

Pflichtenheft Anticheat

Projekt Anticheat

| im Auftrag von Riot Games

Autoren:

Louis Linnemann, Marcel Kozik

Datum:

31.08.2022

Projekt:

Konzeption und Realisierung einer Anti-Cheat-Software

1. Problembeschreibung mit Ist-Zustand

2. Zielsetzung

2.1 Musskriterien

2.2 Sollkriterien

2.3 Kannkriterien

2.4 Abgrenzungskriterien

3. Produkteinsatz

3.1 Anwendungsbereich

3.2 Zielgruppen

4. Funktionale Anforderungen

4.1 System- und Softwarearchitektur

4.2 Klassendiagramm

5. Technische Produktumgebung

5.1 Hardware

5.2 Software

6. Lieferumfang

6.1 Zu übergebene Dokumente

6.2 Software-Bereitstellung

6.3 Schulungs- und Supportleistungen

7. Abnahmekriterien/Testfälle/Testplan

7.1 Qualitätsziele

7.2 Sicherheitstests

8. Glossar

1. Problembeschreibung mit Ist-Zustand

Ein Cheat ist eine Software, die die Spielumgebung in einer von den Spieleentwicklern nicht beabsichtigten Weise verändert, um den Spielern einen unfairen Zugang und Vorteil gegenüber anderen Spielern zu verschaffen.

Die Spieleindustrie ist im letzten Jahrzehnt erheblich gewachsen. Spiele, die jetzt auf dem Markt sind, bieten den Spielern einen Online-Spielmodus, in dem sie gegen andere Spieler antreten können (auch bekannt als PVP/Player vs Player), was vor fünf oder sieben Jahren nicht der Fall war, als es nur Einzelspieler-Spiele gab. Dies ist eine große Chance für Cheat-Entwickler, da die wettbewerbsorientierte Szene ein begehrter Ort für Spieler und sogar professionelle Spieler auf der ganzen Welt ist. Mit einer größeren Anzahl von Spielern in der wettbewerbsorientierten Szene, in der das "Geld zu holen" ist, ist sie nun ein attraktives Ziel für Leute, die versuchen, diese Spiele mit Cheats auszunutzen um illegale Gewinne zu erzielen.

Zurzeit hat Riot Games keine Möglichkeit sich vor Zugriff von Drittanbieter Programmen (sogenannten “Cheats”) zu schützen und diese zu erkennen.

2.Zielsetzung

Um das Aufkommen von Cheats zu bekämpfen, ist eine Anti-Cheat-Software unabdingbar. Es gibt viele Anti-Cheat-Software auf dem Markt, die auf kleine bis große Spiele abzielt. Allerdings sind EAC (EasyAntiCheat) und BE (BattlEye) (die bekanntesten Anti-Cheat-Softwares in der Branche) nicht in der Lage auf C# basierte Spiele zu unterstützen.

Darum haben wir uns als Ziel genommen, auch C# basierte Spiele zu unterstützen und diese ordnungsgemäß vor Zugriff von unerlaubten Drittanbieter Programmen zu schützen.

2.1 Musskriterien

- Ermöglichung einer einfachen Implementierung seitens der Entwickler
- Automatisiertes starten der Anticheat-Software bei integrierung
- Absicherung des Spiels, dass unsere Softwarelösung implementiert
 - Abblocken von DLL Injection
 - Erkennen und verhindern von externen Debuggern
- Erweiterbar anhand einer Schnittstelle

2.2 Sollkriterien

- Verhindern des Spielstarts bei nicht erfolgreichem Start des Anti-Cheats
- Überprüfen von unerlaubten externen Zugriff auf den Speicher des Spiels

2.3 Kannkriterien

- Unterstützung für Native(in C/C++ entwickelte) Spiele

2.4 Abgrenzungskriterien

- Das System soll keine grafische Bedienoberfläche haben.

3. Produkteinsatz

3.1 Anwendungsbereich

Die Anti-Cheat-Lösung dient zur Erkennung von unerwünschten Drittanbieterprogrammen.

Entwickler können unsere Lösung implementieren und diese so in ihren eigenen Spielen integrieren.

3.2 Zielgruppen

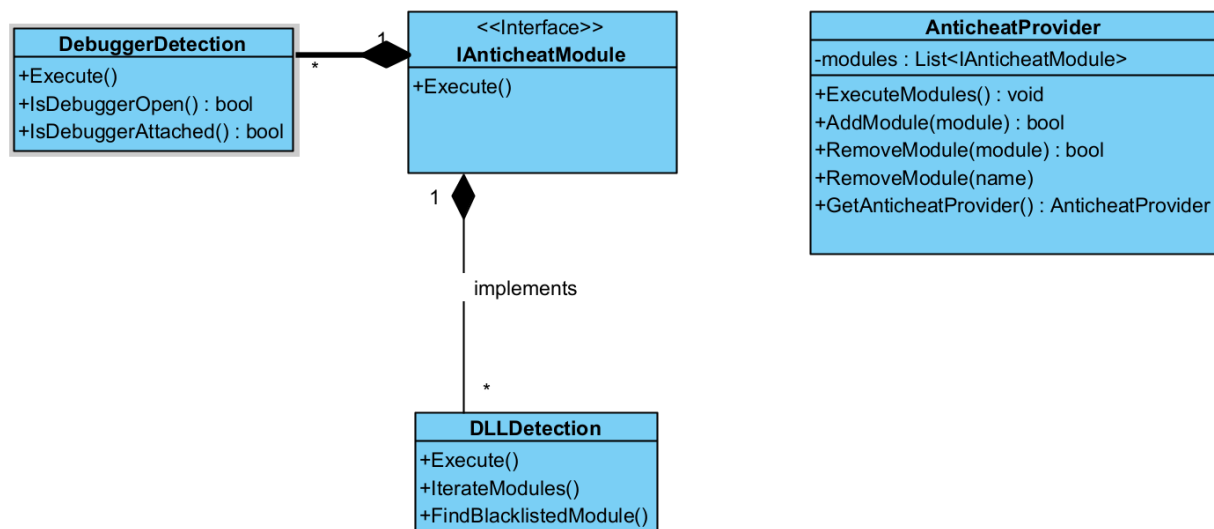
Unsere Software ist darauf abgestimmt von Spieleentwicklern einfach in bereits bestehende und zukünftige Spiele implementiert zu werden.

4. Funktionale Anforderungen

4.1 System- und Softwarearchitektur

Für die Entwicklung unserer Softwarelösung nutzen wir C# unter .NET 6.

4.2 Klassendiagramm



5. Technische Produktumgebung

5.1 Hardware

Entwickelt wird unter Windows 10 und Windows 11 (jeweils die neueste Version) um die passende Kompatibilität zu gewährleisten.

5.2 Software

Zur Entwicklung wird Visual Studio 2022 mit installiertem .NET 6 support genutzt.

6. Lieferumfang

6.1 Zu übergebene Dokumente

Übergeben wird eine Dokumentation, die die Implementierung unseres Systems in das System des Kunden beschreiben und für den Leser nachvollziehbar machen.

6.2 Software-Bereitstellung

Die Software wird als Klassenbibliothek zur Verfügung gestellt.

7. Abnahmekriterien

7.1 Qualitätsziele

Die mitgelieferte Klassenbibliothek soll einfach implementierbar, dokumentiert und anhand einer Schnittstelle beliebig erweiterbar sein.

7.2 Sicherheitstests

Die Grundfunktionen unseres Systems kann anhand von verschiedenen Angriffsmethoden die üblicherweise seitens der Angreifer/Cheat-Entwickler genutzt werden, anhand eines Penetrationstests getestet werden.

Derartige Tests sollten durch Experten dieses Gebiets durchgeführt werden.

8. Glossar

Begriff	Beschreibung
DLL Injection	Das hinzufügen einer DLL Datei in einen Prozess.
Penetrations-Test	Das Testen eines Systems durch Angriffe.