Kapitel 1

Überprüfung der Anforderungen

In diesem Kapitel wird geklärt, in wie fern das entwickelte System den zu Beginn des Projekts aufgestellten funktionalen und nichtfunktionalen Anforderungen gerecht wird. Dafür wird zunächst kurz auf deren Priorisierung eingangen. Anschließend werden die Anforderungen aufgelistet und kurz erklärt, ob diese erfüllt worden oder nicht.

1.1 Priorisierung der Anforderungen

Um Anforderungen zu strukturieren und nach Wichtigkeit zu priorisieren, wird in der Regel ein System zur Klassifizierung der Eigenschaften verwendet. Hier wurde eine Priorisierung nach der **MuSCoW**-Methode vorgenommen:

Must: Diese Anforderungen sind unbedingt erforderlich und nicht verhandelbar. Sie sind erfolgskritisch für das Projekt.

Should: Diese Anforderungen sollten umgesetzt werden, wenn alle Must-Anforderungen trotzdem erfüllt werden können.

Could: Diese Anforderungen können umgesetzt werden, wenn die Must- und Should-Anforderungen nicht beeinträchtigt werden. Sie haben geringe Relevanz und sind eher ein „Nice to have“.

Won't: Diese Anforderungen werden im Projekt nicht explizit umgesetzt, werden aber eventuell für die Zukunft vorgemerkt.

1.2 Funktionale Anforderungen

Die funktionalen und die nichtfunktionalen werden in einzelnen Unterkapiteln getrennt behandelt. Für beide Arten wird zunächst die Tabelle aus dem Pflichtenheft erneut dargestellt. Nach der Auflistung wird eine Überprüfung zum einen anhand des Testdrehbuch und nach anderen Methoden vorgenommen.

1.2.1 Auflistung der funktionalen Anforderungen

Funktionale Anforderungen legen konkret fest, was das System können soll. Hier wird unter anderem beschrieben, welche Funktionen das System bieten soll. Die folgende Tabelle zeigt diese

funktionalen Anforderungen.

ID	Name	Beschreibung	MuSCoW
F01	Lokale Administration	Das System muss lokal per Command-Line-Interface administriert werden können.	Must
F02	Angriffsarten	Das System muss die Folgen der aufgelisteten (D)DoS-Angriffe abmildern können: • SYN-Flood • SYN-FIN Attack • SYN-FIN-ACK Attack • TCP-Small-Window Attack • TCP-Zero-Window Attack • UDP-Flood Dabei ist vorausgesetzt, dass das Ziel eines Angriffes eine einzelne Station in einem Netzwerk ist und kein Netzwerk von Stationen. Es sind also direkte Angriffe auf einzelne Server, Router, PC, etc. gemeint.	Must
F03	Keine zusätzliche Angriffsfläche	Besonders darf das System den unter „Angriffsarten" spezifizierten Angriffen keine zusätzliche Angriffsfläche bieten, d.h. es darf es auch nicht durch Kenntnis der Implementierungsdetails möglich sein, das System mit diesen Angriffen zu umgehen.	Must
F04	L3/ L4 Protokolle	Das System muss mit gängigen L3/ L4 Protokollen klarkommen.	Must
F05	Modi	Passend zum festgestellten Angriffsmuster muss das System eine passende Abwehrstrategie auswählen und ausführen.	Must
F06	Position	Das System soll zwischen dem Internet-Uplink und dem zu schützenden System oder einer Menge von Systemen platziert werden.	Must
F07	Weiterleiten von Paketen	Das System muss legitime Pakete vom externen Netz zum Zielsystem weiterleiten können.	Must
F08	Installation und Deinstallation	Das System muss durch Befehle in der Kommandozeile zu installieren und zu deinstallieren sein. Hilfsmittel hierzu sind: Installationsanleitung, Installationsskript, Meson und Ninja.	Must
F09	Mehrere Angriffe nacheinander und zeitgleich	Das System muss mehreren Angriffen nacheinander und zeitgleich standhalten, hierbei muss berücksichtigt werden, dass auch verschiedene Angriffsarten und Muster zur gleichen Zeit erkannt und abgewehrt werden müssen.	Must
F10	IPv4	Das System muss mit IPv4-Verkehr zurechtkommen.	Must

ID	Name	Beschreibung	MuSCoW
F11	Hardware	Das System soll nicht Geräte- bzw. Rechner-spezifisch sein.	Should
F12	Zugriff	Der Zugriff auf das lokale System soll per SSH oder Ähnlichem erfolgen, um eine Konfiguration ohne Monitor zu ermöglichen.	Should
F13	Betrieb	Das System soll auf Dauerbetrieb ohne Neustart ausgelegt sein.	Should
F14	Privacy	Das System soll keine Informationen aus der Nutzlast der ihm übergebenen Pakete lesen oder verändern.	Should
F15	Konfiguration	Der Administrator soll die Konfiguration mittels Konfigurationsdateien ändern können.	Can
F16	Abrufen der Statistik	Der Administrator soll Statistiken über das Verhalten des Systems abrufen können.	Can
F17	Starten und Stoppen des Systems	Der Administrator soll das System starten und stoppen können.	Can
F18	Informieren des Anwenders	Der Anwender soll über Angriffe informiert werden.	Can
F19	Administration über graphische Oberfläche	Das System soll über eine graphische Oberfläche administriert werden können.	Can
F20	IPv6	Das System soll mit IPv6-Verkehr zurechtkommen können	Can
F21	Weitere Angriffsarten	Das schützt weder vor anderen außer den genannten DoS-Angriffen (siehe F02 „Angriffsarten“)-insbesondere nicht vor denjenigen, welche auf Anwendungsebene agieren-, noch vor anderen Arten von Cyber-Attacken, die nicht mit DoS in Verbindung stehen. So bleibt ein Intrusion Detection System weiterhin unerlässlich.	Won’t
F22	Anzahl der zu schützenden Systeme	Das System wird nicht mehr als einen Server, Router, PC, etc. vor Angriffen schützen.	Won’t
F23	Fehler des Benutzers	Das System soll nicht vor Fehlern geschützt sein, da es durch eine nutzungsberechtigte Person am System ausgeführt wird. So sollen beispielsweise Gefährdungen, welche aus fahrlässigem Umgang des Administrators mit sicherheitsrelevanten Softwareupdates resultieren, durch das zu entwickelnde System nicht abgewehrt werden.	Won’t
F24	Softwareupdates	Das System soll keine Softwareupdates erhalten und soll nicht gewartet werden.	Won’t

ID	Name	Beschreibung	MuSCoW
F25	Router-/Firewall-Ersatz	Das System soll nicht als Router oder als Firewall-Ersatz verwendet werden.	Won't
F26	Hardware-Ausfälle	Das System soll keine Hardwareausfälle (zum Beispiel auf den Links) beheben.	Won't
F27	Fehler in Fremdsoftware	Das System kann nicht den Schutz des Servers bei Fehlern in Fremdsoftware garantieren.	Won't

1.2.2 Überprüfung der funktionalen Anforderungen

Jede einzelne funktionale Anforderungen wird hier unter einer eigenen kleinen Überschrift überprüft. Die Reihenfolge ist dabei die gleiche wie in der oben stehenden Tabelle.

1.2.2.1 F01: Lokale Administration

Diese Muss-Anforderung wurde erfüllt. Ein Command-Line-Interace wurde mithilfe von Konventionen wie „Human first design" oder der Forderung, dass das Programm bei der Benutzung unterstützt. So soll es beispielsweise vorschlagen, was als Nächstes gemacht werden kann.

Mit Hilfe der Kommandos „help" oder „h" bekommt der Nutzer alle gültigen Eingaben angezeigt.

Die Eingabe von „exit"

1.2.2.2 F02: Angriffsarten

1.2.2.3 F03: Keine zusätzliche Angriffsfläche

1.2.2.4 F04: L3/L4 Protokolle

Sowohl Protokoll L3, zuständig für die Vermittlung von Daten über einzelne Verbindungsanschnitte und Netzwerkknoten und Adressierung der Kommunikationspartner, als auch Protokoll L4, das die Quell- und Zieladressen und weitere Informationen des Pakets enthält, werden vom System akzeptiert und verwendet.

1.2.2.5 F05: Modi

Das System erkennt und unterscheidet verschiedene Angriffsmethoden und errechnet selbst die passende Abwehrstrategie sowie eine Anpassung der Durchlassrate von Paketen.

1.2.2.6 F06: Position

«««< HEAD Das System ist vorgesehen zwischen zwei Systemen, die wiederrum auch eine Menge von Systemen sein können, jedoch beide klar voneinander getrennt als `Externes` und `Internes` System. Die Unterscheidung der externen/internen Systeme erfolgt über die Zuweisung der Ports der Netzwerkkarte. ======= In Kapitel 2 ist auf Seite **??** in Abbildung **??** der Versuchsaufbau zu sehen. Das System wurde im Labor im Zusebau der TU Ilmenau auch tatsächlich in dieser Reihenfolge aufgebaut. Damit ist diese Anforderung erfüllt. »»»> 81217d8ba845561661ce8bf4404094b3c2178fb9

1.2.2.7 F07: Weiterleiten von Paketen

Zur Überprüfung dieser Anforderung ist der erste Test des in der Planungs- und Entwurfsphase geschriebenen Testdrehbuchs gedacht. Für den Test der Paketweiterleitung werden zunächst Pakete mit DPDK von einem Port der Netzwerkkarte entgegengenommen und auf den andern Port weitergegeben. Danach wurde begonnen, einzelne Ping-Anfragen vom äußeren System über die Mitigation-Box zum Server laufen zu lassen. Im Anschluss wurde ein Lasttest durchgeführt. Diese Tests haben ergeben, dass...

1.2.2.8 F08: Installation und Deinstallation

Das System wird mit einer Installationsanleitung und Installationsskripten ausgeliefert. Das Installationsskript für abhängige und notwendige Systemeinstellungen und Programme installiert alle notwendigen zusätzlichen Programme und Bibiliotheken und nimmt alle notwendigen Systemeinstellungen vor, soweit möglich. Bei Fehlern wird dem Benutzer ein Hinweis zur Lösung des Problems angezeigt. Die Installation kann auch selbst mit der Installationsanleitung vorgenommen werden, die einzelnen Schritte sind in eigenen Unterkapiteln genauer erklärt. Für einen lokalen Bau der Software kann `Meson` und `Ninja` verwendet werden, deren Benutzung in der die Installationsanleitung fortführenden Seite `Usage` erklärt wird. Dort ist auch ein kurzer Einstieg in die Benutzung von `AEGIS` erklärt. Zur systemweiten Installation von `AEGIS` kann ebenfalls `Meson` genutzt werden, das durch ein Installationsskript erweitert wurde. Das gesamte Programm und zusätzlich installierten Programme können mit einem Deinstallationsskript wieder vom System gelöscht werden.

1.2.2.9 F09: Mehrere Angriffe nacheinander und zeitgleich

1.2.2.10 F10: IPv4

1.2.2.11 F11: Hardware

«««< HEAD

1.2.2.12 F12: Zugriff

======= Diese Should-Anforderung wurde mit dem entwickelten System nicht erfüllt. Die Software läuft im derzeitigen Zustand ausschließlich auf dem Testbed im Rechenlabor. Durch kleine Anpassung kann die Nutzung auf alternativer Hardware allerdings ermöglicht werden.

1.2.2.13 F12: Zugriff

Es ist ein ssh-Zugriff auf das System möglich. Die Konfiguration ermöglicht die Authentifizierung nicht mit einem Passwort möglich ist. Mithilfe des Befehls `ssh-keygen` kann ein Schlüsselpaar generiert werden und ein Public-Key ist in einer Textdatei zu finden. Weiterhin ist es möglich, zu überprüfen, welche anderen Teammitglieder derzeit auf diese Art mit dem System verbunden sind. »»»> 81217d8ba845561661ce8bf4404094b3c2178fb9

1.2.2.14 F13: Betrieb

1.2.2.15 F14: Privacy

1.2.2.16 F15: Konfiguration

Der Administrator kann die Einstellungen von `AEGIS` durch zwei Konfigurationsdateien anpassen. Innerhalb der `meson_options.txt` Datei kann der Bau von Unit Tests und der Dokumentation ein- und ausgeschalten werden. In der Datei `config.json` können verschiedene Einstellungen wie die zu verwendenden Systemkerne und Durchlassraten eingestellt werden.

1.2.2.17 F16: Abrufen der Statistiken

1.2.2.18 F17: Starten und Stoppen des Systems

Das Programm `AEGIS` kann über ein Terminal vom Benutzer gestartet und gestoppt werden. Das CLI unterstützt den Nutzer dabei mit Hinweisen.

1.2.2.19 F18: Informieren über graphische Oberfläche

1.2.2.20 F19: Administration über grafische Oberfäche

Diese Can-Anforderung wurde aus Zeitgründen nicht umgesetzt. Die Administration erfolgt stattdessen über das in F01 beschriebene CLI.

1.2.2.21 F20: IPv6

Das System ist vorerst auf IPv4 ausgelegt und funktionsfähig aber schon darauf ausgerichtet auch IPv6 zu unterstützen.

1.2.2.22 F21: Weitere Angriffsarten

Bei den Anforderungen F21 - F27 handelt es sich solche, die mit Won't priorisiert wurden. Das heißt, dass sie nicht erfüllt werden dürfen. Bei dem während des Softwareprojekts entwickelten System gibt es auch keine Anzeichen dafür, dass es ungewollter Weise vor anderen Gefahren schützt.

1.2.2.23 F22: Anzahl der zu schützende Systeme

Das System schützt nur ein einzelnes System. Mit moderaten Änderungen kann die Software aber auf anderen Servern etc. installiert werden und dadurch mehrere schützen. Aus hardwareseitigen Gründen wurde dies aber auch nicht getestet.

1.2.2.24 F23: Fehler des Benutzers

Auch nach der Entwicklung kann festgehalten werden, dass das System durch einen kompetenten Administrator installiert und genutzt werden muss.

1.2.2.25 F24: Softwareupdates

Das System, das während des Projekts entwickelt wurde, wird nach Abschluss der Lehrveranstaltung nicht direkt weiterentwickelt. Während der Erstellung dieses Dokuments wird allerdings

davon ausgegangen, dass einige Studierende der Gruppe auch nach Abschluss der Lehrveranstaltung evtl. noch an dem System weiterarbeiten. Features, die möglicherweise dadurch noch hinzugefügt werden, können allerdings nicht als Softwareupdates angesehen werden.

1.2.2.26 F25: Router-/Firewall-Ersatz

Auch diese Won't-Anforderung wurde nicht erfüllt. Eine Firewall oder vergleichbare schützende Systeme bleiben nach wie vor unerlässlich.

1.2.2.27 F26: Hardware-Ausfälle

Aegis kann keine Hardwareausfälle beheben.

1.2.2.28 F27: Fehler in Fremdsoftware

Ebenso gibt keine Anzeichen dafür, dass das System vor Fehlern irgendeiner anderen Software schützen kann.

1.3 Nichtfunktionale Anforderungen

Auch hier wird genau wie bei den funktionalen Anforderungen vorgegangen.

1.3.1 Auflistung der nichtfunktionalen Anforderungen

Nichtfunktionale Anforderungen gehen über die funktionalen Anforderungen hinaus und beschreiben, wie gut das System eine Funktion erfüllt. Hier sind zum Beispiel Messgrößen enthalten, die das System einhalten soll. Im folgenden werden diese nichtfunktionalen Anforderungen beschrieben.

ID	Name	Beschreibung	MuSCoW
NF01	Betriebssystem	Die entwickelte Software muss auf einer Ubuntu 20.04 LTS Installation laufen. DPDK muss in Version 20.11.1 vorliegen und alle Abhängigkeiten erfüllt sein.	Must
NF02	Verfügbarkeit	Die Verfügbarkeit des Systems soll bei mindestens 98% liegen. Verfügbarkeit heißt hier, dass das System in der Lage ist, auf legitime Verbindungsanfragen innerhalb von 10 ms zu reagieren.	Must
NF03	Datenrate	Die anvisierte Datenrate, welche vom externen Netz durch das zu entwickelnde System fließt, muss bei mindestens 20 Gbit/s liegen.	Must
NF04	Paketrate	Die anvisierte Paketrate, welche vom zu entwickelnden System verarbeitet werden muss, muss bei mindestens 30 Mpps liegen.	Must
NF05	Transparenz	Der Anwender soll das Gefühl haben, dass die Middlebox nicht vorhanden ist.	Should

ID	Name	Beschreibung	MuSCoW
NF06	Abwehrrate SYN-Flood	Die für die Angriffe anvisierten Abwehrraten sind für die SYN-Flood, SYN-FIN und SYN-FIN-ACK jeweils 100%.	Should
NF07	False Positive	Der maximale Anteil an fälschlicherweise nicht herausgefiltertem und nicht verworfenem illegitimen Traffic, bezogen auf das Aufkommen an legitimem Traffic, soll 10% im Angriffsfall und 5% im Nicht-Angriffsfall nicht überschreiten.	Should
NF08	False Negative	Der maximale Anteil an fälschlicherweise nicht verworfenem bösartigem Traffic, bezogen auf das Gesamtaufkommen an bösartigem Traffic, soll 5% nicht überschreiten.	Should
NF09	Round Trip Time	Die Software soll die Round-Trip-Time eines Pakets um nicht mehr als 10 ms erhöhen.	Should

1.3.2 Überprüfung der nichtfunktionalen Anforderungen

Auch bei den nichtfunktionalen Anforderungen werden jeweils einzelne Unterkapitel genutzt.

1.3.2.1 NF01: Betriebssystem

Die genannte Software, also Ubuntu 20.04 LTS und DPDK 20.11.1, wurde von allen Teammitgliedern installiert. Schließlich wurde auch nur mit diesen Versionen des Betriebssystems bzw. Frameworks entwickelt und getestet. Es kann also dokumentiert werden, dass das System unter diesen Voraussetzungen wie bei den anderen Tests beschrieben funktioniert. Es kann keine Aussage darüber getroffen werden, inwiefern das System unter anderen Versionen funktioniert.

1.3.2.2 NF02: Verfügbarkeit

1.3.2.3 NF03: Datenrate

1.3.2.4 NF04: Paketrate

1.3.2.5 NF05: Transparenz

1.3.2.6 NF06: Abwehrrate SYN-Flood

1.3.2.7 NF07: False Positive

1.3.2.8 NF08: False Negative

1.3.2.9 NF09: Round Trip Time