Abwehr von Denial-of-Service-Angriffen durch effiziente User-Space Paketverarbeitung: AEGIS

Review für die Implementierungsphase

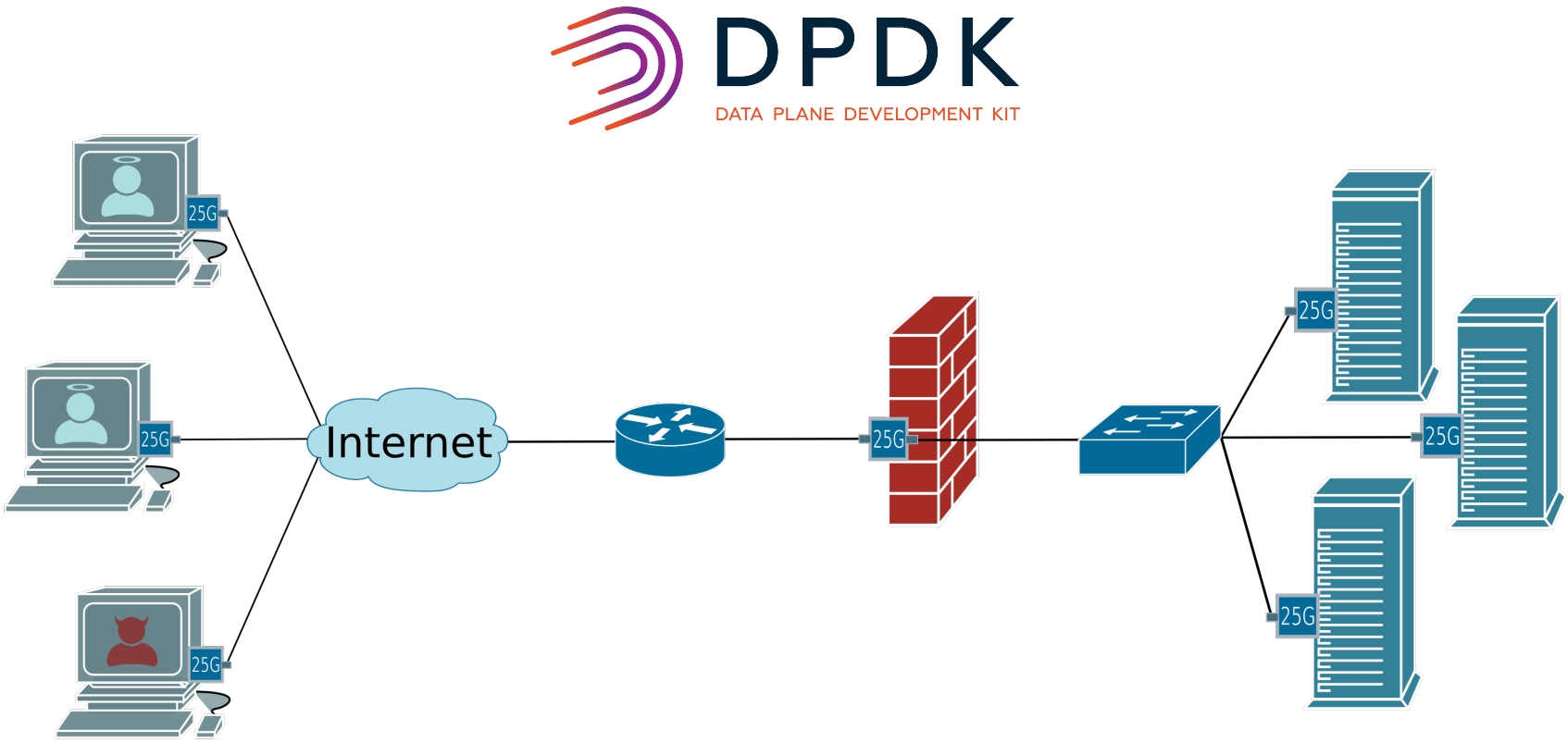
Johannes Lang, Jakob Lerch

24.06.2021

Technische Universität Ilmenau

AEGIS

Aufgabenstellung



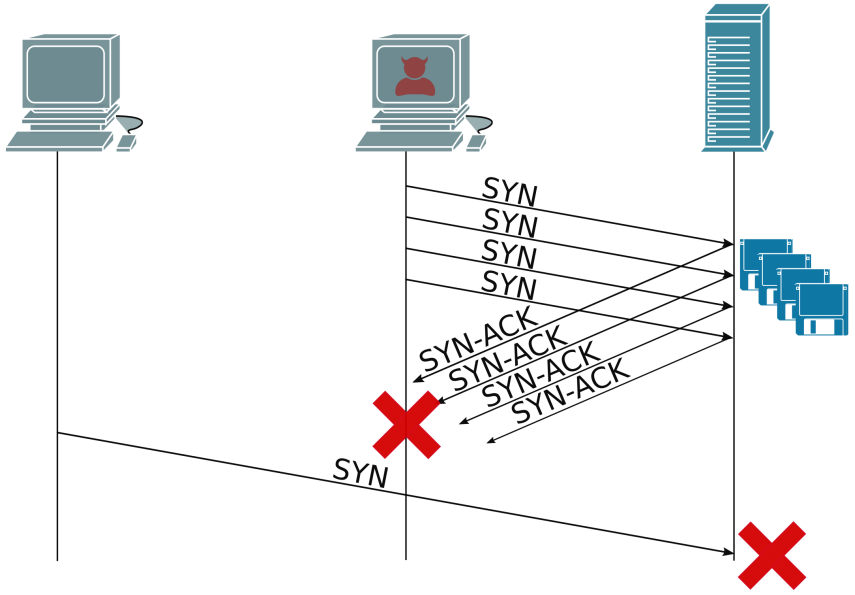
Abwehrsystem gegen DoS-Angriffe

EGIS

Aufgabenstellung

- Die Software soll mehrere Varianten von Attacken abwehren
- Nur eine davon ist für diesen Vortrag relevant:

SYN-Flood-Attacke



Gliederung

EGIS

Grobentwurf

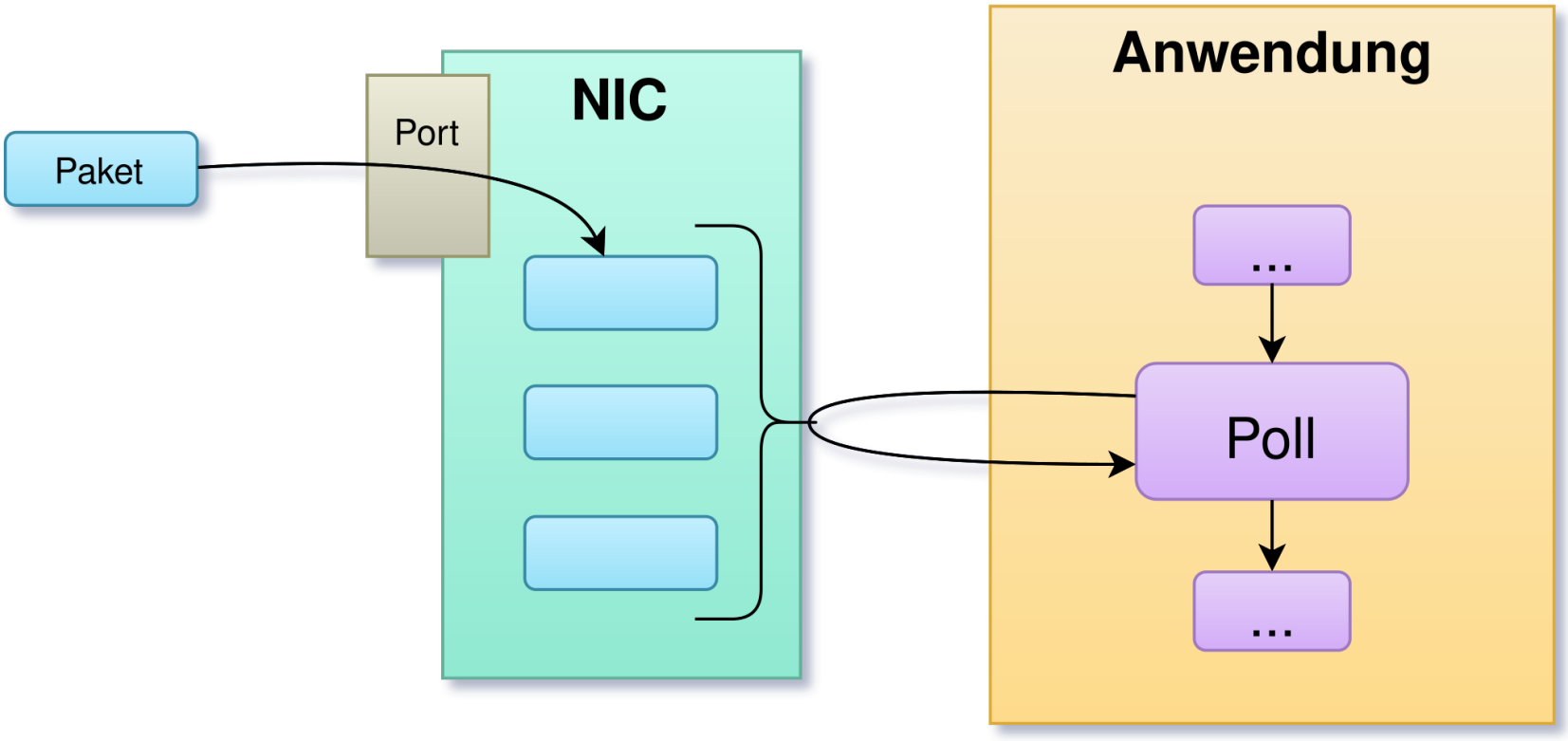
Grobentwurf

Die Architektur folgt dem Pipeline-Modell.

EGIS

Feinentwurf

Feinentwurf: NicManagement



effizient Pakete von der NIC bekommen: Polling

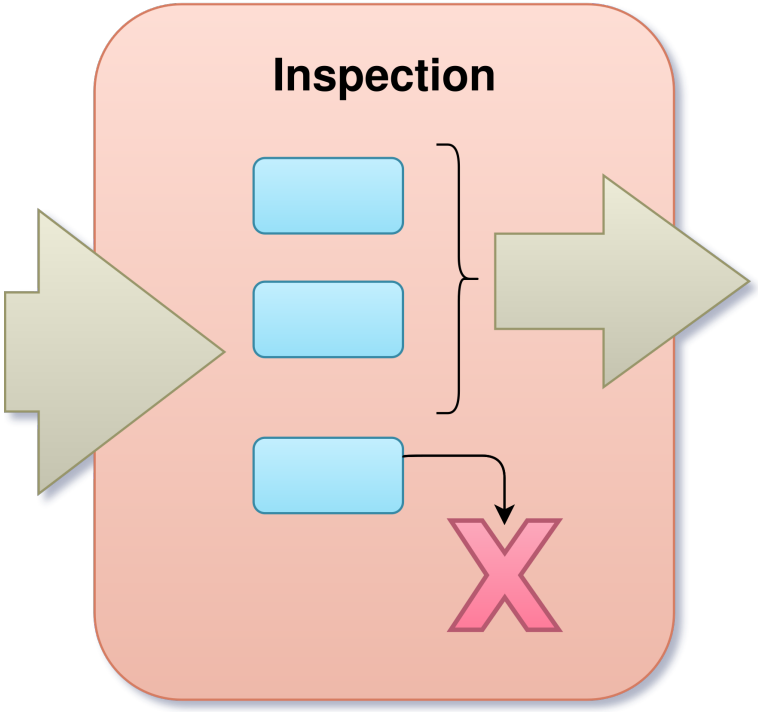
Feinentwurf: PacketDissection

Feinentwurf: PacketDissection

- extrahiert Informationen aus den Paketen
- stellt diese für die folgenden Komponenten bereit

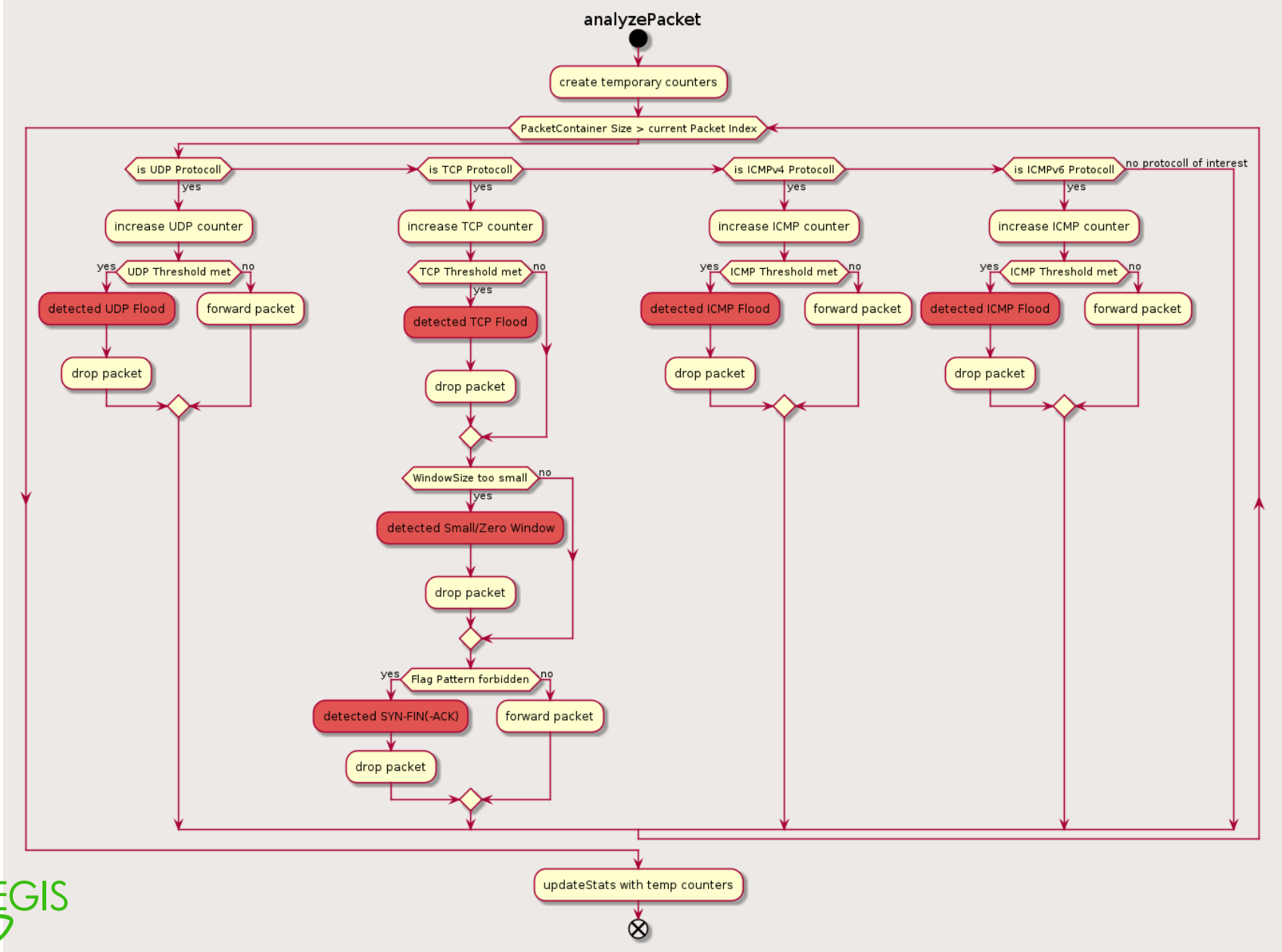
Feinentwurf: Inspection

Feinentwurf: Inspection



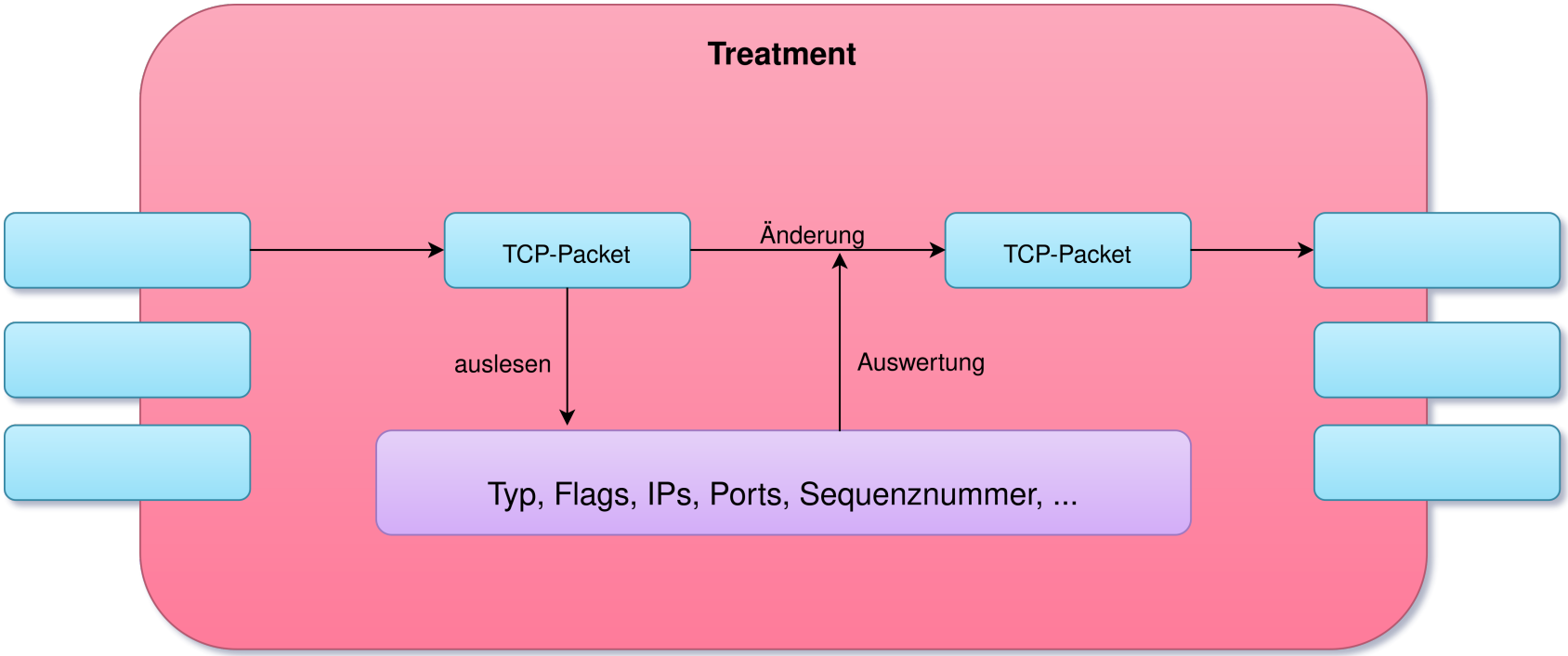
- Klasse Analyzer
- Filterung aller Pakete der Netzwerkprotokolle UDP, TCP, ICMP
- Abwehr von SYN-FIN-Angriffen

Feinentwurf: Inspection

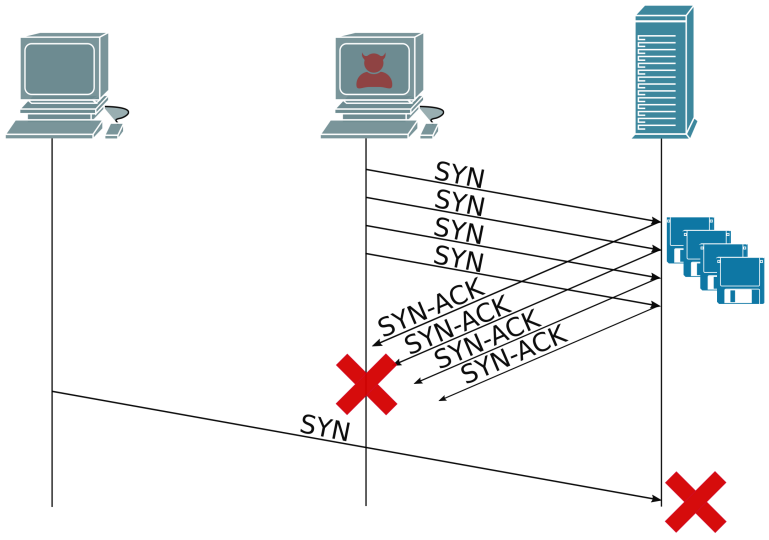


Feinentwurf: Treatment

Feinentwurf: Treatment

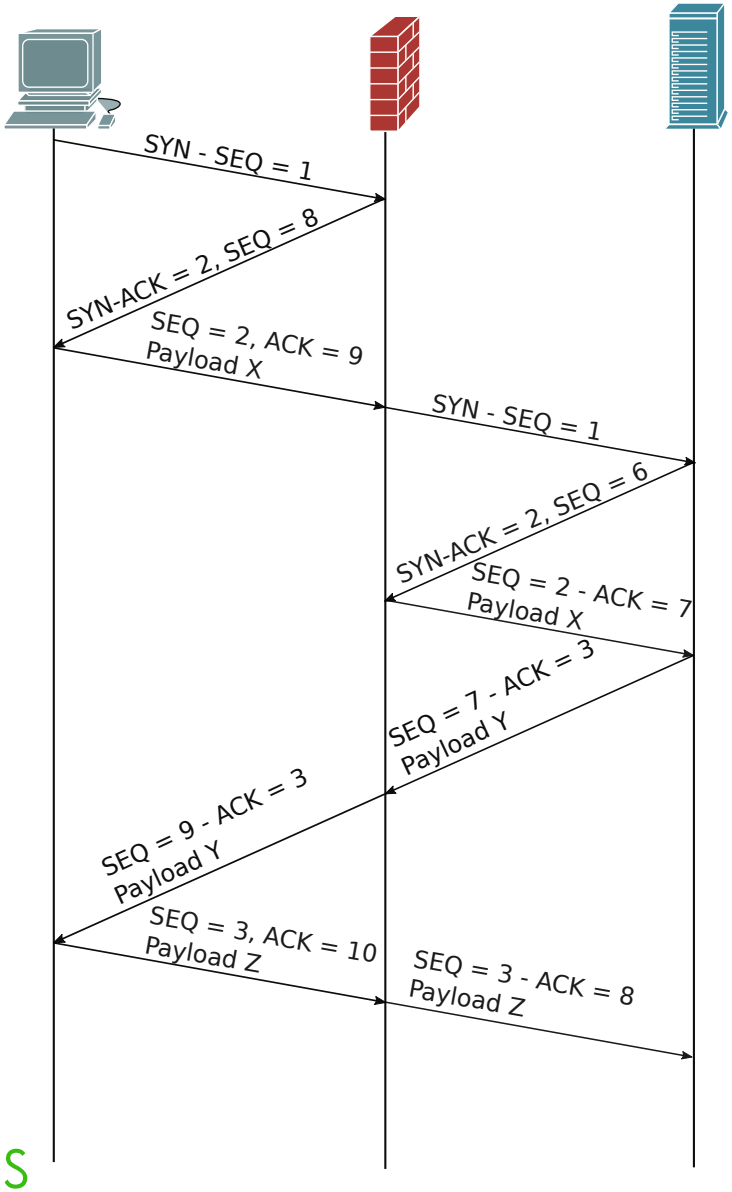


Feinentwurf: Treatment



- SYN-Flood-Abwehr mit SYN-Cookies
- keine Reservierung von Ressourcen beim Aufbau

Feinentwurf: Treatment



- TCP-Proxy
- Middle-Box als Vermittler

Implementierung Treatment

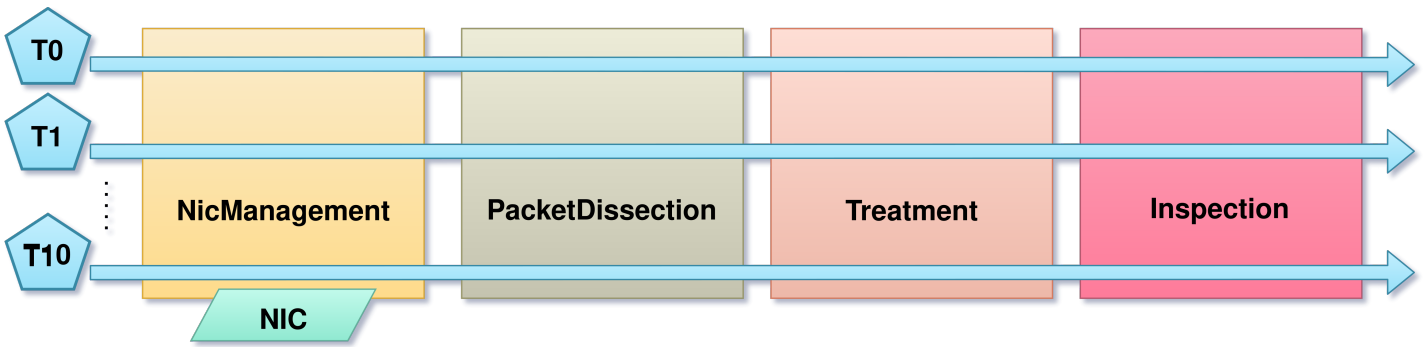
```
Treatment::treat_packtes(){
  for packet in packet_to_inside{
    if(packet.get_type() == packet_type_tcp){
      flags = packet.get_flags();
        if(flags.is_pure_syn()){
          syn_cookie = calc_cookie(connection_data);
          reply_packet = get_empty_packet_to_outside;
          reply_packet.fill(connection_data,syn_cookie);
        }
        else if(...){...}
        ...
    }
  }
}
```

Feinentwurf: Einsatz von Threads

- eine Pipeline $\rightarrow$ nur ein Thread nötig

Feinentwurf: Einsatz von Threads

- Wünsche für Effizienz:
  - mehrere Threads parallel
  - gleichmäßig ausgelastet
  - keine Kommunikation



Feinentwurf: Einsatz von Threads

- Pakete aufgeteilt durch „RSS” (Receive Side Scaling)
  - realisiert durch Hashing
  - Schlüssel: `[Src-IP; Dst-IP; Src-Port; Dst-Port]`

- **gleichmäßige Auslastung** (wegen Hashing)

EGIS

Feinentwurf: Einsatz von Threads

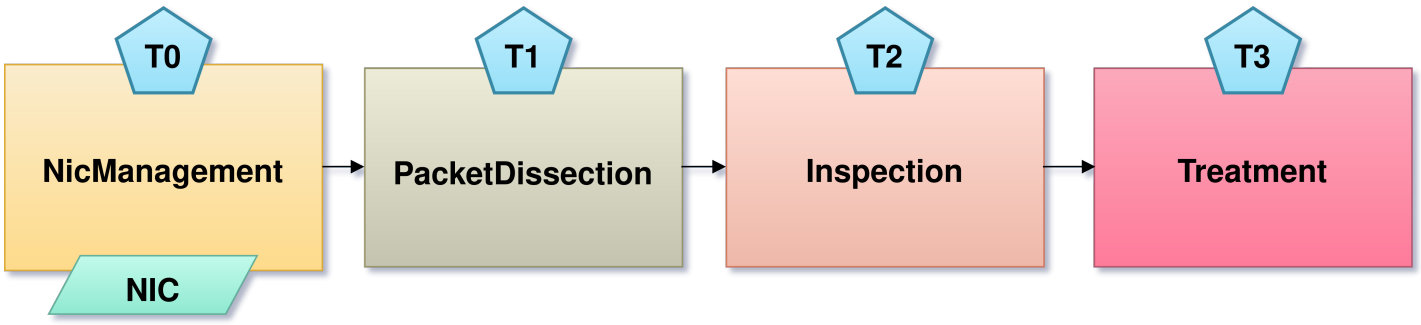
- Problem: Verschiedene Zuordnung je Seite
  $\rightarrow$ Inter-Thread-Kommunikation nötig!

Feinentwurf: Einsatz von Threads

- Lösung: „Symmetric RSS"
  - [Src-IP; *Dst-IP*] $\equiv$ [*Dst-IP*; Src-IP]

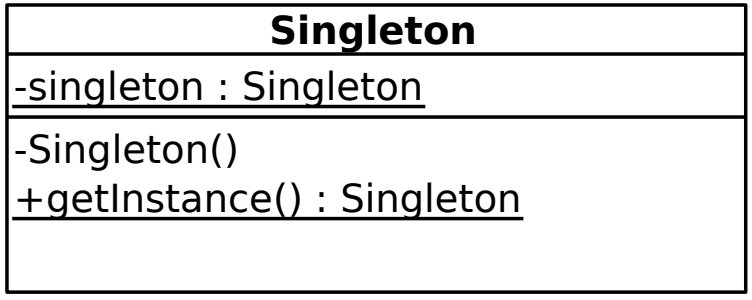
  $\rightarrow$ **keine Inter-Thread-Kommunikation nötig**

Feinentwurf: Alternative Entwürfe



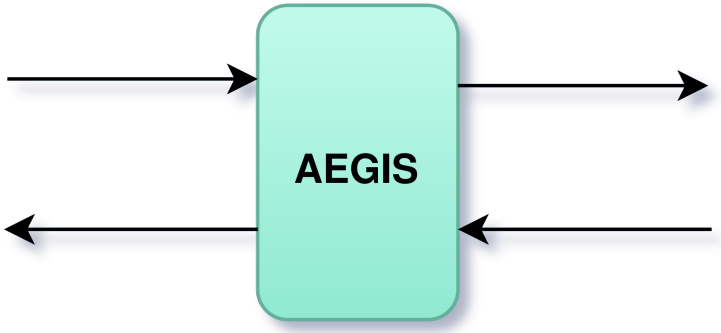
- alternativ: ein Thread pro Komponente
- Nachteil: zu viel Inter-Thread-Kommunikation

Entwurfsmuster



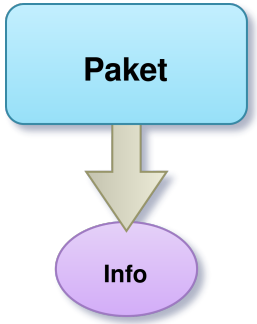
- Erzeugungsmuster
- Nur ein Objekt dieser Klasse
- Globale Informationsbereitstellung
- Verwendung im Configurator

Was AEGIS bisher kann



Multithreading

Pakete weiterleiten



Informationen aus Paketen extrahieren

Ausblick

- Anforderungen unverändert
- Überprüfung wichtiger Anforderung
- Erweiterung um Angriffe und ihre Abwehrmechanismen

Bildquellen

- https://www.onlinewebfonts.com/icon/571002 [Abgerufen am 22.06.2021]

Vielen Dank für Ihre Aufmerksamkeit!