Erkennung von Stepping Stones in breitbandigem Netzwerkverkehr

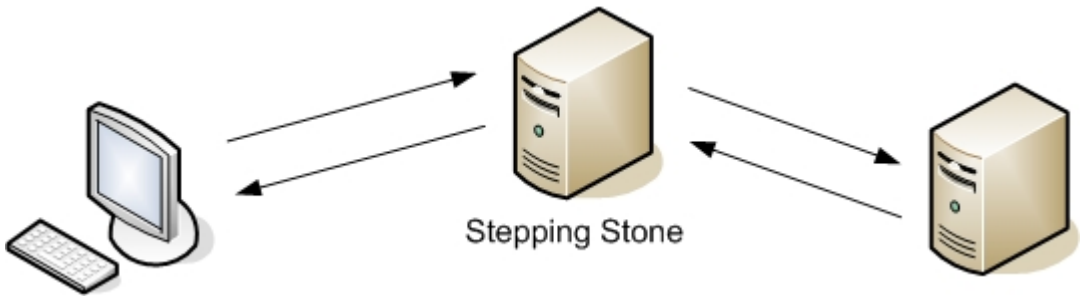
Agenda

- Einleitung
- Algorithmen zur Erkennung
- Vergleich der Algorithmen
- Fazit und Ausblick

Einleitung

- Intrusion Detection

- Stepping Stones



Algorithmen zur Erkennung

- Vergleichskriterien
- Algorithmen
- Vorauswahl der betrachteten Algorithmen

Algorithmen zur Erkennung: Vergleichskriterien

- Inhalts- / Timingbasiert
- Aktiv / Passiv
- Realtime- / Offlineanalyse
- Anzahl der Meßpunkte

Algorithmen zur Erkennung: Qualitative Merkmale

- Erkennungsrate
- Geschwindigkeit
- Speicherverbrauch

Algorithmen zur Erkennung: Umgehung der Erkennung

- Veränderung des Timings
- Chaff-Pakete
- Überlastung

Algorithmen zur Erkennung: Algorithmen

- ON/OFF Algorithmus
- Watermarking Algorithmus
- Abstandsalgorithmus
- Interpacket-Delay Algorithmus
- Wavelet Algorithmus
- Paket-Zähler Algorithmus
- Hop-Count Algorithmus

Algorithmen zur Erkennung: ON/OFF Algorithmus

- Zhang / Paxson (2000)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- geringe CPU-Last
- geringer Speicherverbrauch

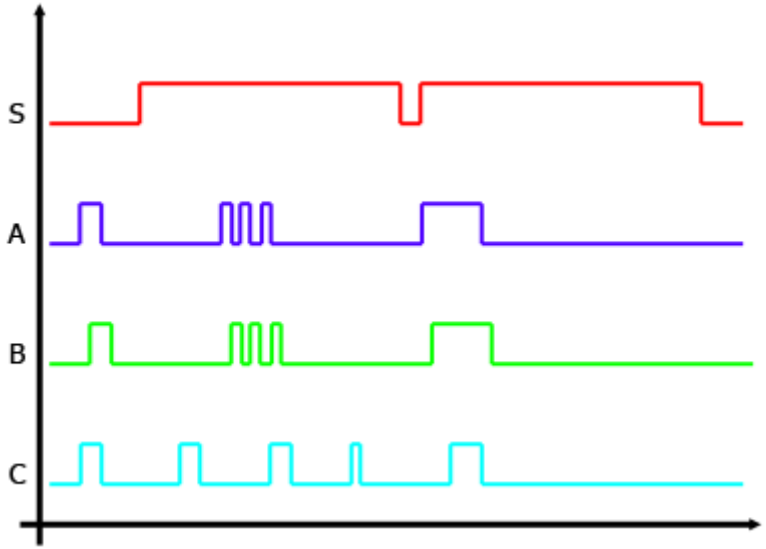
Algorithmen zur Erkennung: ON/OFF Algorithmus

- Parameter T_{idle}
- Parameter δ
- Parameter γ
- Parameter γ‘
- Parameter min_{csc}

$$\frac{OFF_{1,2}}{\min(OFF_1, OFF_2)} \geq \gamma$$

$$\frac{OFF^*_{1,2}}{\min(OFF_1, OFF_2)} \geq \gamma'$$

Algorithmen zur Erkennung: ON/OFF Algorithmus



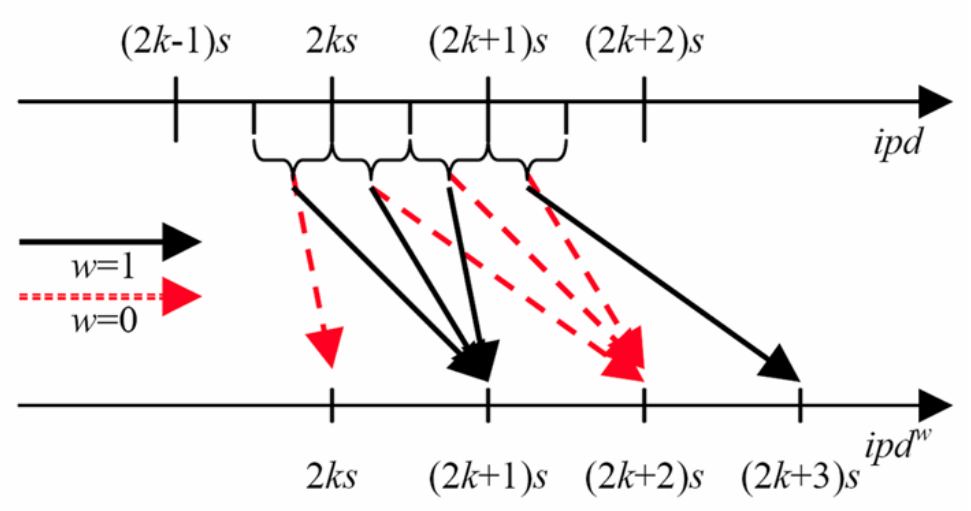
Algorithmen zur Erkennung: Watermarking Algorithmus

- Wang / Reeves (2003)
- timingbasierte Analyse
- aktiv
- nur Online möglich
- mehrere Meßpunkte möglich
- geringe CPU-Last
- geringer Speicherverbrauch

Algorithmen zur Erkennung: Watermarking Algorithmus

- Einbettung eines Wasserzeichens
- Anschließende Erkennung
- Robust gegen Timingveränderung

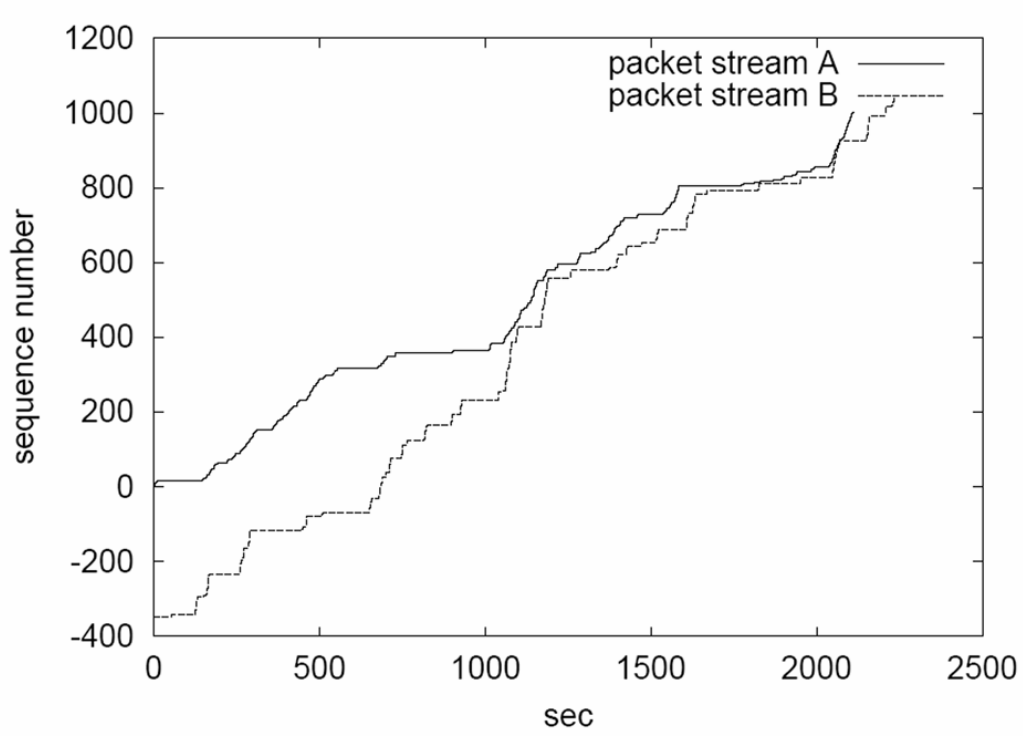
Algorithmen zur Erkennung: Watermarking Algorithmus



Algorithmen zur Erkennung: Abstandsalgorithmus

- Yoda / Etoh (2000)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- hohe CPU-Last
- hoher Speicherverbrauch

Algorithmen zur Erkennung: Abstandsalgorithmus



Algorithmen zur Erkennung: Interpacket-Delay Algorithmus

- Wang / Reeves / Wu (2002)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- hohe CPU-Last
- hoher Speicherverbrauch

Algorithmen zur Erkennung: Interpacket-Delay Algorithmus

- Correlation Point Funtion (CPF)
  - NDP1
  - NDP2
  - STAT
  - MIN/MAX
- Correlation Value Function (CVF)

$$CVF(C_x, C_y) = \begin{cases} 0 & , n = 0 \\ \rho(C_x, C_y) & , n > 1 \\ 1 & , n = 1 \end{cases}$$

$$CPF(X, Y, j, k, s)_{MMS} = \frac{\sum_{i=j}^{j+s+1} \min(x_i, y_{i+k})}{\sum_{i=j}^{j+s+1} \max(x_i, y_{i+k})}$$

$$\rho(C_x, C_y) = \frac{\sum_{i=1}^{n} (j_i - E(C_x)) \times (j_i + k_i - E(C_y))}{\sqrt{\left[\sum_{i=1}^{n} (j_i - E(C_x))^2\right] \times \left[\sum_{i=1}^{n} (j_i + k_i - E(C_y))^2\right]}}$$

Algorithmen zur Erkennung: Wavelet Algorithmus

- Donoho / Flesia / Shanker / Paxson / Coit / Staniford (2002)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- mittlere CPU-Last
- hoher Speicherverbrauch

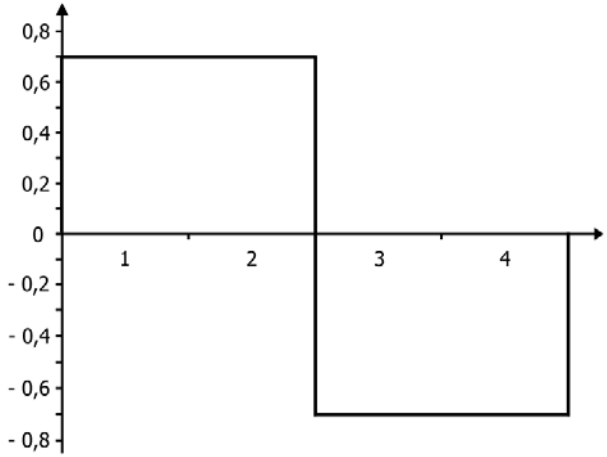
Algorithmen zur Erkennung: Wavelet Algorithmus

Wavelet-Transformation

- Signalverarbeitung
- Bildkompression
- Fokusierung auf relevante Teile
- Komposition einer einfachen Funktion

Algorithmen zur Erkennung: Wavelet Algorithmus

- Haar-Wavelet (1909)



- Daubechies 12 - Wavelet

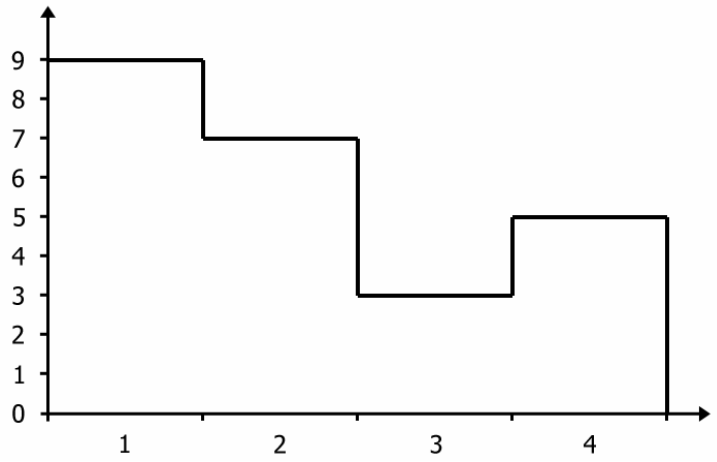
Algorithmen zur Erkennung: Wavelet Algorithmus

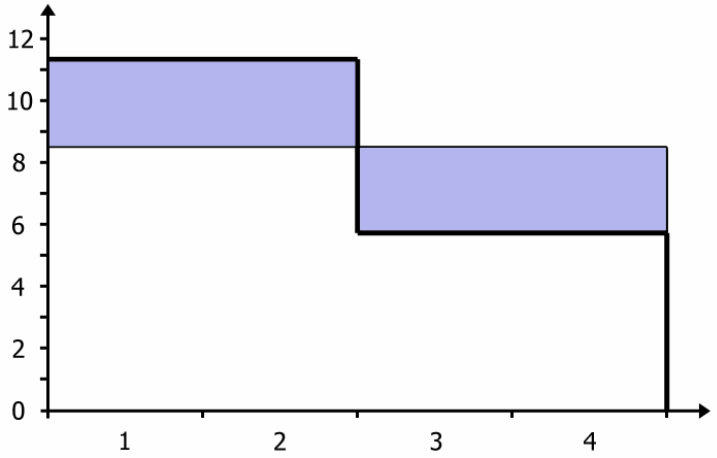
- Ausgangsfunktion

  (9,7,3,5)
- Wavet-Transformation

  (12, 4, √2, - √2)
- Erster Schritt der

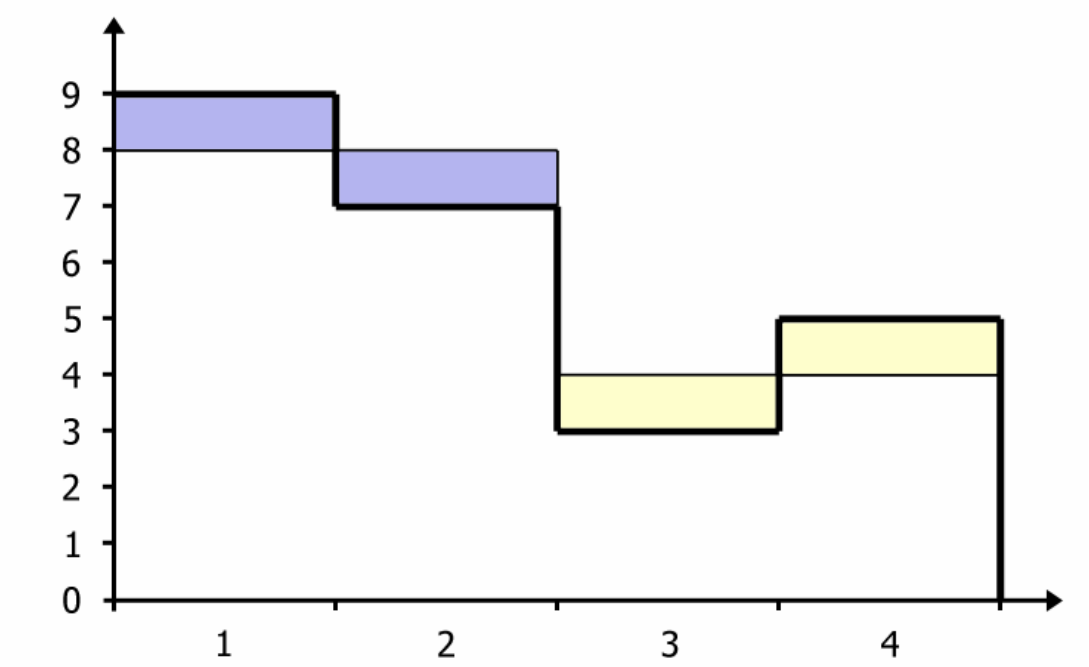
  Rekonstruktion





Algorithmen zur Erkennung: Wavelet Algorithmus

- Zweiter Schritt
- perfekte Rekonstruktion



Algorithmen zur Erkennung: Wavelet Algorithmus

- Filterbankimplementierung
- $\log_2(2^n)$ Schritte
- Summe der Pakete je Zeiteinheit
- 64 Zeiteinheiten

Algorithmen zur Erkennung: Paket-Zähler Algorithmus

- Blum / Song / Venkataraman (2004)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- geringe CPU-Last
- geringer Speicherverbrauch

Algorithmen zur Erkennung: Paket-Zähler Algorithmus

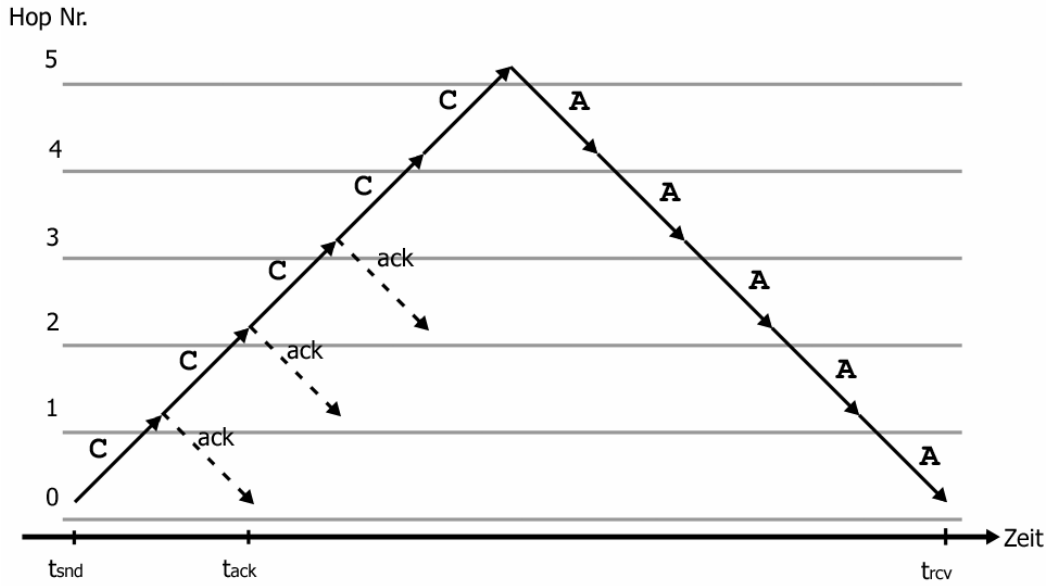
- Paketabhängigkeit zwischen Strömen
- Divergierende Paketanzahl
- Mathematische Sicherheiten

Algorithmen zur Erkennung: Hop-Count Algorithmus

- Yung (2002)
- timingbasierte Analyse
- passiv
- Online und Offline möglich
- ein Meßpunkt
- geringe CPU-Last
- geringer Speicherverbrauch

Algorithmen zur Erkennung: Hop-Count Algorithmus

- Delayed Acknowledgement
- Echo Reply
- Ab 2 Steps downstream



Algorithmen zur Erkennung: Vorauswahl

	Aktiv/ Passiv	**Online/ Offline**	**Inhalt/ Timing**	**Referenz-implement.**
*** On/Off Alg.**	*Passiv*	*On-/Offline*	*Timing*	*Ja (bro)*
Watermarking Alg.	*Aktiv*	*Online*	*Timing*	*Ja*
Abstandsalg.	*Passiv*	*On-/Offline*	*Timing*	*Ja*
*** IPD Algorithmus**	*Passiv*	*On-/Offline*	*Timing*	*Ja*
*** Wavelet Alg.**	*Passiv*	*On-/Offline*	*Timing*	*Nein*
*** Paket-Zähler Alg.**	*Passiv*	*On-/Offline*	*(Timing)*	*Nein*
Hop-Count Alg.	*Passiv*	*On-/Offline*	*Timing*	*Ja*

Vergleich der Algorithmen

- Datenumfeld
- Implementierungsumfeld
- Gemeinsamkeiten der Implementierung
- Vorfilterung
- Besonderheiten der Implementierung
- Ergebnisse

Vergleich der Algorithmen: Datenumfeld

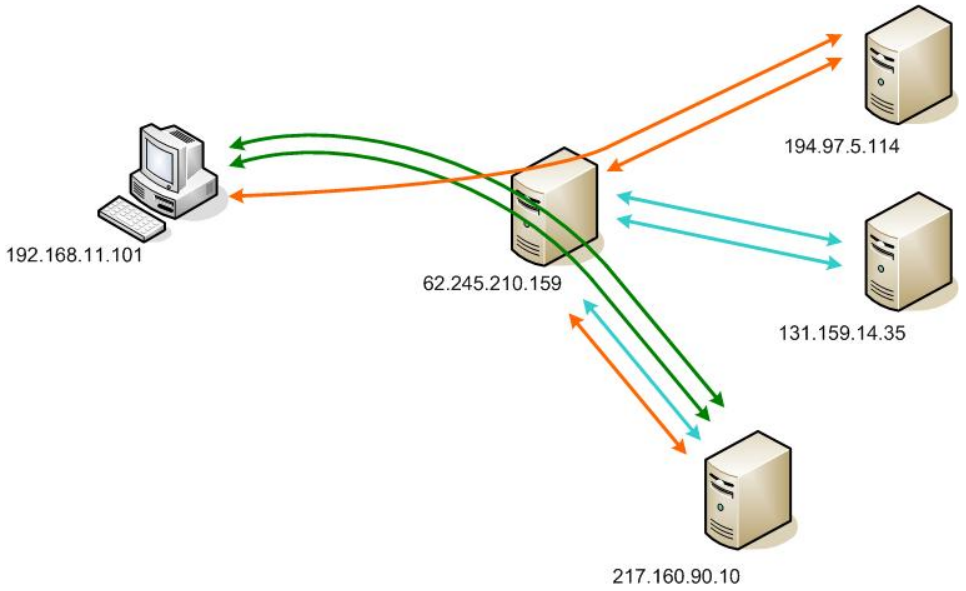
- MWN
- Traces
  - 10 GB Traces
  - Mit / Ohne leo.org
  - Login-Daten (SSH/Telnet/RLogin)
  - Kleinere Traces net-cs, net-login, halle-cs, halle-login

Vergleich der Algorithmen: Implementierungsumfeld

- Bro
- Waveletbibliothek

Vergleich der Algorithmen: Gemeinsamkeiten

- C++ Anteil
- Bro-Script
- Testparcours



Vergleich der Algorithmen: Vorfilterung

- 2-stufige Vorfilterung
  - libpcap-Filter
  - Analyse-Filter

Vergleich der Algorithmen: Besonderheiten

- IPD: Analyse bei Verbindungsende
- WT: CPU-lastigere Implementierung
- PZ: 1 Objekt je Verbindungspaar
- OO: basierend auf On-Periode

Vergleich der Algorithmen: Ergebnisse

- kein Algorithmus läuft auf großen Traces
- Problem: quadratische Laufzeit / Speicherverbrauch
- fast ausschließlich SSH-Verbindungen

Vergleich der Algorithmen: Ergebnisse On/Off Algorithmus

- Performance gut
- Kontroll-Stepping-Stone nicht gefunden
- Speicher-Bug

Vergleich der Algorithmen: Ergebnisse IPD Algorithmus

- Performance schlecht
- Kontroll-Stepping-Stone gefunden
- Problem: lange Verbindungen

Vergleich der Algorithmen: Ergebnisse Wavelet Algorithmus

- Performance gemischt
- Kontroll-Stepping-Stone nicht gefunden
- Problem: lange Verbindungen
- Abhängigkeit vom Schwellwert

Vergleich der Algorithmen: Ergebnisse Paketzähler

- Performance gut
- Kontroll-Stepping-Stone (nicht) gefunden
- Problem: Speicherverbrauch
- Extrem abhängig von der Parametern

Fazit

- Breitbandiger Einsatz problematisch
- Auf kleineren Netzen gut einsetzbar
- Vorfilterung sehr wichtig
- Problem: Legaler Einsatz