå hur det korrekta lösenordet kan beräknas fram.
- Studera källkoden och förklara vad programmet i sin helhet gör och hur relationerna mellan programmets klasser är uppbyggda.
- Använda klassen "EncryptDecrypt" för att skriva färdigt metoden "decryptDataUsingMethod1(...)" och använd sedan metoden i ett eget skapat Java-program (detta för att kunna dekryptera informationen i *sk3md8a22ss.d03*). Ni vet att ert program dekrypterat informationen korrekt när ni döper om filen *sk3md8a22ss.d03* till *dekrypteratIjud.au* och kan lyssna på dess innehåll.
- Svara på frågan: "Kan programmet sparat information angående styrelsens olika beslut?"
- Sammanställa en skriftlig rapport där ni förklarar
 - Vad varje klass används till.
 - Vad varje metod används till och vad metoden i korthet gör.
 - Programmets helhet (gärna med UML-diagram, detta kan genereras automatiskt med JGrasp).

- Hur ni gick tillväga för att förstå programmets funktion.

RAPPORTERING

OBS! Rapporten skall vara på max 4 sidor (+försättsblad). I rapporten skall enbart kod som producerats (kod som ni tar fram) i projektet finnas med, alltså inte den kod som återskapas av class-filerna.

Ett förslag på rubriker till den skriftliga rapporten:

- **Inledning** – Förklara vilket problem som ligger bakom och syftet till projektet.
- **Metod** – Förklara vilka tekniker som använts för att förstå programmet.
- **Resultat** – Visa strukturen (inte koden) på den omvandlade class-filerna och hur dessa klasser och metoder förhåller sig till varandra. Använd gärna UML-diagram.
- **Analys** – Resonera kring hur ni tolkar källkoden, de övriga filerna och UML-diagrammet.
- **Slutsats** – Sammanfatta era slutsatser kring programmets funktion i denna del av rapporten.

ANVÄNDBARA LÄNKAR OCH TIPS

Verktyg för att omvandla Java BYTE-kod till källkod, enkelt att installera och använda

<http://java.decompiler.free.fr/>

En beskrivning av "reverse engineering"

http://en.wikipedia.org/wiki/Reverse_engineering#Reverse_engineering_of_software

En beskrivning av hur man krypterar och dekrypterar data med tekniken "XOR"

http://en.wikipedia.org/wiki/XOR_encryption

En beskrivning av hur man krypterar och dekrypterar data med tekniken "Caesar krypto"

http://en.wikipedia.org/wiki/Caesar_cipher

En beskrivning av hur man använder XOR-operatoren i java

Java Software Solutions 6th edition, J.Lewis & W.Loftus (2009), p.709-710.

Ett tips: En del kod som blivit dekompilerad kan ibland ge fel vid kompilering, för MListener.java är detta fallet, ersätt därför koden:

```
import javax.sound.sampled.AudioFileFormat.Type;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine.Info;
import javax.sound.sampled.TargetDataLine;
```

med

```
import javax.sound.sampled.*;
```

Detta gäller även filen "PasswordDialog.java" där raden:

```
new PasswordDialog.PasswordButton(null)
```

skall ersättas med

```
new PasswordButton()
```