

Beveiligingsverslag DankBank

Inhoud

Inleiding	2
Problemen	3
Probleem 1:	3
Probleem 2:	3
Probleem 3:	3
Probleem 4:	3
Probleem 5:	4
Conclusie	5
Literatuurlijst	6

Inleiding

In medewerking met DankBank is, aangaande het verzoek om een onderzoek te starten, onderzoek gepleegd naar de beveiliging van de door DankBank gemaakte interface en client. Eerst is de code geanalyseerd, en daarna zijn de bevindingen van deze analyse besproken binnen het onderzoeksteam en zijn de problemen die door het onderzoeksteam ter verbetering vatbaar bevonden opgeschreven en aan de hand daarvan zijn er een aantal adviezen geformuleerd. Uiteindelijk zullen al deze punten onze conclusie vormen.

Problemen

Probleem 1:

Binnen de code (zie DankBankGui) is er weinig onderscheid tussen files en staan verschillende files door elkaar, dit maakt het moeilijk leesbaar en veel zoekwerk voor anderen om de code te begrijpen en deze dusdanig te kunnen bewerken of te beoordelen. Hierbij zou categoriseren dit al verhelpen. Files van dezelfde soort in dezelfde map zetten helpt om de overzichtelijkheid te verbeteren.

Ons advies voor probleem 1 is, zorg dat je code leesbaar is zodat andere het kunnen begrijpen om het verbeteren, uitbreiden of beoordelen. Je kan dit aanpakken door folders aan te maken. In Visual Studio is daar de mogelijkheid voor.

Probleem 2:

De code is normaal geschreven zonder commentaar, voor de leesbaarheid van de code zou het aan te raden zijn om deze te voorzien van commentaar op een stuk code of bij enkele regels er extra uitleg, indien nodig, bij te schrijven. Ook consistent wezen bij het gebruik haakjes zou de code beter leesbaar maken.

Ons advies voor probleem 2 is, plaats commentaar achter een methode. We wisten wat de programma doet in grote lijnen maar bij de classes moesten we meer tijd nemen om goed te begrijpen wat de class doet, en waarvoor het gebruikt word.

Probleem 3:

Op het moment staan er meer libraries dan nodig zijn wat er voor zorgt dat tijdens het compileren libraries worden aangeroepen die niet gebruikt worden. Deze kunnen weggelaten worden om de werking van het programma te bevorderen. Een voordeel die hieruit te behalen is dat de programma sneller compileert.

Probleem 4:

Binnen het bestand Application.cs staat een klasse die niks bevat, deze is totaal onnodig en kan weggelaten worden aangezien deze nergens aan wordt geroepen. Deze weghalen is het verstandigst.

Probleem 5:

De benamingen die zijn gekozen voor de class files zijn origineel maar ze geven niet weer wat ze doen met betrekking tot de code. De file genaamd Kekkeronipizza.cs zou je GUI.cs of Main.cs kunnen noemen. Dat zorgt ervoor dat degene die je code leest weet wat de class doet.

.

Conclusie

Over het algemeen is de beveiliging redelijk, de code is alleen niet geweldig leesbaar en zou eerst aangepast moeten worden op consistentie en dergelijken. Voor de volgende keer zou het overzichtelijker en netter zijn als de files op een gestructureerde manier worden onderverdeeld in mapjes, zodat de reviewer of programmeur het makkelijk kan snappen om het uit te breiden of verbeteren.

Literatuurlijst