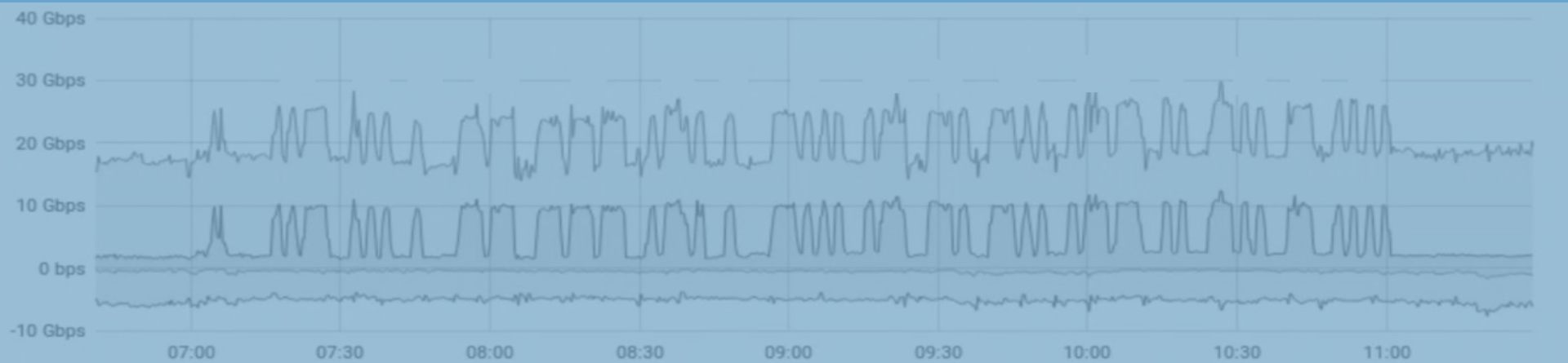
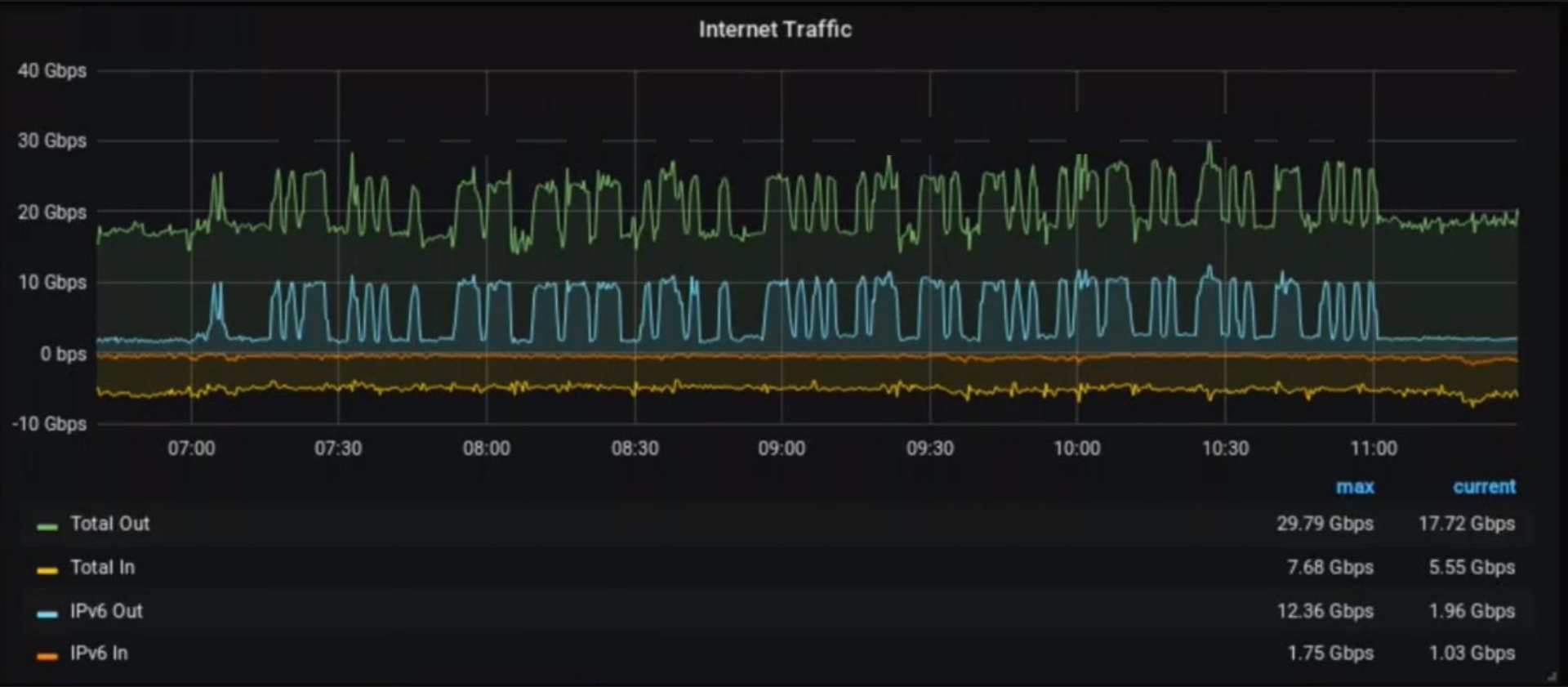
Covert Channels in Hash Chains




Internet Traffic
40 Gbps
30 Gbps
20 Gbps
10 Gbps
0 bps
-10 Gbps
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00
max
current
Total Out
29.79 Gbps
17.72 Gbps
Total In
7.68 Gbps
5.55 Gbps
IPv6 Out
12.36 Gbps
1.96 Gbps
IPv6 In
1.75 Gbps
1.03 Gbps

USE MORE BANDWIDTH, 35C3, [1]

Inhalt

Intro & Definitionen

Kategorisierungen von Covert Channels

Covert Channels in Hash Chains

Ausblick

Intro

Ein covert channel …

"[is a] channel[] intended for other uses, onto which the information is encoded" ([2], S. 615)

"is an unforeseen communication channel in a system design" ([3], S. 1)

"enable[s] stealthy communication over innocent appearing carriers" (ebd.)

"exploits the legitimate processes in a way that allows the signaling of hidden information via the shared resource" (ebd., S. 2)

→ Netzwerk-Steganographie

Schutz gegen Zensur/Überwachung

TOR, z. B. für Einreichungen bei Wikileaks [4, 5]

DNS-Tunnel, z. B. gegen "Captive Portals" [6, 7]

Aktionsraum für "bad actors"

Schadsoftware in den Metadaten von Steam-Profilbildern [8]

Okrum/Ketrican malware: Kommuniziert mit dem C&C-Server über die Cookies/Set-Cookies-Felder in HTTP-Requests [9]

Platinum malware: Befehle verschlüsselt in der Reihenfolge von HTML-Attributen (wie "align" oder "bgcolor") [10]

Kategorisierungen von Covert Channels

Übliche Bewertungen

Capacity: Wie viele Daten können verborgen gesendet werden?

Stealthiness: Wie gut verborgen ist die Kommunikation?

Robustness: Wie störungsanfällig ist ein Channel (insb. für noisy Channels)?

Steganographic Cost: Wie stark ist das Trägermedium durch die versteckten Informationen beeinträchtigt?

[3, 11, 12]

Aktiv

Covert Sender generiert seinen eigenen Traffic

Passiv

Covert Sender versteckt die Nachricht
in legitimem Traffic

Intentional

Covert Channel.
Channel zum absichtlichen Senden von Informationen

Unintentional

Side Channel.
Unabsichtliche Preisgabe von Informationen

Direct

Nachricht selbst wird verpackt

Indirect

z. B. via Server-Load

Noisy

störungsanfällig

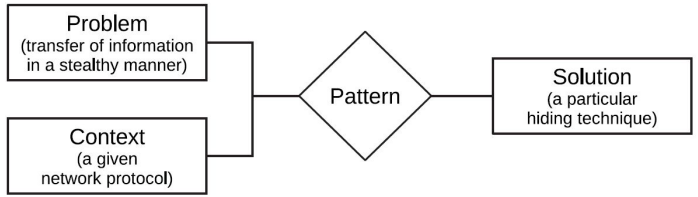
Noise-free

nicht störungsanfällig

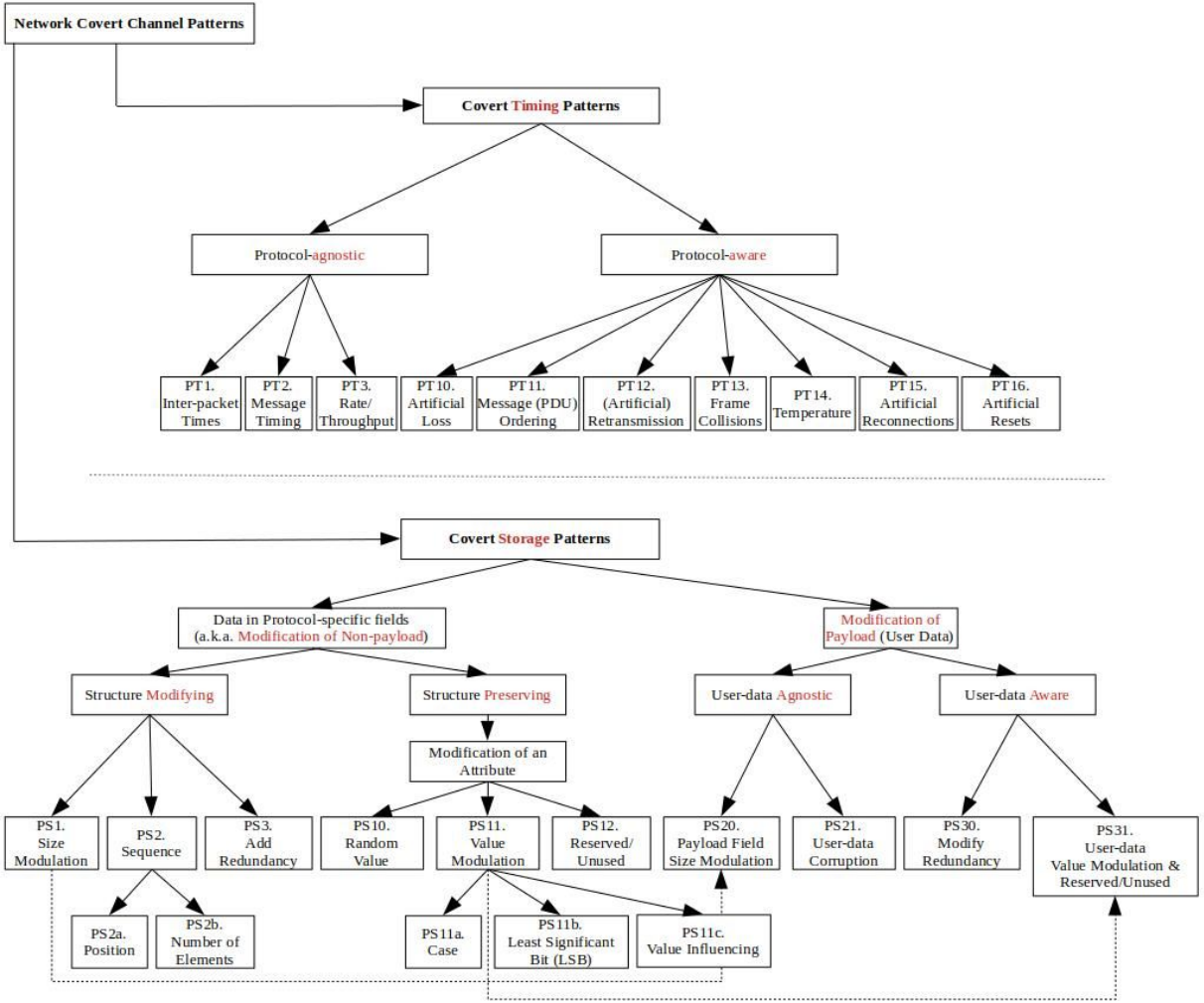
Hiding Patterns

Pattern:
Wiederverwendbare Lösung zu einem häufig auftretenden Problem in einem bestimmten Kontext

Hiding Patterns:
"Hiding Patterns describe the key idea of hiding techniques on an abstract level. They help cleaning up terminology, and can be used to form a taxonomy." [15]



[13, 14]


Network Covert Channel Patterns
Covert Timing Patterns
Protocol-agnostic
Protocol-aware
PT1. Inter-packet Times
PT2. Message Timing
PT3. Rate/ Throughput
PT10. Artificial Loss
PT11. Message (PDU) Ordering
PT12. (Artificial) Retransmission
PT13. Frame Collisions
PT14. Temperature
PT15. Artificial Reconnections
PT16. Artificial Resets
Covert Storage Patterns
Data in Protocol-specific fields (a.k.a. Modification of Non-payload)
Modification of Payload (User Data)
Structure Modifying
Structure Preserving
User-data Agnostic
User-data Aware
Modification of an Attribute
PS1. Size Modulation
PS2. Sequence
PS3. Add Redundancy
PS10. Random Value
PS11. Value Modulation
PS12. Reserved/ Unused
PS20. Payload Field Size Modulation
PS21. User-data Corruption
PS30. Modify Redundancy
PS31. User-data Value Modulation & Reserved/Unused
PS2a. Position
PS2b. Number of Elements
PS11a. Case
PS11b. Least Significant Bit (LSB)
PS11c. Value Influencing

[16]

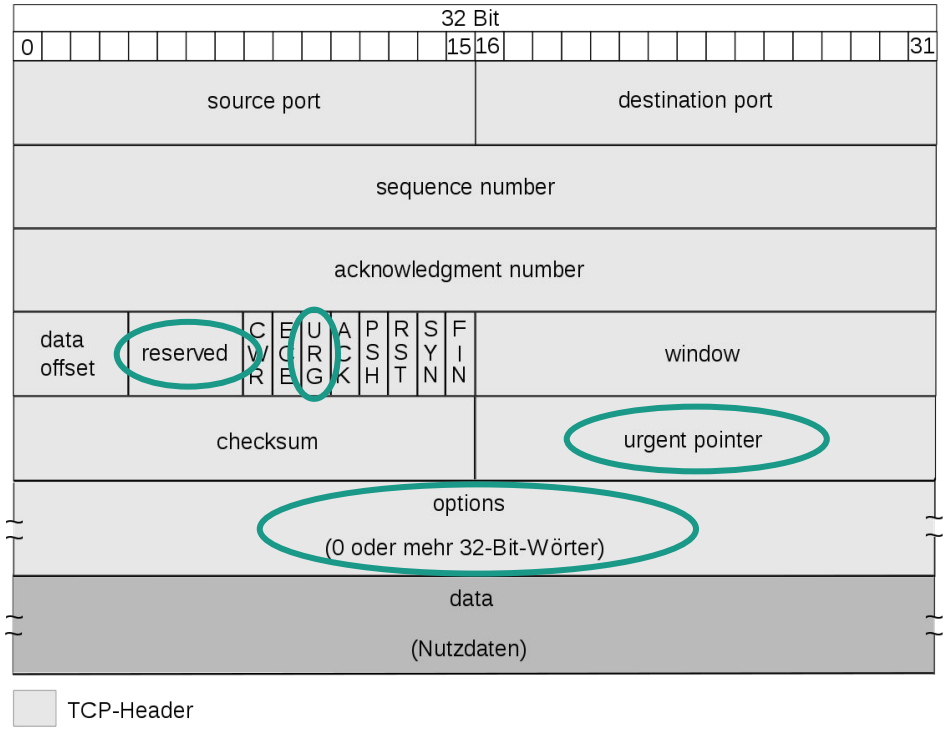
Covert **Storage** Channels

Bsp.: TCP header fields

"[E]mbedding secrets directly in the unused bits of the TCP could lead to a channel with a capacity of up to 4 bits/segment" ([17], S. 4)

Plus weitere Möglichkeiten.

Pattern: PS12 Reserved/Unused


32 Bit
0
15
16
31
source port
destination port
sequence number
acknowledgment number
data offset
reserved
CWR
ECE
URG
ACK
PSH
RST
SYN
FIN
window
checksum
urgent pointer
options
(0 oder mehr 32-Bit-Wörter)
data
(Nutzdaten)
TCP-Header

[18]

Covert **Timing** Channels

Pattern: PT3 Rate/Throughput

Gegenmaßnahmen

Was tun gegen CCs?

Detection

Statistische Auswertungen, Anomalie-Detektion, Application-Layer Firewalls

Elimination

Traffic-Normalisierung

Limitation

Buffering, Verzögerungen einfügen

Prevention

Protokoll-Blockierungen, Standardisierung

[11, 17]

Covert Channels in Hash Chains

Vorbereitungen

One-Time Passwords

"The Problem"

Passwort-Authentifizierung über einen unsicheren/abhörbaren Kanal

"The Solution"

Verwendung vieler verschiedener **EINMAL**-Passwörter
→ ein abgehörtes Passwort ermöglicht keine zukünftige Anmeldung

Generierung durch Verknüpfung von Berechnungen, die leicht in eine Richtung durchzuführen, aber nicht umkehrbar sind

[19]

Kryptografische Hash-Funktionen

Funktionen, die eine potenziell unendliche Definitionsmenge auf eine endliche Wertemenge abbilden

Nicht umkehrbar
(preimage resistance)

Schwache Kollisionsresistenz
(second preimage resistance)

Starke Kollisionsresistenz
(collision resistance)

[20]

Hash Chain

Mehrfach verkettete Hash-Funktion: Hash Chain $h(x),\ h(h(x)),\ h(h(h(x))),\ \ldots$

Wir schreiben z. B.:

$h(h(h(h(x)))) = h^4(x)$

OTPs mit Hash Chains nach Lamport

Hashfunktion: $h(x)$

Startwert der Hash-Kette (Seed): *s*

Aktuelles Einmal-Passwort: *p*

Wir nehmen an, unsere Hash-Kette sei 1000 Werte/Funktions-Anwendungen lang

Vorbereitung: Server erhält auf sicherem Weg vorab $h^{1000}(s)$

Erste Anmeldung: Client sendet $h^{1000-1}(s) = h^{999}(s) = p$ an den Server

Überprüfung: Server berechnet $h(p) \stackrel{?}{=} h^{1000}(s)$, denn $h\left(h^{999}(s)\right) = h^{1000}(s)$

Das lässt sich bei 1000 vorab berechneten Werten 1000 mal wiederholen

Nur der Client speichert die ganze Liste
(oder wiederholt die Berechnungskette jedes Mal bis zum aktuell geforderten Passwort)

Der Server speichert immer nur das letzte gültige Passwort

Für uns relevant:

Hash Chains sind weit verbreitet:
OTPs, Blockchains, Cryptocurrencies

Lange pseudo-random Strings werden regelmäßig verschickt. Ideal, um darin etwas zu verstecken

Covert Channels in Hash Chains (Keller/Wendzel 2021 [3])

Eigenschaften

Pattern: PS10 Random Value Covert Storage Channel

Semi-passive

Intentional

Direct

Noise-free

Plausible deniability

(Intrinsically) reversible

Könnte auch lokal und zeitversetzt verwendet werden. Meistens aber vernetzt und simultan.

Beispiel-Variante

Zu Illustrationszwecken: 8-Bit Pseudo-Random String `11110100`

Ein Zeichen pro Hash-Wert verschicken

x-tes Zeichen unseres Eingabe-Alphabets wird durch
Invertierung des x-ten Bits des Hash-Werts kodiert

Man in the Middle

Alice	Covert Sender	Covert Receiver	Bob
`11110100` $= x_{i+1}$	`11110100`	`11110100`	`11110100`

Überprüft ob

h(`11110100`) $\stackrel{?}{=} x_{i+2}$

Nächstes PW:

`10010011` $= x_i$

mit
h(`10010011`) = `11110100`

$$h(x_i) = x_{i+1}$$

CS

10010011 $= x_i$

10110011 $= T_{s_2}(x_i) = x_i'$

CR

10110011 $h(10110011) \overset{?}{=} x_{i+1}$

00110011 $h(00110011) \overset{?}{=} x_{i+1}$

11110011 $h(11110011) \overset{?}{=} x_{i+1}$

10010011 $h(10010011) \overset{!}{=} x_{i+1}$

Gegenmaßnahmen

Normierung

Im TCP-Header soll ein bestimmtes Feld nur Nullen enthalten.
Enthält nicht nur Nullen? Behebung ist einfach: mit Nullen überschreiben.

Hier haben wir es aber mit Hash-Werten zu tun

SHA-3

```
3a669155dd738f20 60cace2975ada58d
3ed250f1f9a5453e e7b709e0586b922e
c0224e3ae3454cfc 5d68ab9b145669a5
9659f675583b5f20 1e68a9bbb65a5954
```

= SHA-3("Keller/Wendzel 2021")

```
4d2739efaa599d72 be71b0c6f3f49cba
de915aa9fbcd1b0e 504865bae1430b6b
376cf677187bb884 bfa50c6c31111d2c
075715cfd440a048 8d7f44c8e6e3b65f
```

= SHA-3(SHA-3("Keller/Wendzel 2021"))

Randomness

Hash-Werte sammeln und einem Test auf Randomness unterziehen

Gegen-Gegenmaßnahme: Zu sendende Nachricht vorher noch verschlüsseln

Timing/Energieverbrauch

Wenn CR und B in derselben Maschine sind:
Timing und Energieverbrauch können beobachtet werden, um die mehrfache Ausführung der Hash-Funktion zu entdecken, während CR testet, welches Zeichen kodiert wurde.

Für Netzwerk-CC:
Möglichkeiten, den CC anhand der Paket-Laufzeit zu entdecken [21]

"Warden in the middle"

Wenn es eine Wächter-Institution mit Kenntnis der Logik der gesendeten Pakete (Hash Chain) gibt:

Dann kann die Warden zwei aufeinander folgende Hash-Werte mitlesen und mit Hilfe der bekannten Hash-Funktion auf Manipulationen überprüfen.

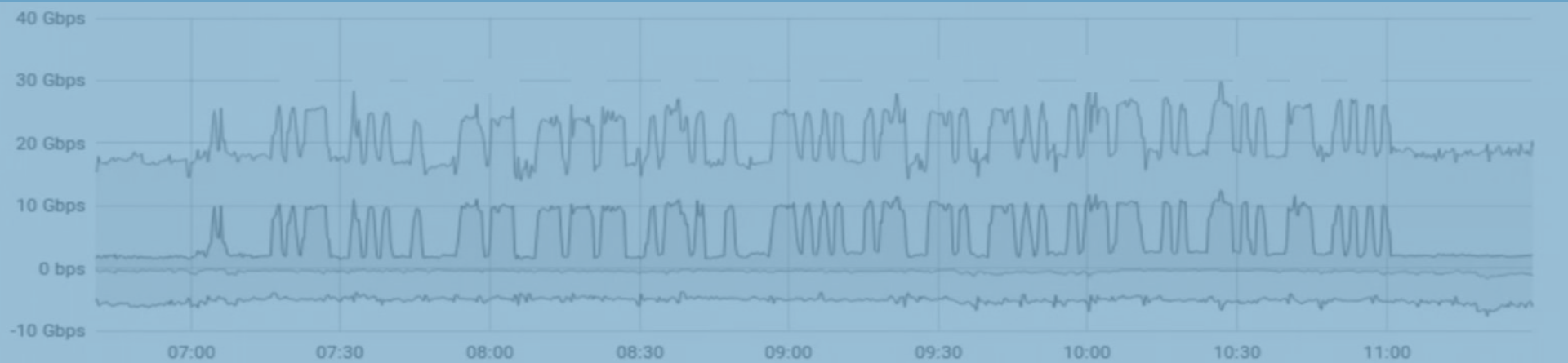
Ausblick

Reversible Covert Channels

Weitere Gegenmaßnahmen

New Media: Streaming/IoT/Blockchain

Vielen Dank.


40 Gbps
30 Gbps
20 Gbps
10 Gbps
0 bps
-10 Gbps
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
11:00

Literatur

[1] https://techpeople.dk/a-congress-running-six-communication-platforms/, abgerufen am 01.06.2022
[2] Lampson, B. A Note on the Confinement Problem. Communications of the ACM, 16, 10 (Oct. 1973), pp. 613-615.
[3] Keller, J.; Wendzel, S. Reversible and Plausibly Deniable Covert Channels in One-Time Passwords Based on Hash Chains. Appl. Sci. 2021, 11, 731. https://doi.org/10.3390/app11020731
[4] https://www.wikileaks.org/wiki/Wikileaks:Submissions, abrufen am 14.06.2022
[5] https://www.wikileaks.org/wiki/WikiLeaks:Tor, abrufen am 14.06.2022
[6] Plenz, J.: https://dnstunnel.de/, abrufen am 10.06.2022

[7] Kaminsky, D.: https://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Kaminsky.pdf, abrufen am 10.06.2022
[8] https://www.gdatasoftware.com/blog/2021/06/36861-malware-hides-in-steam-profile-images, abgerufen am 10.06.2022
[9] https://www.welivesecurity.com/wp-content/uploads/2019/07/ESET_Okrum_and_Ketrican.pdf, abgerufen am 10.06.2022
[10] https://securelist.com/platinum-is-back/91135/, abgerufen am 10.06.2022
[11] Tian, J.; Xiong, G.; Li, Z.; Gou, G. A Survey of Key Technologies for Constructing Network Covert Channel. Security and Communication Networks 2020. https://doi.org/10.1155/2020/8892896
[12] Wendzel, S. Network Covert Channels - A University-Level Course. Ch. 4. https://github.com/cdpxe/Network-Covert-Channels-A-University-level-Course/blob/master/slides/NIH_Ch4.pdf, abgerufen am 01.06.2022

[13] Wendzel, S.; Zander, S.; Fechner, B.; Herdin, C. Pattern-Based Survey and Categorization of Network Covert Channel Techniques. Comput. Surv. 2015, 47. http://dx.doi.org/10.1145/2684195
[14] Wendzel, S. et al., A Revised Taxonomy of Steganography Embedding Patterns. The 16th International Conference on Availability, Reliability and Security (ARES 2021). Association for Computing Machinery. https://doi.org/10.1145/3465481.3470069
[15] Wendzel, S.; Caviglione, L; Fechner, B. Poster: Steganographic Hiding Patterns: A Brief Review. Proc. EICC 2022. https://wendzel.de/dr.org/files/Papers/EICC22_poster.pdf, abgerufen am 05.06.2022
[16] http://ih-patterns.blogspot.com/p/test.html, abgerufen am 05.06.2022

[17] Caviglione, L. Trends and Challenges in Network Covert Channels Countermeasures. Appl. Sci. 2021, 11, 1641. https://doi.org/10.3390/app11041641
[18] https://de.m.wikipedia.org/wiki/Datei:TCP_Header.svg, abgerufen am 16.05.2022
[19] Lamport, L. Password Authentication with Insecure Communication. Communications of the ACM 24.11 (November 1981), 770-772. http://lamport.azurewebsites.net/pubs/password.pdf, abgerufen am 01.04.2022
[20] Wohlfeil, S. Kurstext Sicherheit im Internet, FernUniversität in Hagen
[21] Schmidbauer, T; Wendzel, S. Detection Of Computational Intensive Reversible Covert Channels Based On Packet Runtime. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 13(1):137-166, 2022. http://dx.doi.org/10.22667/JOWUA.2022.03.31.137