Sondervorlesung „Netz-Sicherheit“

IDS Evasion: Eine technische Einführung

Tillmann Werner, Edwin Reusch

CERT-Bund, Bundesamt für Sicherheit in der Informationstechnik

Universität Bonn, 4. Juli 2007

BSI

- Bundesamt für Sicherheit in der Informationstechnik, Bonn
- Drei Fachabteilungen + Verwaltungsabteilung
- Abteilung 1: „Sicherheit in Anwendungen, KRITIS und im Internet“
- Referat 121 - CERT-Bund, „Computer-Notfallteam für Bundesbehörden“

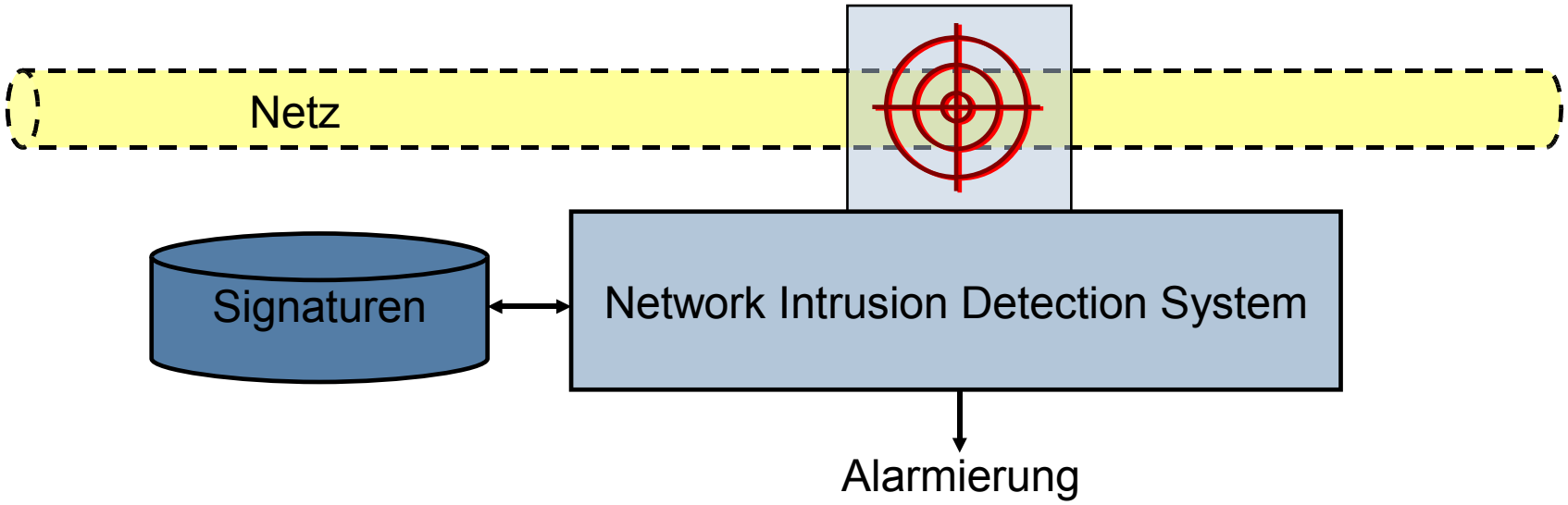
- Aufgaben von CERT-Bund
  - Veröffentlichen von Advisories zu aktuellen Schwachstellen
  - Monitoring und Bewertung der Bedrohungslage im Internet
  - Incident Handling bei sicherheitskritischen Vorfällen (im Bereich Bund)
  - Anlassbezogene Analyse von Schadprogrammen und Angriffen
  - Point-of-Contact für internationale CERTs

Agenda

- Einleitung – Signaturbasierte Network Intrusion Detection Systems
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

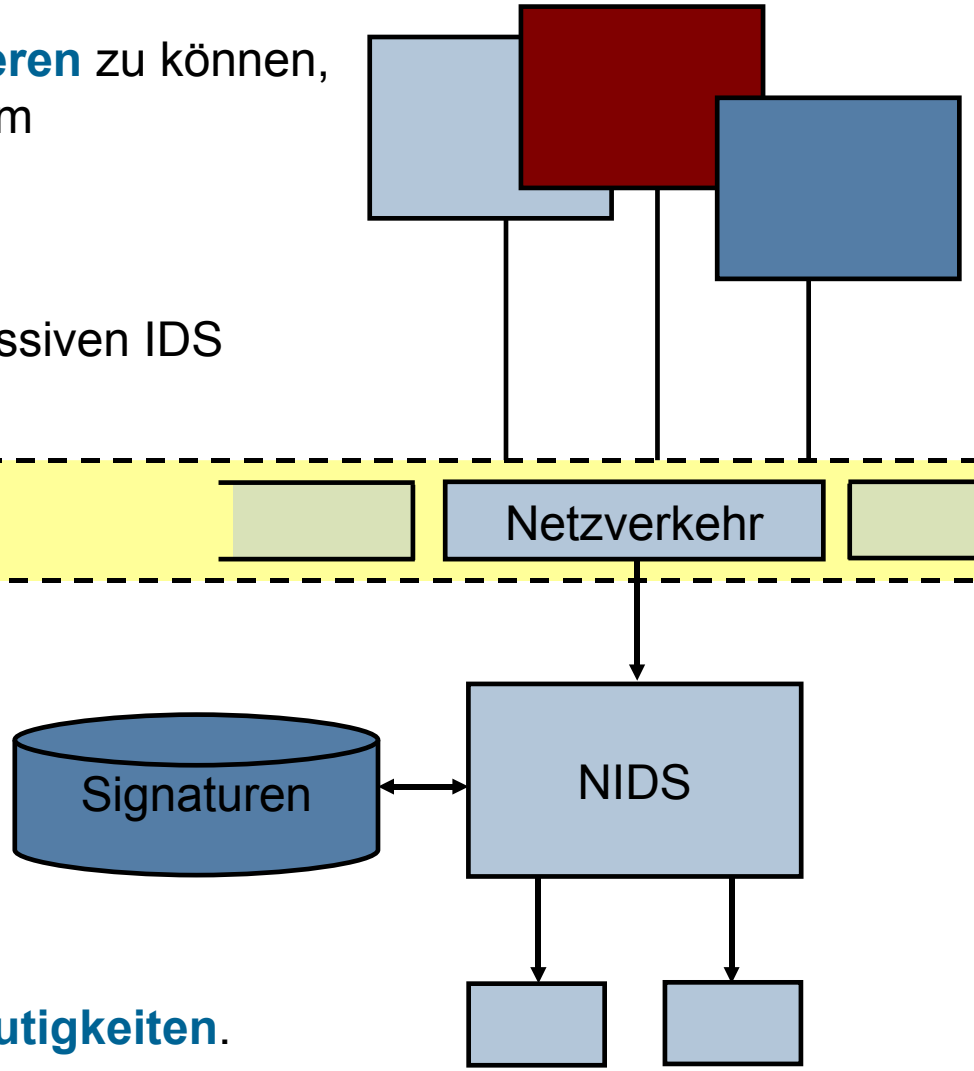
Überblick: Network Intrusion Detection Systems

- Ein **Network Intrusion Detection System** ist eine spezielle, netzwerkfähige Komponente zur **Feststellung von Angriffen gegen IT-Systeme**.
- Dazu wird der **Netzverkehr passiv mitgelesen** und anhand hinterlegter **Signaturen** auf bestimmte Eigenschaften hin untersucht.
- Wird eine solche Signatur **getriggert**, kann ein **Alarmierungsprozess** angestoßen werden.



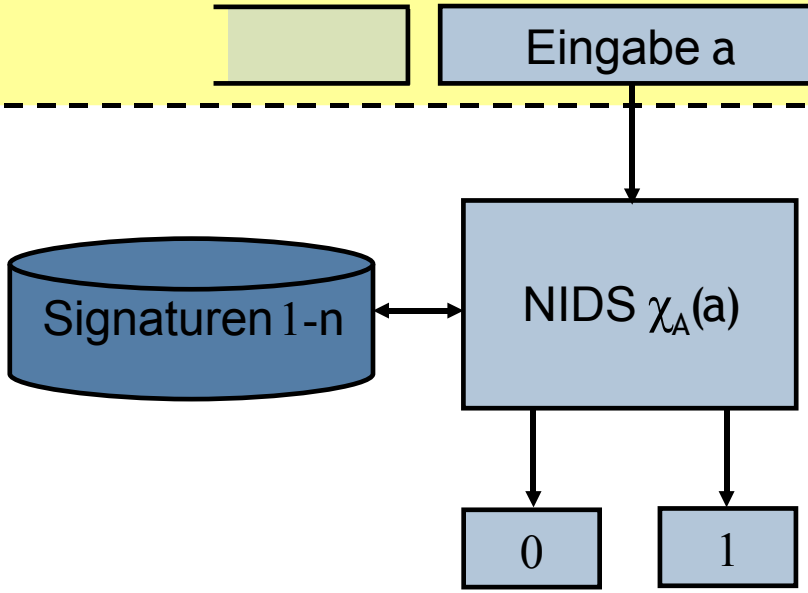
Praktische Grenzen signaturbasierter IDS

- Um Netzverkehr **korrekt klassifizieren** zu können, muss ein IDS wissen, wie dieser vom **Zielsystem interpretiert** wird.
- In größeren Netzen ist dies dem passiven IDS **praktisch unmöglich**.
- **Verschiedene Betriebssysteme** verhalten sich unterschiedlich.
- Ohne Detailkenntnis über das Netz entstehen so zwangsläufig **Mehrdeutigkeiten**.



Theoretische Grenze signaturbasierter IDS

- Lässt sich ein optimales signaturbasiertes IDS **formal beschreiben**?
- $A_i := \{a \in \{0,1\}^* \mid a \text{ triggert Signatur } i\}$
- $a \text{ Angriff} \Leftrightarrow a \in A := A_1 \cup A_2 \cup \cdots \cup A_n$
- $\chi_A : \{0,1\}^* \rightarrow \{0,1\}$
- $\chi_A(a) = \begin{cases} 1, & a \in A \\ 0, & a \notin A \end{cases}$
- $a \text{ Angriff} \Leftrightarrow \chi_A(a) = 1$



Theoretische Grenze signaturbasierter IDS

- Ist es für **jede Signatur** möglich zu entscheiden, ob sie auf eine **beliebig gewählte Eingabe** passt?

- Obige Frage kann als **Instanz des Entscheidungsproblems** aufgefasst werden: Eine Menge A heißt entscheidbar, falls die **charakteristische Funktion** von A

$$\chi_A(a) = \begin{cases} 1, & a \in A \\ 0, & a \notin A \end{cases}$$

berechenbar ist.

- Ist $\chi_A(a)$ berechenbar?

  Oder in der Terminologie der Automatentheorie: Ist A **entscheidbar**?

Theoretische Grenze signaturbasierter IDS

- **Programmiersprachen** wie C, C++ oder Java sind **turing-vollständig**, das heißt, ihre Mächtigkeit ist (unter bestimmten Annahmen) äquivalent mit der einer Turing-Maschine.

- Ein Angriff kann Funktionen in einer turing-vollständigen Sprache enthalten. Turing-vollständige Sprachen sind vom **Typ 0**, also **semi-entscheidbar**:

$$\chi'_A(a) = \begin{cases} 1, & a \in A \\ undefiniert, & a \notin A \end{cases}$$

- Das **IDS** basiert ebenfalls auf einer turing-vollständigen Sprache, **simuliert also seinerseits eine Turingmaschine**.

- Damit wäre für ein optimales IDS das **Halteproblem** zu lösen.

Theoretische Grenze signaturbasierter IDS

- Eine alternative Betrachtung basiert auf dem **Satz von Rice**:

> Sei S eine nicht-triviale Teilmenge der turing-berechenbaren Funktionen. Dann ist die Sprache
>
> $$L(S) := \{ M \mid M \text{ berechnet eine Funktion aus } S \}$$
>
> nicht entscheidbar.

- **Extensionale Eigenschaften** von Programmen können mit einem **IDS**, das beliebige Programme verarbeiten muss, nicht entschieden werden.

  Konkret: Es ist im Allgemeinen **nicht möglich zu entscheiden**, ob observierte Daten einen **Angriff darstellen, oder nicht**.

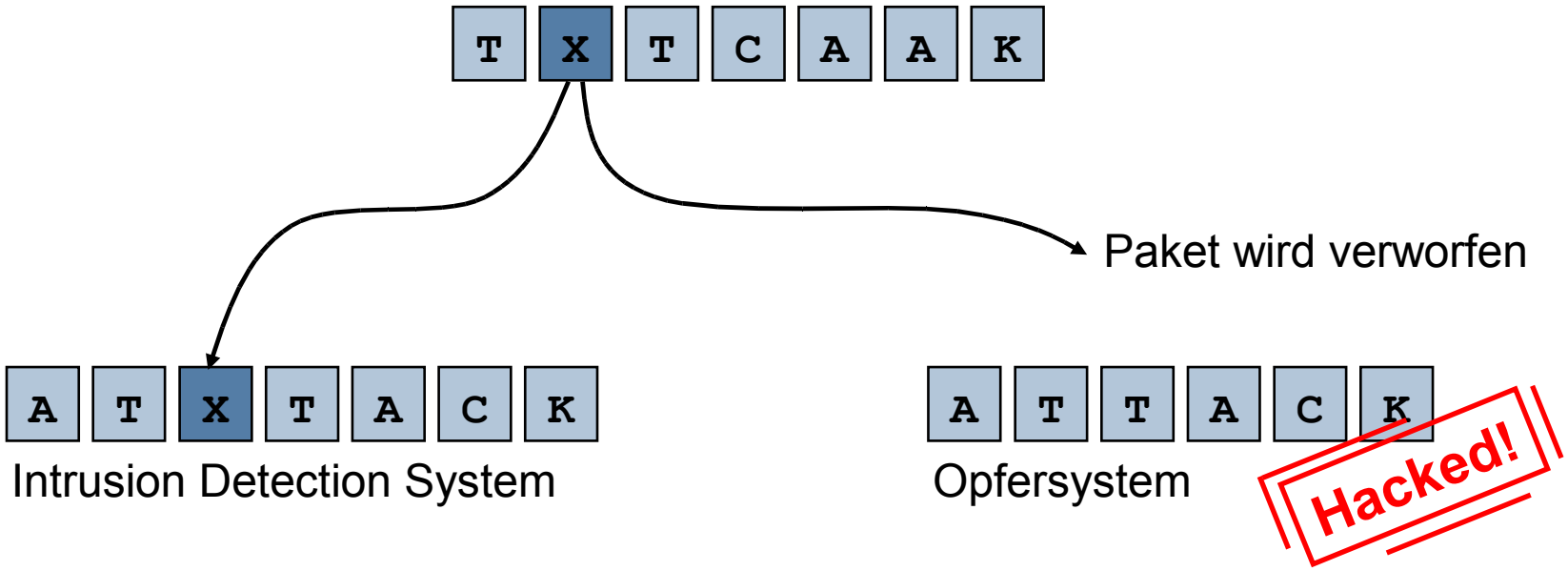
Agenda

- Einleitung – Signaturbasierter NIDS
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

Klassische Evasion-Techniken – Injection

- **Injection**: Das IDS verarbeitet ein Paket, welches vom Opfersystem nicht verarbeitet wird.

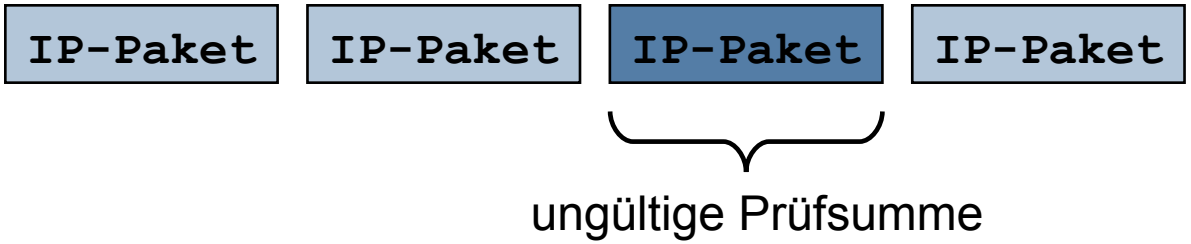
- Der Angreifer fügt dazu **zusätzliche Pakete** in die (eventuell permutierten) Angriffsdaten ein.

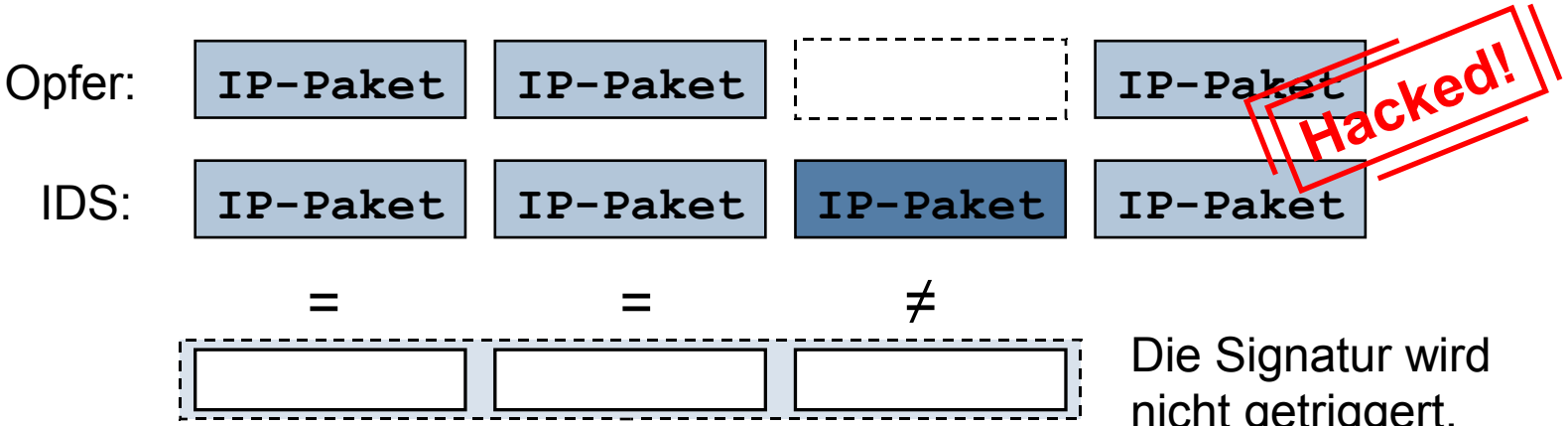


Quelle: [1]

Beispiel: Zu allgemeine Regeln auf dem IDS

- Injection-Beispiel:
  IP-Pakete mit **ungültiger Prüfsumme** werden vom IDS nicht verworfen.

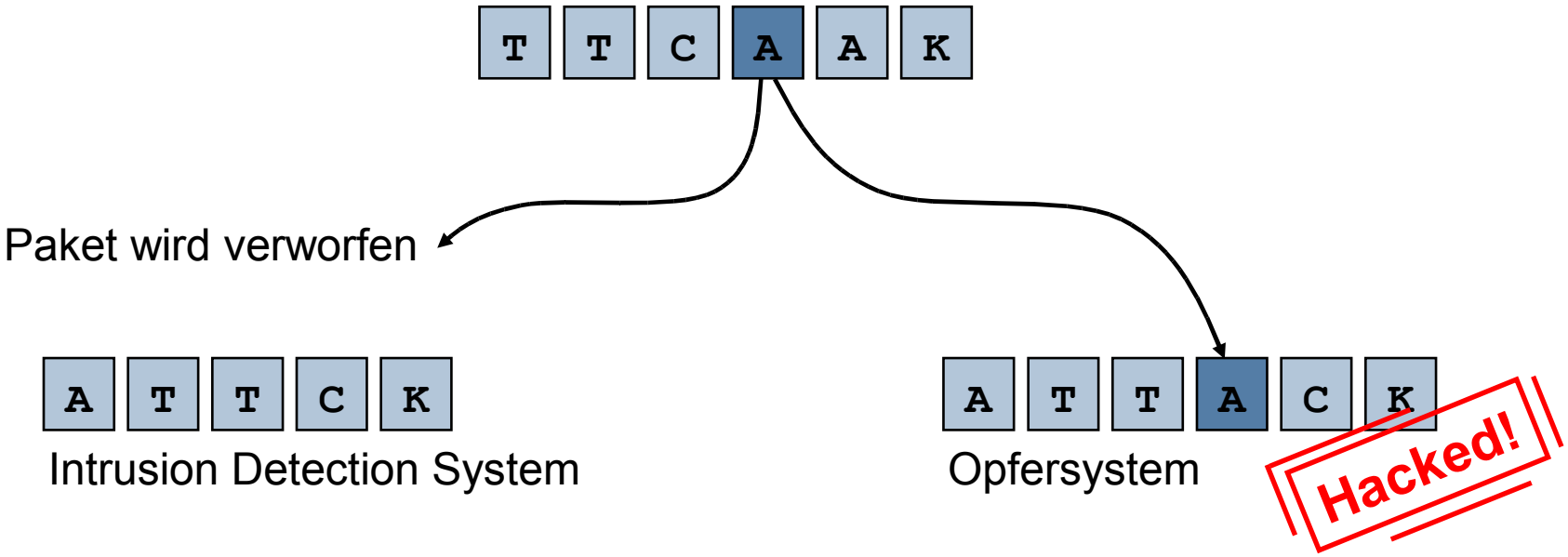




Klassische Evasion-Techniken – Evasion

- **Evasion**: Das IDS verarbeitet ein Paket nicht, welches vom Opfersystem verarbeitet wird.

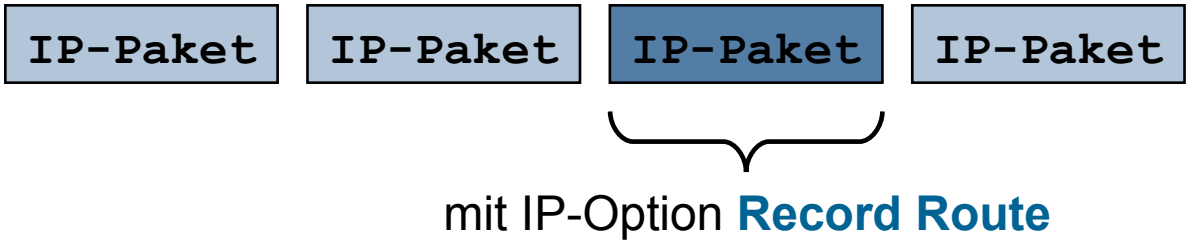
- Der Angreifer gestaltet dazu ein Paket der Angriffsdaten so, dass es vom IDS, aber nicht vom Opfer verworfen wird.

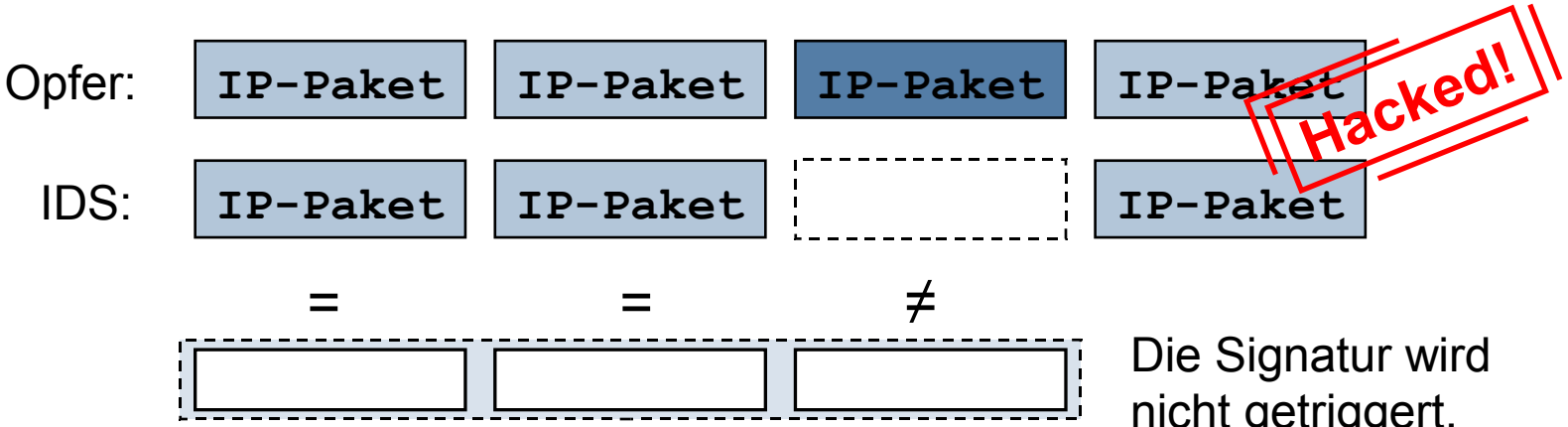


Quelle: [1]

Beispiel: Zu restriktive Regeln auf dem IDS

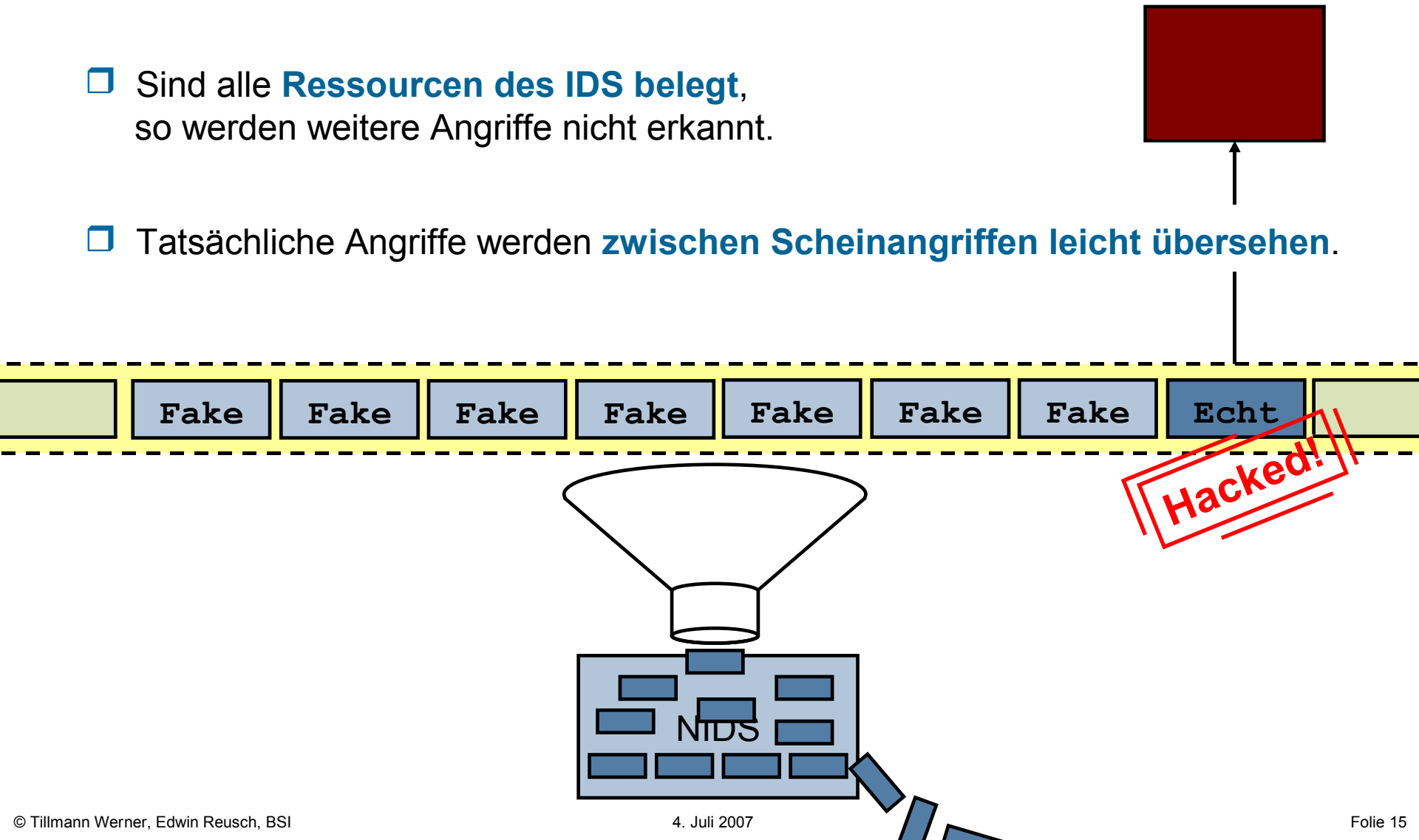
- Evasion-Beispiel:
  IP-Pakete mit bestimmten **IP-Optionen** werden vom IDS verworfen, weil dieses irrtümlich der Policy einer konkreten Implementierung folgt.





Klassische Evasion-Techniken – DoS

- **Denial-of-Service**: Das IDS wird mit Angriffen geflutet.
- Sind alle **Ressourcen des IDS belegt**,
  so werden weitere Angriffe nicht erkannt.
- Tatsächliche Angriffe werden **zwischen Scheinangriffen leicht übersehen**.



Agenda

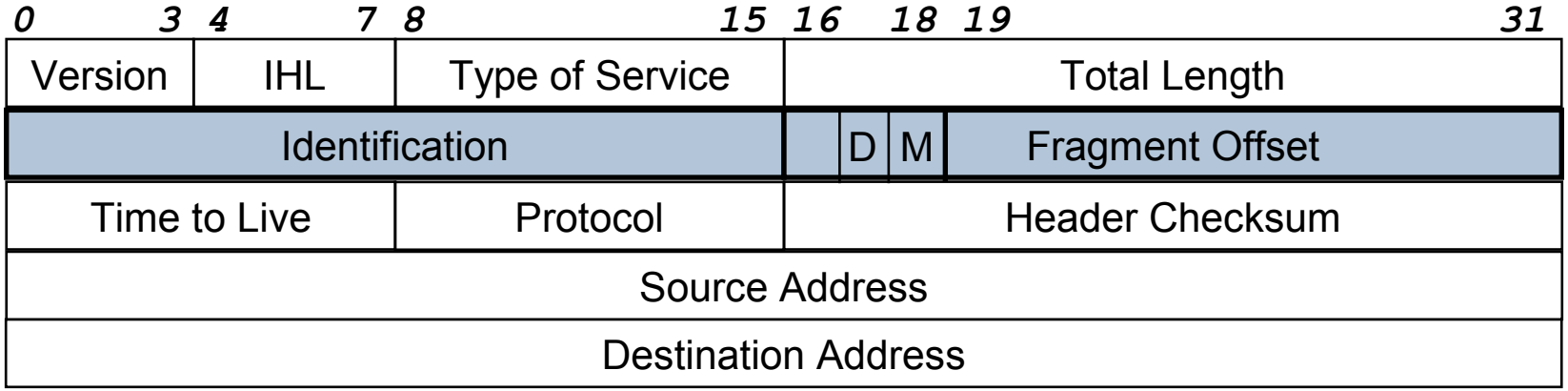
- Einleitung – Signaturbasierter NIDS
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

Fragmentation

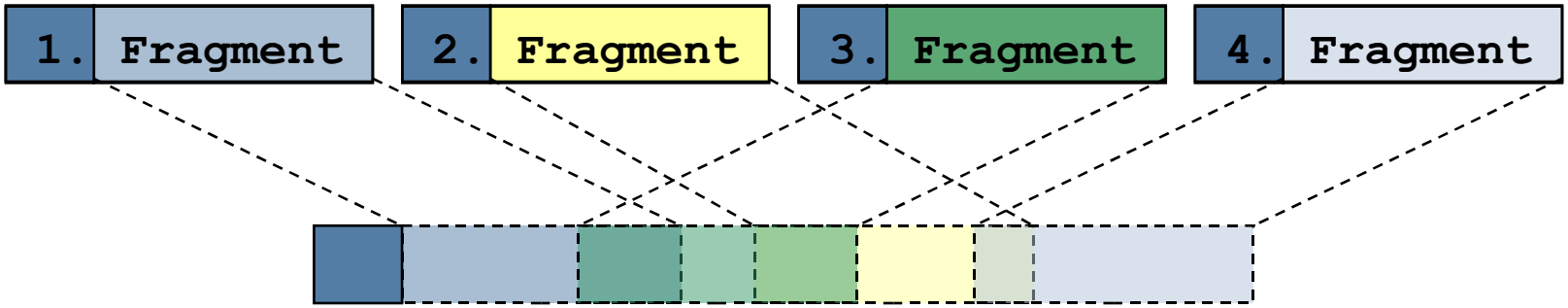
> Increasingly sophisticated attackers have begun to **exploit the more subtle aspects of the Internet Protocol**; fragmentation of IP packets […] poses several potential problems […].

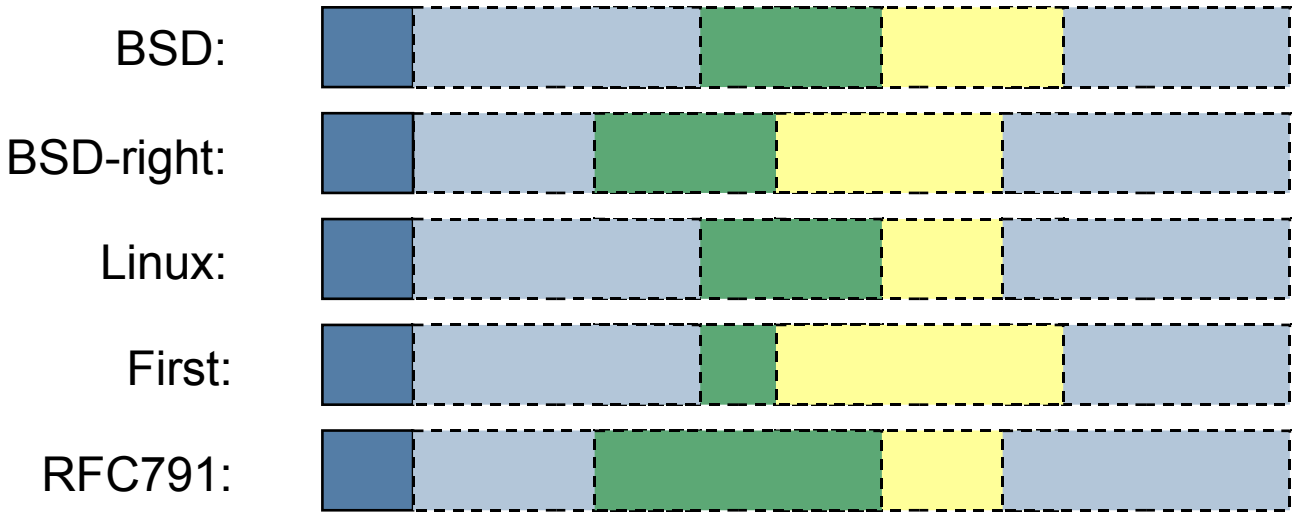
Quelle: RFC1858: Security Considerations for IP Fragment Filtering

Internet Protocol Header:



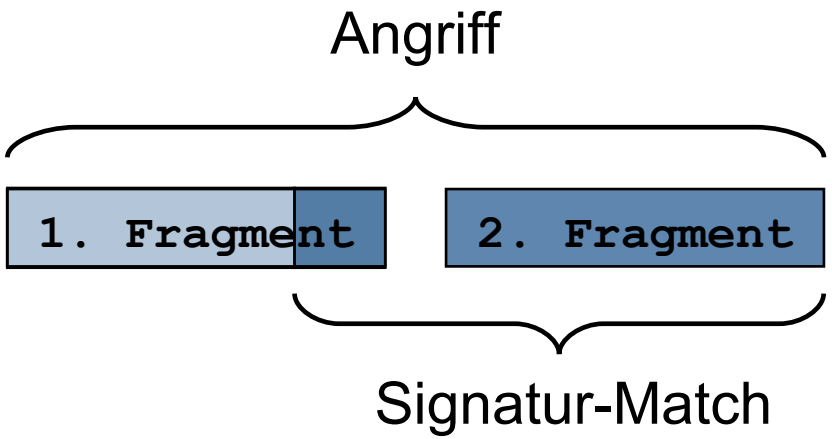
Unterschiedliche Reassembly-Strategien

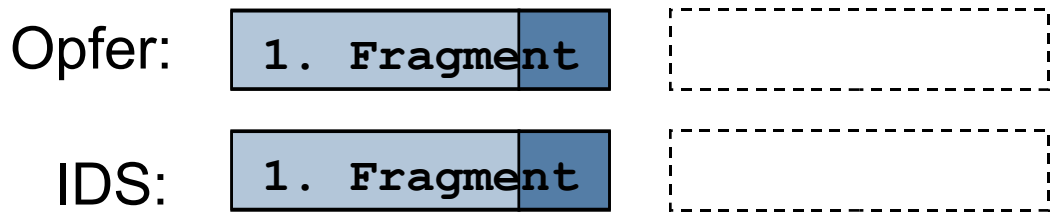




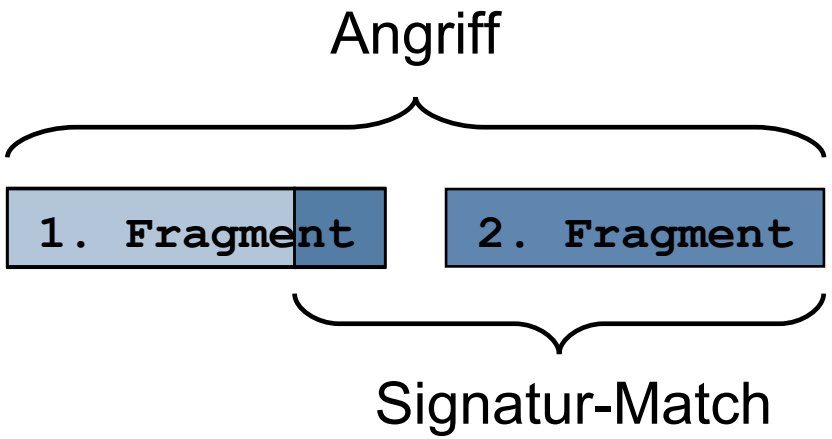
Quelle: [2]

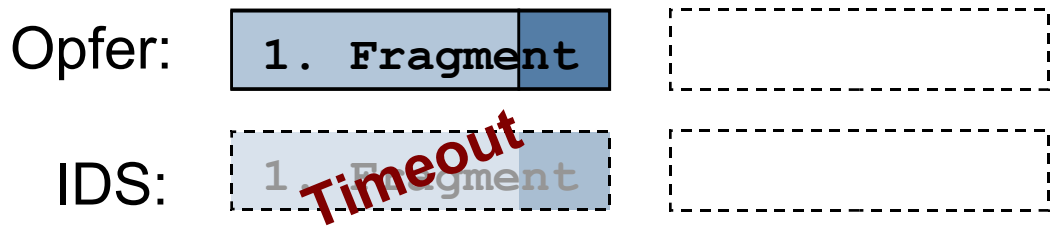
IP Reassembly Timeout des IDS



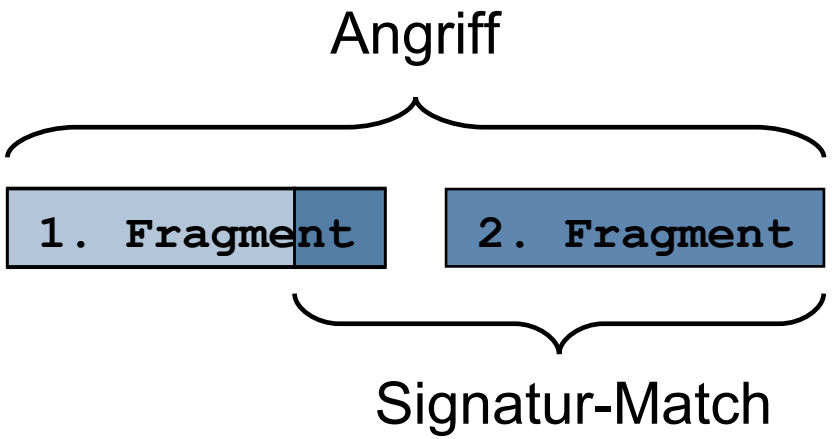


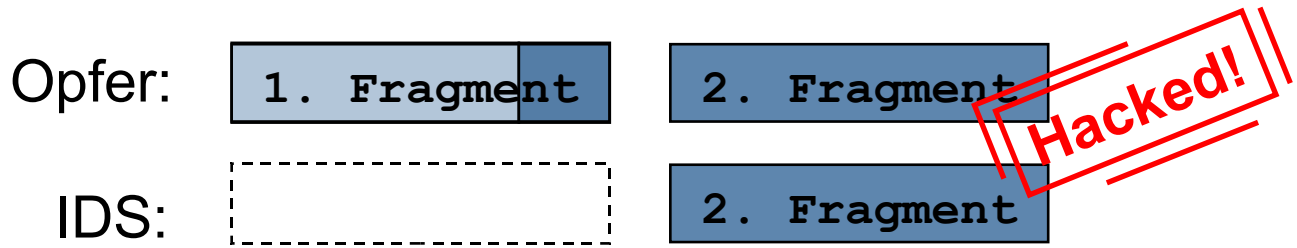
IP Reassembly Timeout des IDS



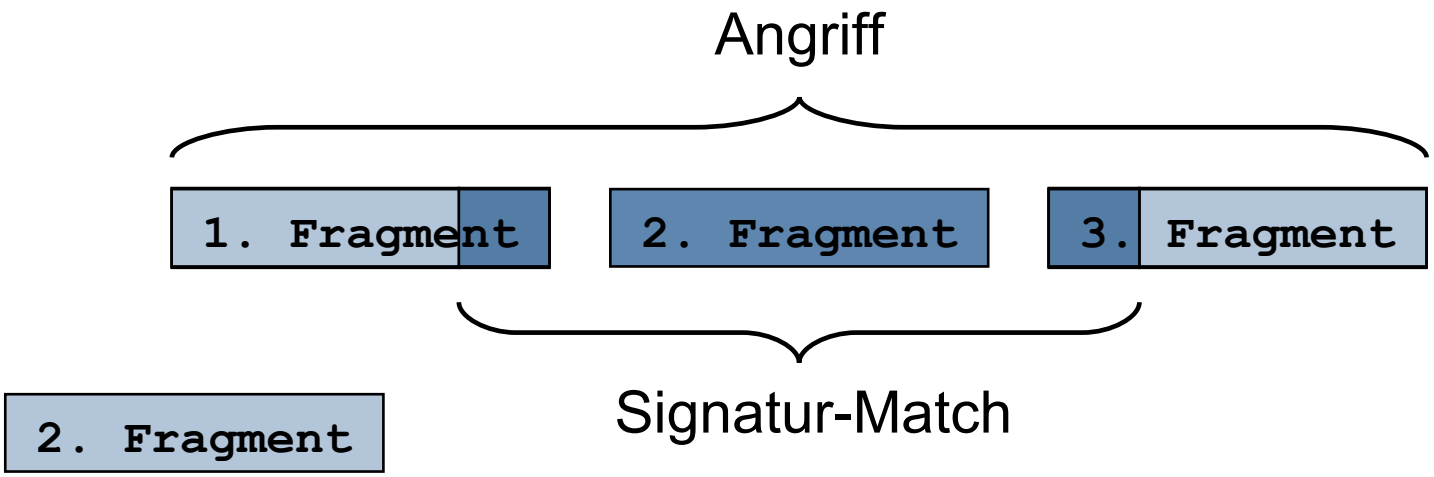


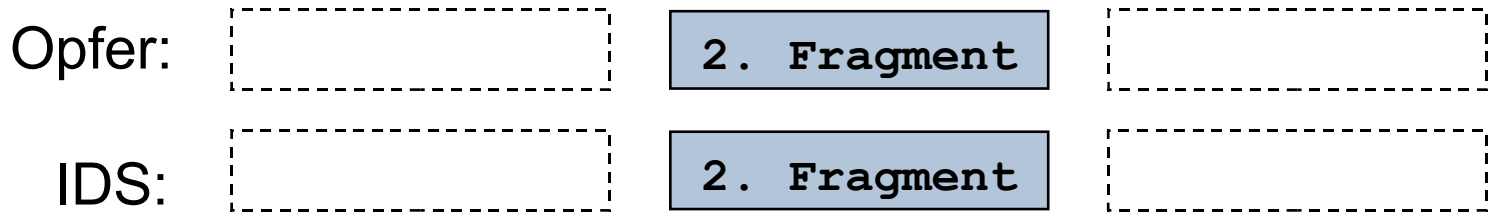
IP Reassembly Timeout des IDS



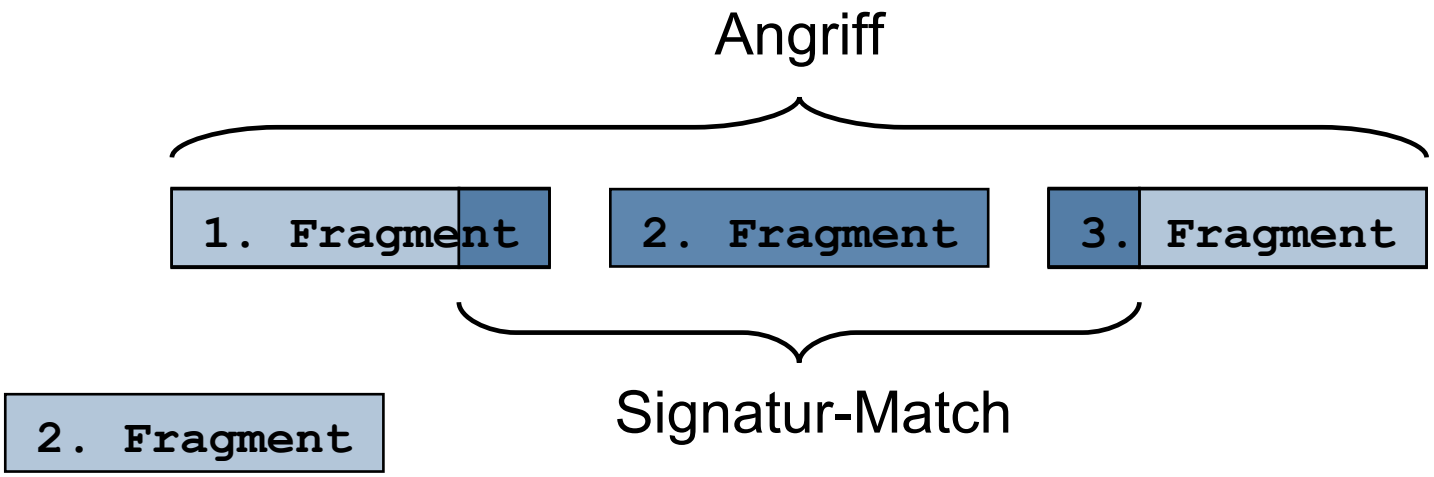


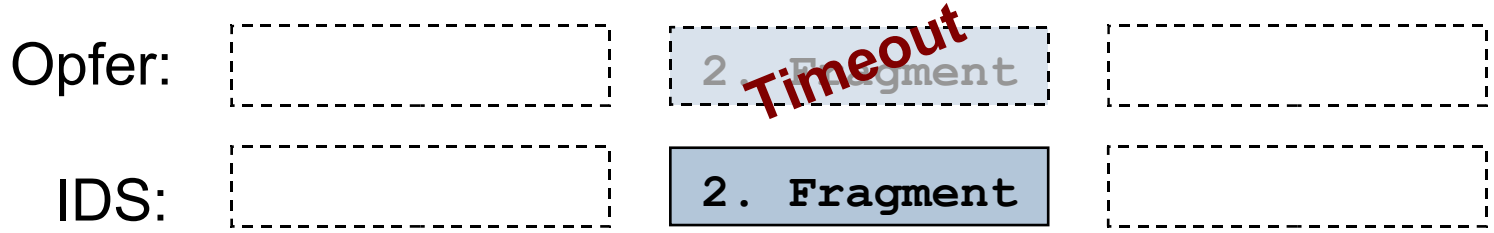
IP Reassembly Timeout des Opfers



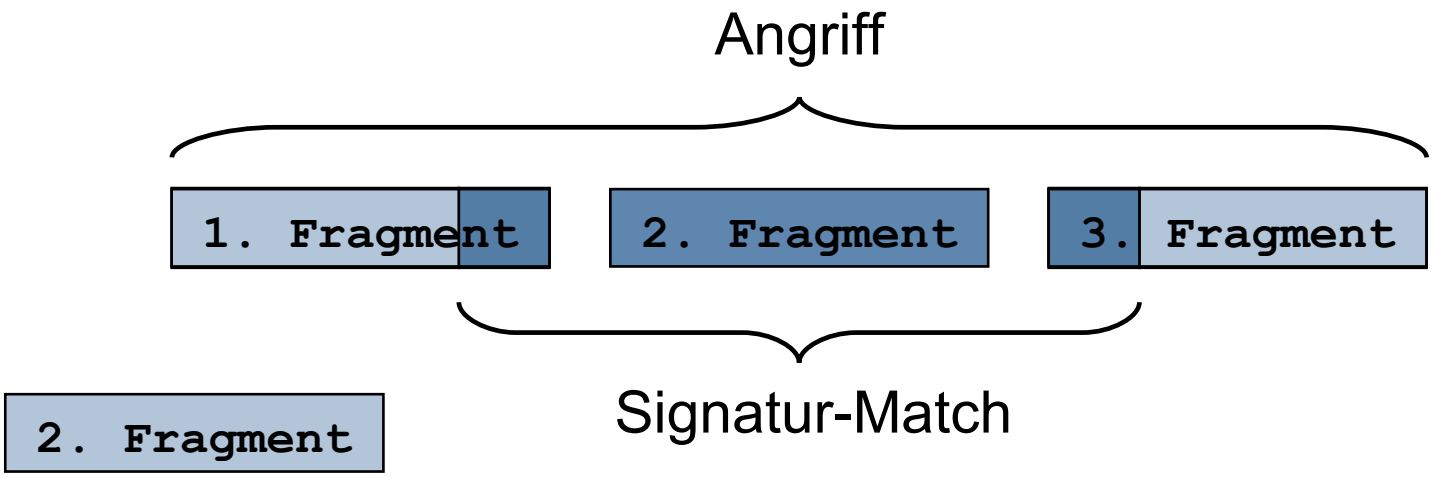


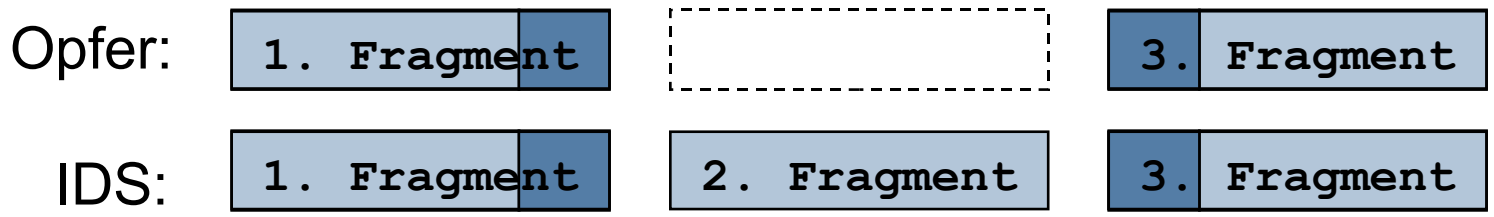
IP Reassembly Timeout des Opfers



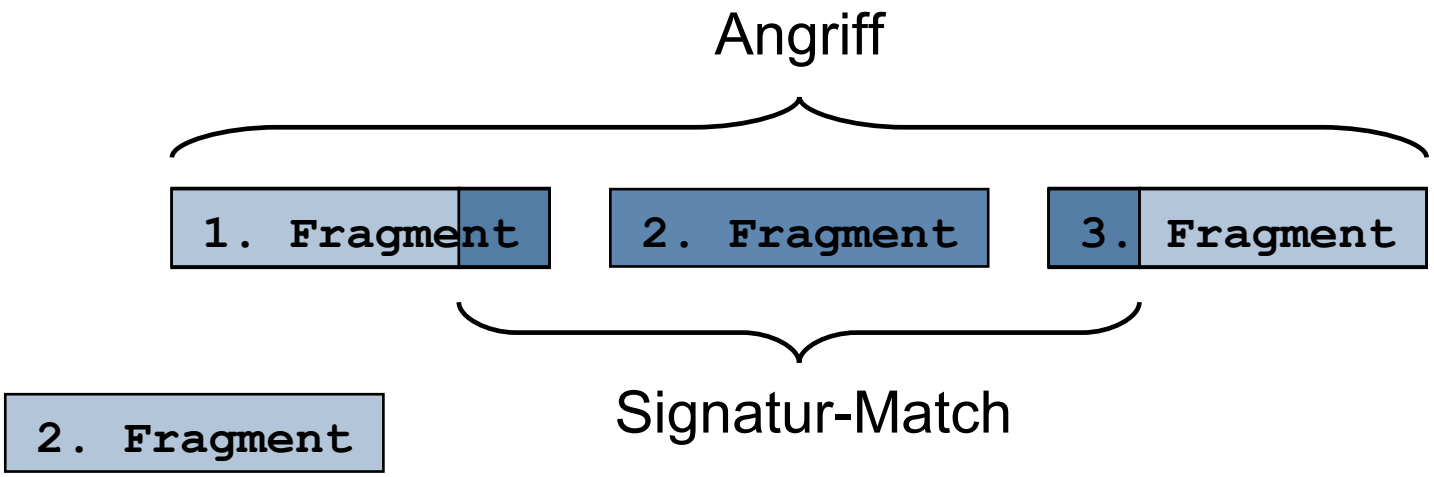


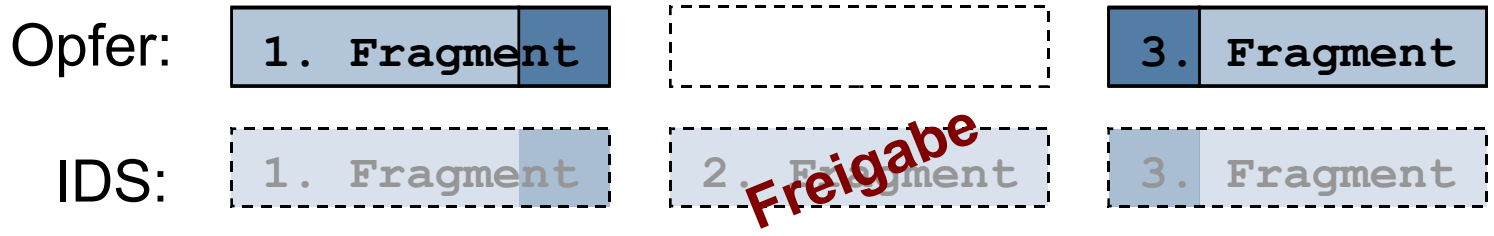
IP Reassembly Timeout des Opfers



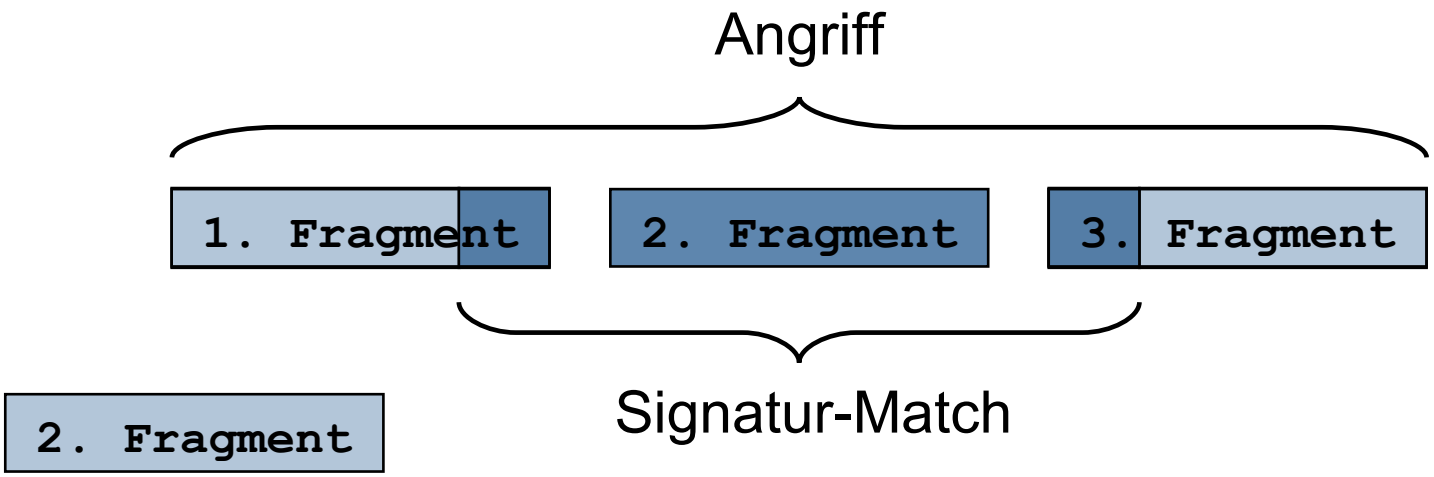


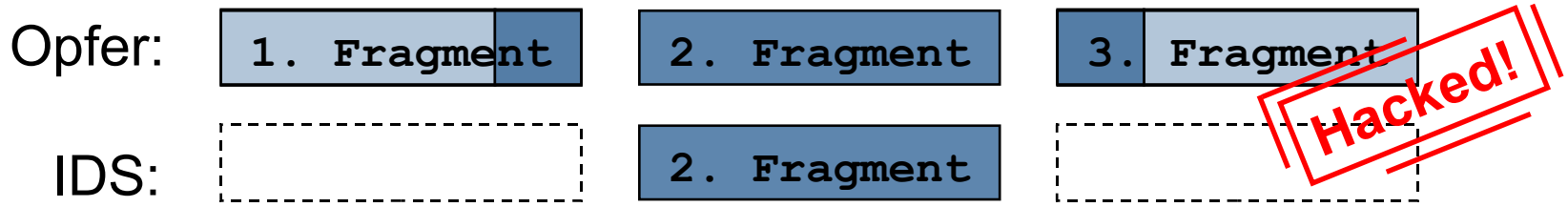
IP Reassembly Timeout des Opfers





IP Reassembly Timeout des Opfers





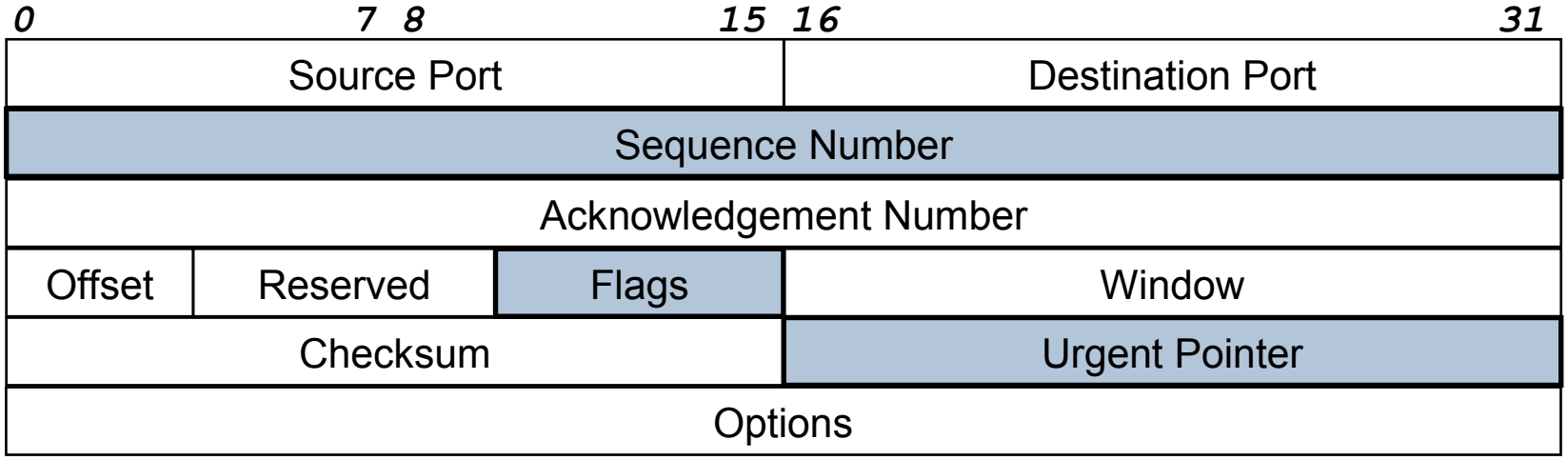
Agenda

- Einleitung – Signaturbasierter NIDS
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

IDS Evasion mit TCP

- Im Zusammenhang mit TCP lassen sich **zwei grundlegende Arten von Evasion-Angriffen** unterscheiden:
- Ähnlich wie auf IP-Ebene kann der **TCP Stream manipuliert** werden, so dass Nutzdaten vom IDS nicht korrekt interpretiert werden.
- Ein Angreifer kann gezielt dafür sorgen, dass das IDS **ungültige State-Informationen** führt.

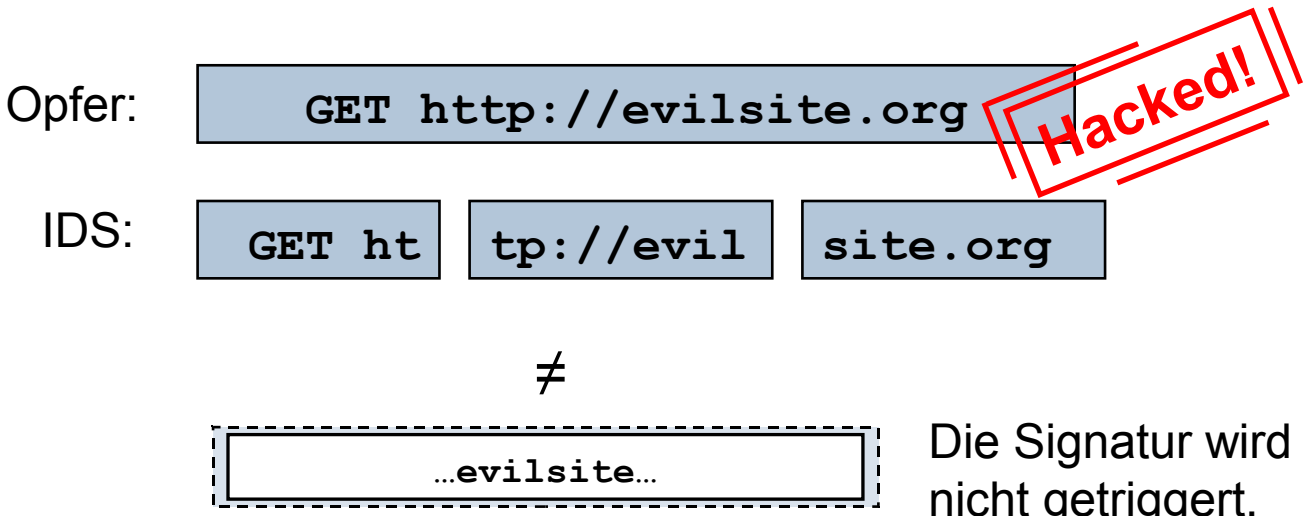
Transmission Control Protocol Header:



TCP Session Splicing

- Mit Daten in mehreren, sehr kleinen TCP-Segmenten kann das Pattern Matching des IDS umgangen werden (**Session Splicing**).

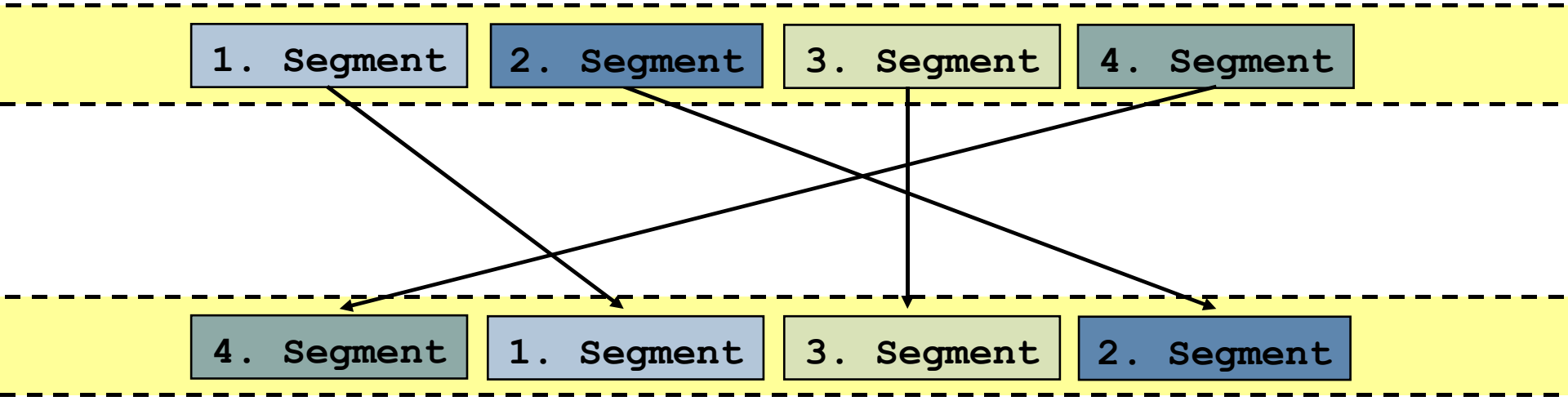
- Das Opfersystem setzt die Segmente wieder zum TCP Stream zusammen, das IDS nicht.



TCP Segment Permutation

- Ein weiterer Angriff besteht darin, die TCP-Segmente in **permutierter Reihenfolge** zu senden, um das Zusammensetzen zu erschweren.

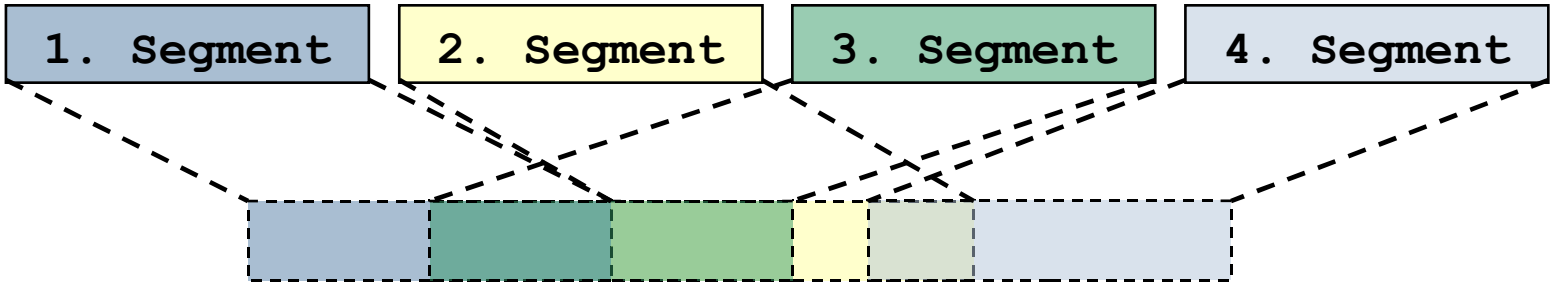
- Setzt das IDS den Stream anhand der Segment-Reihenfolge zusammen, schlägt das Pattern Matching fehl.

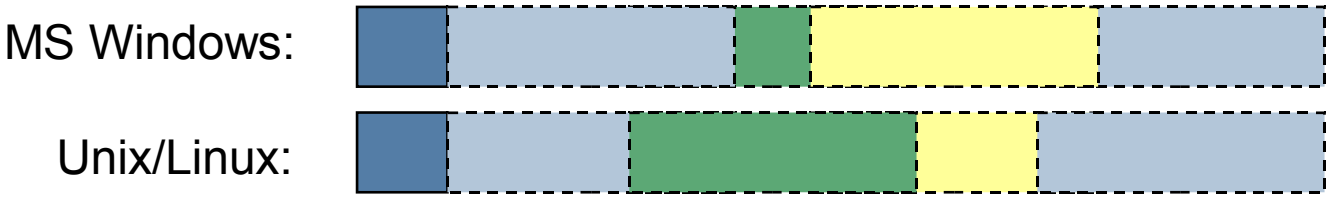


Strategien beim TCP Stream Reassembly

- Damit im IDS ein **Reassembly der TCP Streams** erfolgen kann, ist ein **Tracking der verwendeten Sequenznummern** erforderlich.

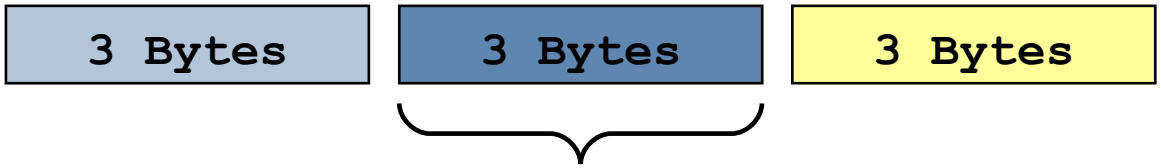
- Oft kann der Synchronisations-Algorithmus mit **speziell konstruierten Segmenten** gezielt attackiert werden.



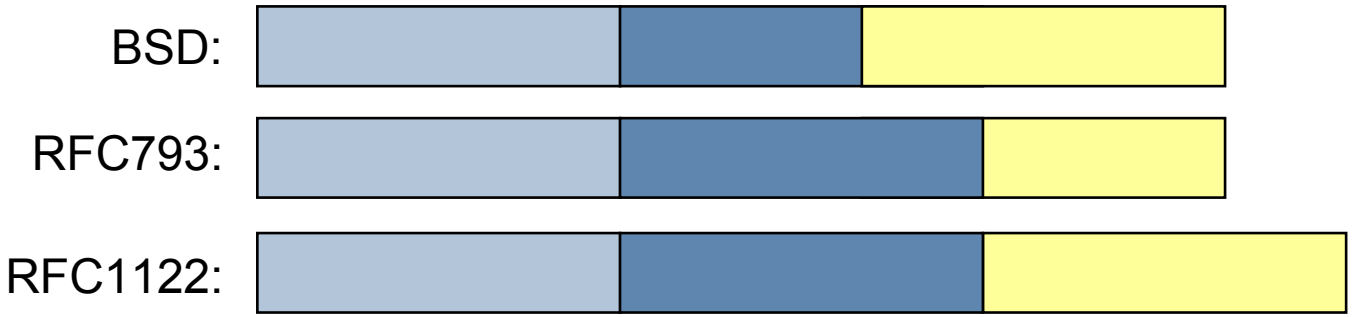


Injection/Evasion-Angriff mit Urgent Data

- Verschiedene Betriebssysteme interpretieren den **Urgent Pointer** im TCP Header unterschiedlich.
- Dies kann wiederum ausgenutzt werden, wenn das IDS und das Opfer eines Angriffs abweichende Strategien verwenden.



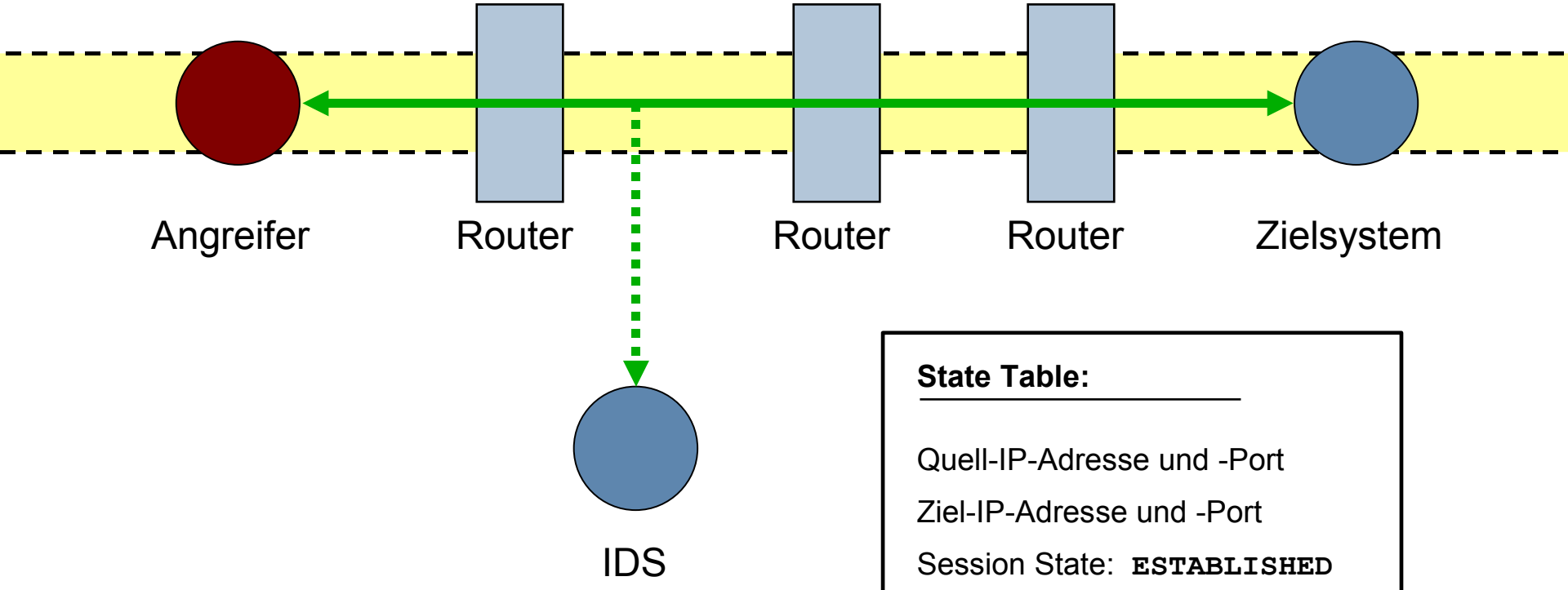
Urgent Data, Urgent Pointer = 3



Quelle: [3]

Manipulation der State Table mit RST-Segmenten

- Ein **TCP-Verbindungsaufbau** (**`SYN`**, **`SYN/ACK`**, **`ACK`**) führt zu einem **Eintrag in der State Table** des IDS.



State Table:

Quell-IP-Adresse und -Port

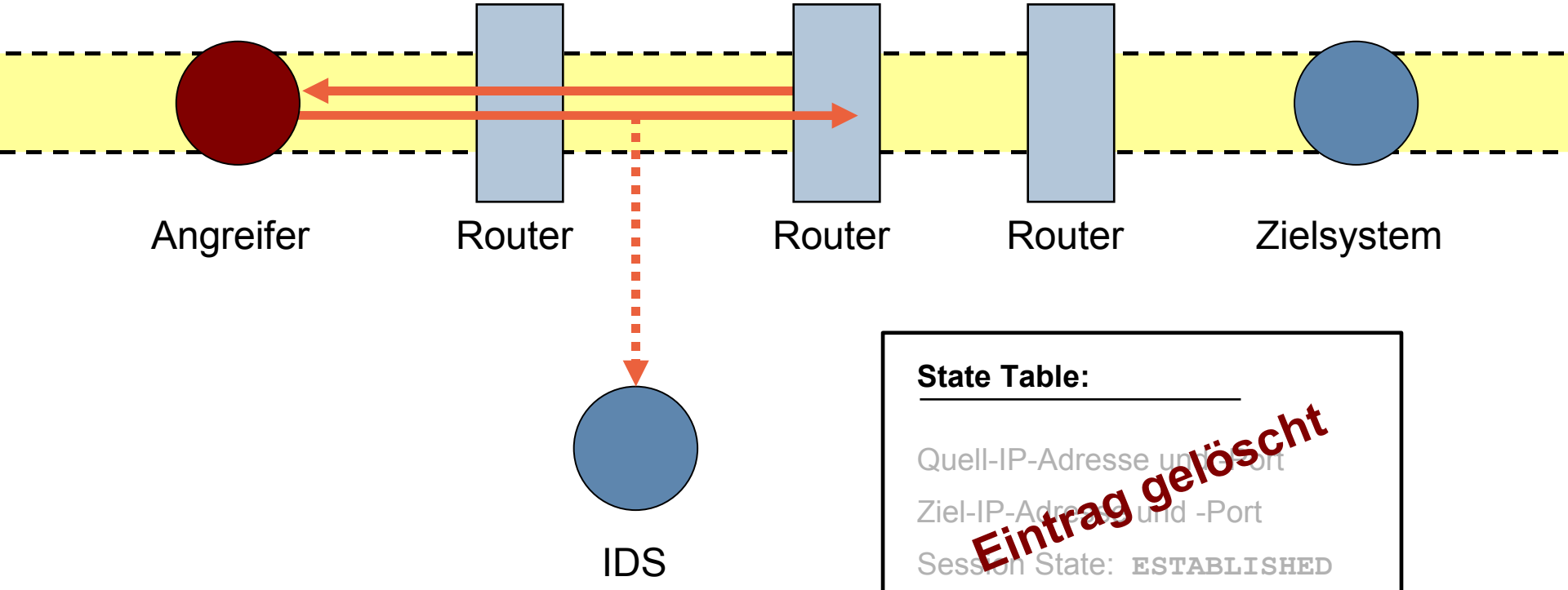
Ziel-IP-Adresse und -Port

Session State: **`ESTABLISHED`**

Manipulation der State Table mit RST-Segmenten

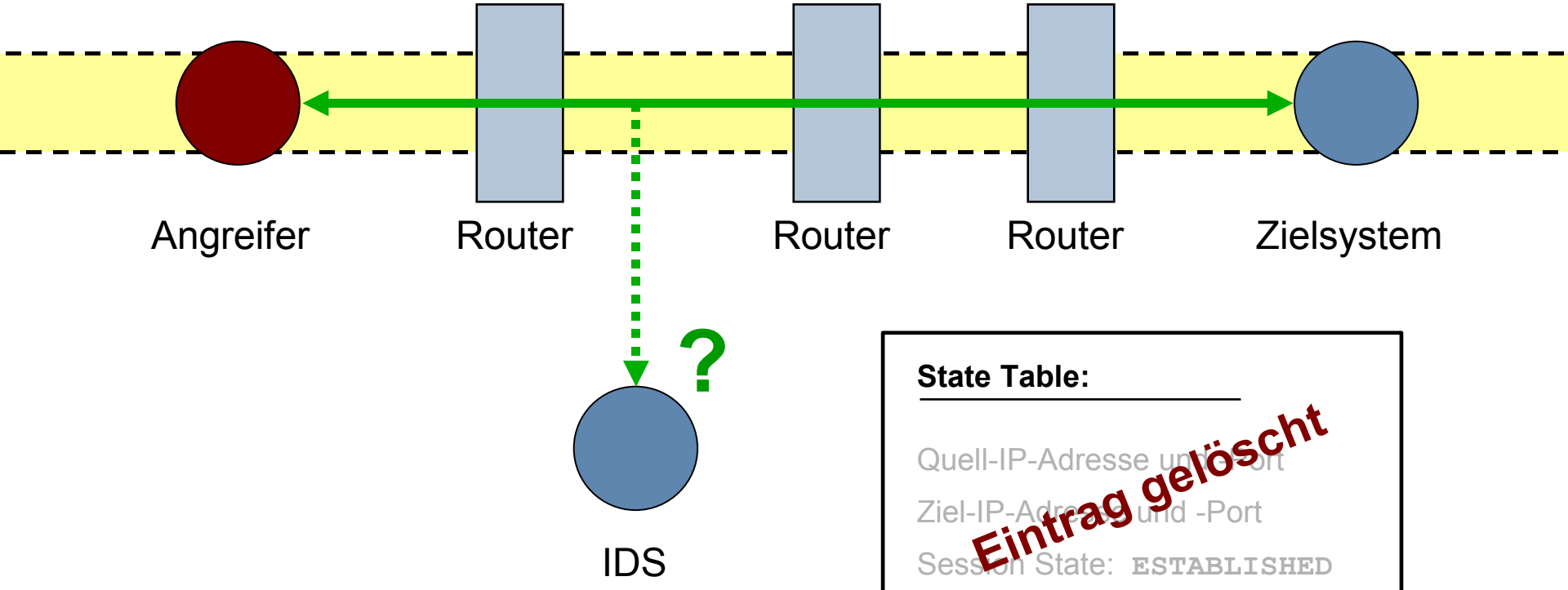
- Vorgetäuschter Verbindungsabruch:
  RST-Segment mit **TTL kleiner als die Anzahl der Hops zum Zielsystem**

- Der Router reagiert mit einer **ICMP-Nachricht** Type 11, Code 0
  (**`TTL EXCEEDED IN TRANSIT`**), die **vom IDS nicht verarbeitet** wird.

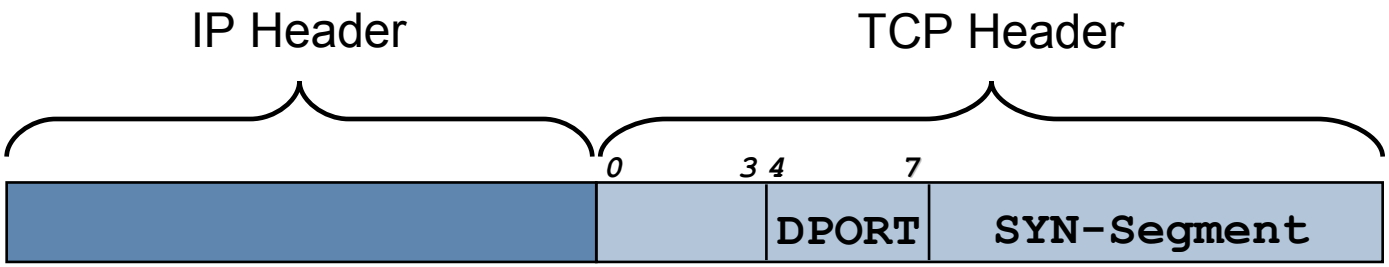


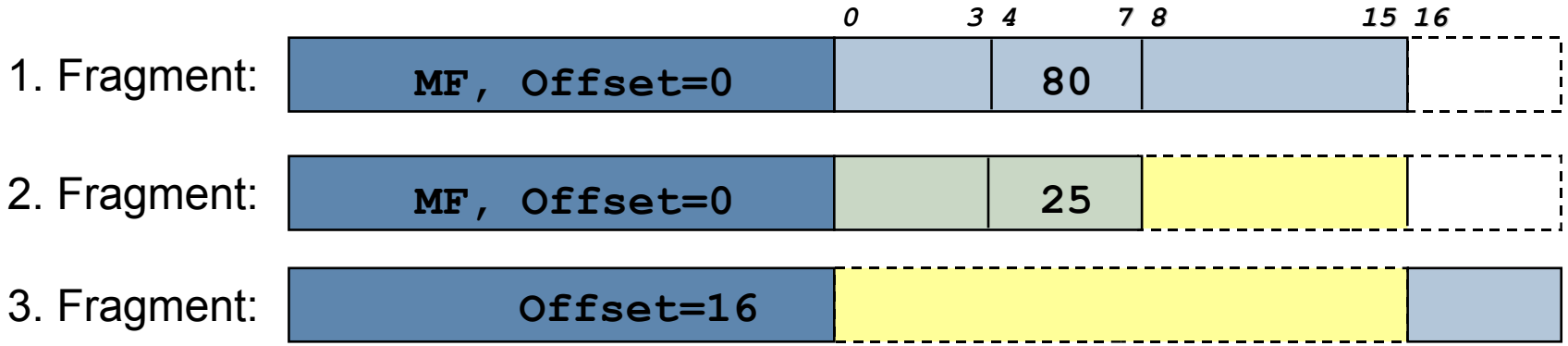
Manipulation der State Table mit RST-Segmenten

- Der Angreifer kann nun **Daten** über die noch existente Verbindung senden.
- Das IDS kann dafür **keine Zuordnung** vornehmen, weil **kein passender Eintrag in der State Table** vorhanden ist.



Manipulation der State Table – TCP Port Overriding



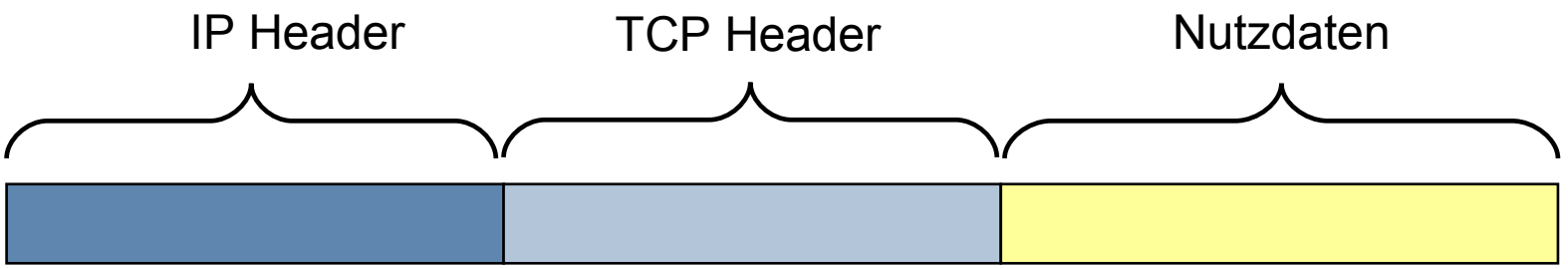


- Bei unterschiedlichen Reassembling-Strategien erhält das IDS ein **SYN**-Segment an Port 80 und das Opfer ein **SYN**-Segment an Port 25.

Agenda

- Einleitung – Signaturbasierter NIDS
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

Manipulation der Anwendungschicht



- Neben den vorgestellten Techniken (IP Fragmentation, Session Splicing, …) existieren weitere Methoden, um **Signaturen für Nutzdaten auszuhebeln**.
- Üblicherweise werden die Daten dabei **ohne Änderung der Semantik** so **umgeformt**, dass Signaturen nicht mehr greifen.
- Um derartig transformierte Angriffe erkennen zu können, muss ein IDS über entsprechende **Logik auf Anwendungsebene** verfügen.

Beispiele für manipulierte Nutzdaten

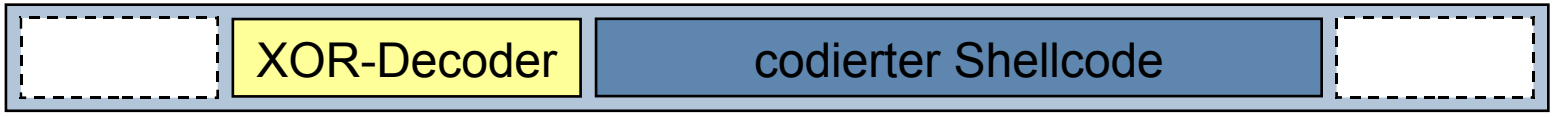
- HTTP unterstützt als **URL-Kodierung** neben ASCII auch Percent Encoding, UTF8 und Unicode.

```
GET%20%7E/cgi.bin%3Fani.jpg%3D1
```

- Verzeichnisangaben in Protokollen wie telnet und ftp können um redundante **Directory-Traversal-Anteile** ergänzt werden.

```
/./etc/../etc/.//passwd
```

- **Polymorpher Shellcode** kann signaturbasiert nur schwer erkannt werden.



URL-Verschleierung in HTTP-GET-Requests

- HTTP-GET-Request:

```
GET ~/cgi-bin?ani.jpg=1
```

- UTF8-Codierung:

```
GET%20%7E/cgi.bin%3Fani.jpg%3D1
GET&#32;&#126;/cgi.bin&#63;ani.jpg&#61;1
```

- URL-Codierung:

```
%47%45%54%20%7E%2F%63%67
%69%2E%62%69%6E%3F%61%6E
%69%2E%6A%70%67%3D%31
```

- Je mehr Variationen möglich sind, desto mehr Logik ist im IDS notwendig.

Quelle: [4]

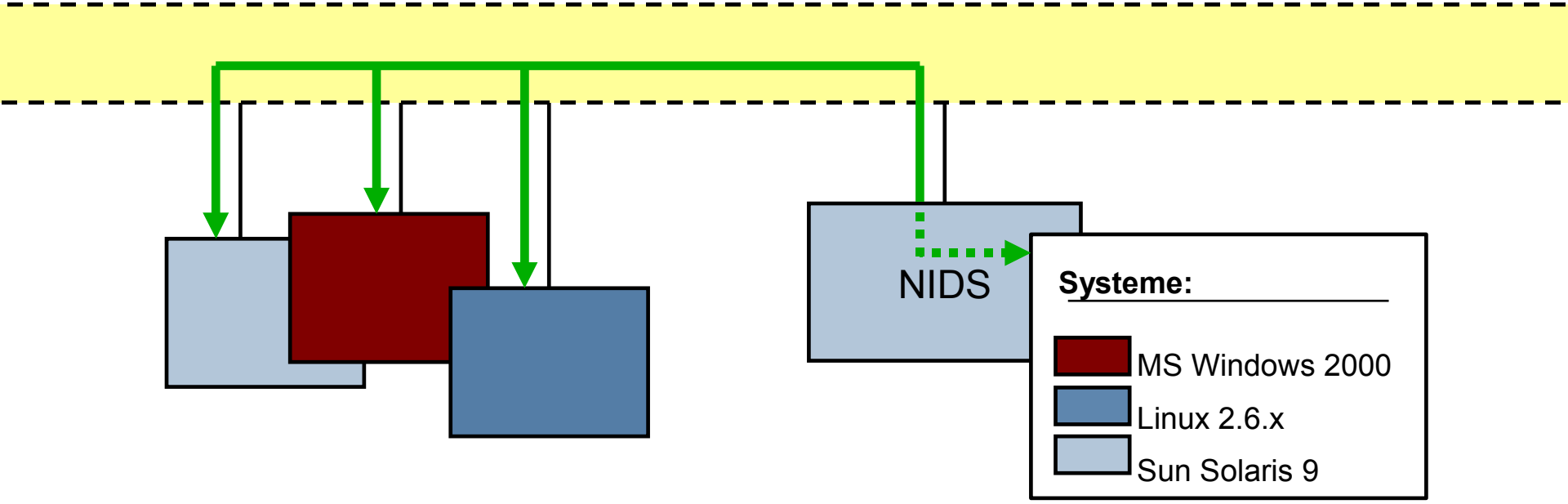
Agenda

- Einleitung – Signaturbasierter NIDS
- Praktische und theoretische Grenzen
- Überblick über klassische Evasion-Techniken
- Evasion mit fragmentierten IP-Paketen
- Evasion mit speziellen TCP-Segmenten
- Evasion auf Anwendungsebene
- Gegenmaßnahmen und Ausblick

Gegenmaßnahmen – Active Mapping

- Bestimmte Injection- und Evasion-Angriffe können verhindert werden, wenn ein IDS **aktiv ermitteln** kann, **wie das zu schützende Netz beschaffen ist**.

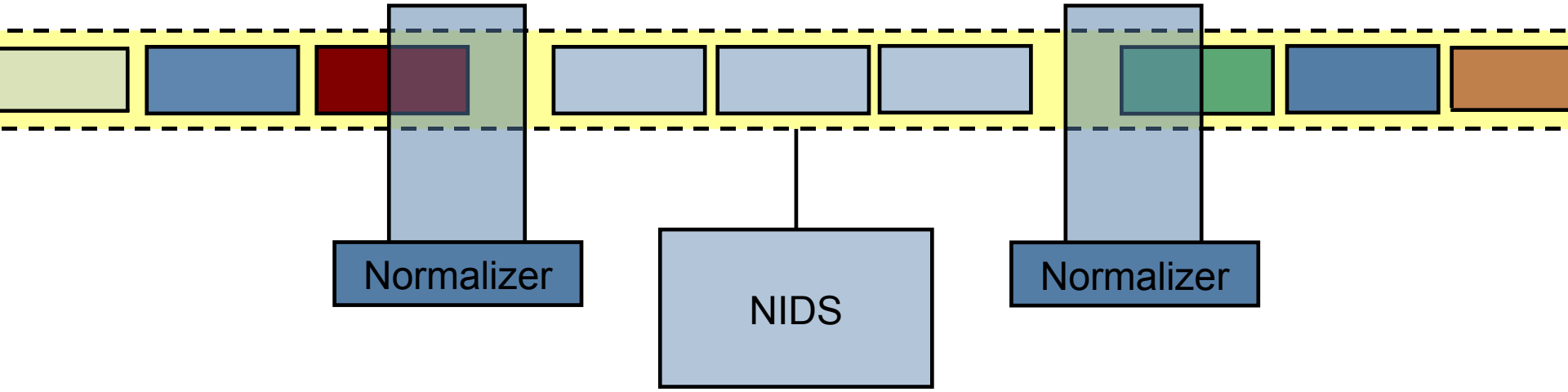
- Mit Active Mapping wird versucht, **Betriebssystem-Versionen** und aktive **Dienste** zu bestimmen.



Gegenmaßnahmen – Normalisierung

- Mit **vor- und nachgeschalteten Normalisierern** können mehrdeutige Daten vereinheitlicht werden. Allerdings bedeutet dieses Setup einen massiven Eingriff in die Netzstruktur.

- Das IDS wird nur mit **eindeutig interpretierbaren Paketen** konfrontiert.



Quellen und Links

- [1] *Insertion, Evasion and Denial-of-Service: Eluding Network Intrusion Detection*
  http://insecure.org/stf/secnet_ids/secnet_ids.pdf
- [2] *Active Mapping: Resisting NIDS Evasion Without Altering Traffic*
  http://www.icir.org/vern/papers/activemap-oak03.pdf
- [3] *Phrack 0x0b, Line Noise, NIDS Evasion Method named "SeolMa"*
  http://phrack.org/issues.html?issue=57&id=3
- [4] *UTF8 Encoder/Decoder*
  http://wwwrsphysse.anu.edu.au/~mxk121/javascript/jsUTF8.html
- [5] *IDS Evasion with Unicode*
  http://www.securityfocus.com/infocus/1232
- [6] *IDS Evasion Techniques and Tactics*
  http://www.securityfocus.com/infocus/1577
- [7] *Evading NIDS, revisited*
  http://www.securityfocus.com/infocus/1852
- [8] *RFC 815 - IP Datagram Reassembly Algorithms*
  http://www.faqs.org/rfcs/rfc815.html
- [9] *RFC 1858 - Security Considerations for IP Fragment Filtering*
  http://www.faqs.org/rfcs/rfc1858.html

Kontakt

Bundesamt für Sicherheit in der
Informationstechnik (BSI)

Tillmann Werner, Edwin Reusch
Referat 121 – CERT-Bund
Godesberger Allee 185-189
53175 Bonn

Tel: +49 (0)1888 9582-5110
Fax: +49 (0)1888 9582-5427

tillmann.werner@bsi.bund.de
edwin.reusch@bsi.bund.de

http://www.bsi.bund.de
http://www.cert-bund.de