DIGITAL FORENSICS

Die schwierige Beurteilung von Cyber-Security-Fällen - Hackerangriff oder Fehlkonfiguration?

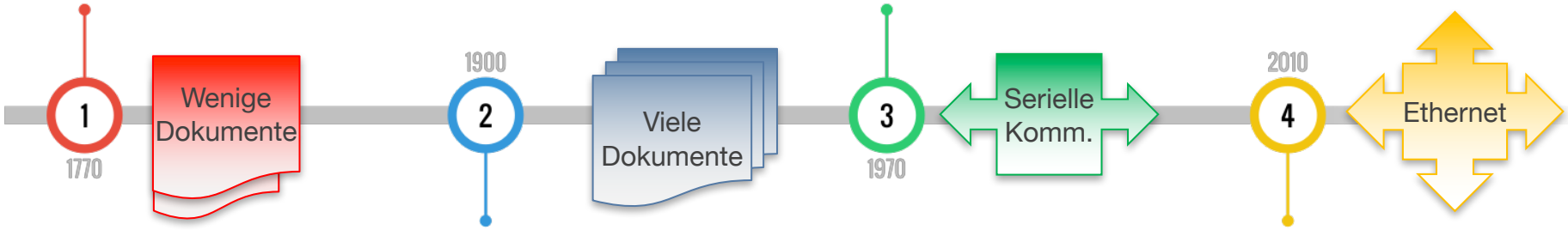
Messekongress Schadenmanagement 29. März 2017 in Leipzig

Dr. Frank Stummer | Geschäftsführer Digital Forensics GmbH

KOMMUNIKATIONSTECHNOLOGIEN IN DER INDUSTRIE

Mechanisierung
- Dampfmaschine
- Webstuhl
- Treibriemenanlage

Automatisierung
- Computer
- Roboter
- Sensoren



Massenproduktion
- Fließband
- Elekrizität
- Verkehrsmöglichkeiten

Industrie 4.0
- Cyber-Physikalische Systeme
- Internet of Things
- Internet of Services

VORTEILE UND HERAUSFORDERUNGEN DER INDUSTRIE 4.0

4

Industrie 4.0

- Cyber-Physikalische Systeme
- Internet of Things
- Internet of Services

Höhere Flexibilität und Effizienz der industriellen Produktion durch:

- vollständig vernetzte Industriesteuerungen,
- ermöglicht durch die Verschmelzung mit den IT-Systemen,
- basierend auf der Internet-Technologie (Ethernet & TCP/IP).

Mit einigen neuen Herausforderungen:

- höhere Komplexität, geringere Sichtbarkeit,
- mehr Fehlerquellen,
- neue und mehr Sicherheitsbedrohungen.

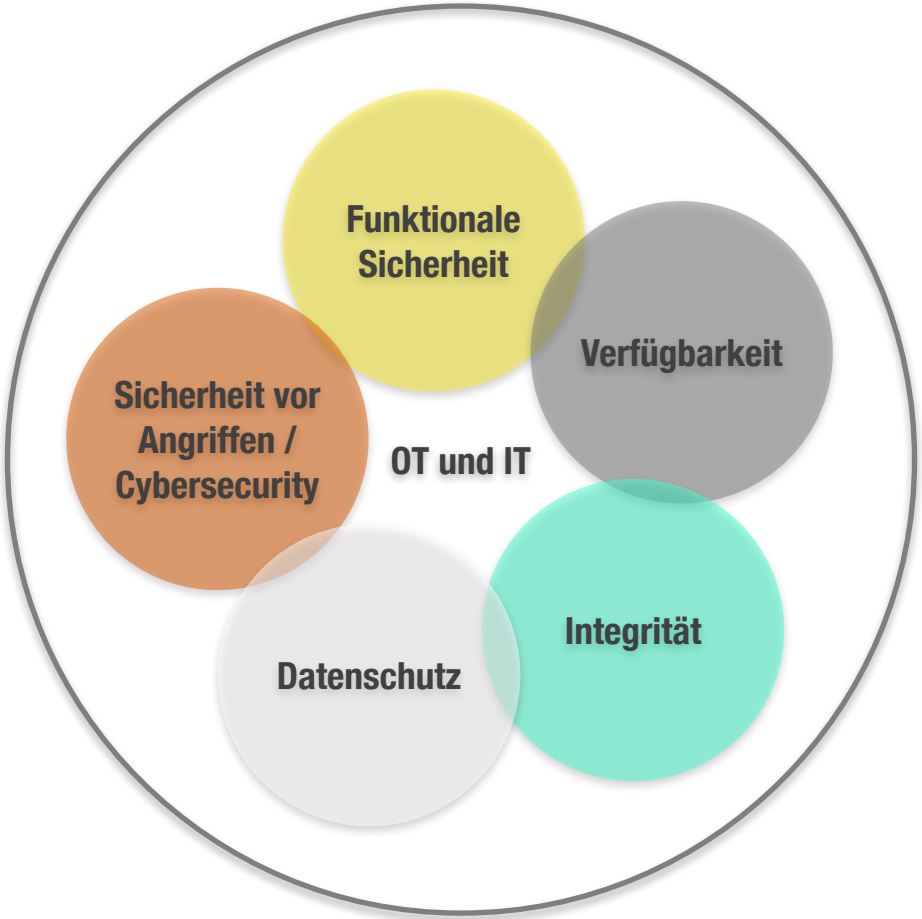
DIE ZIELE DER OT UND DER IT VEREINHEITLICHEN SICH

Produktion/ OT:

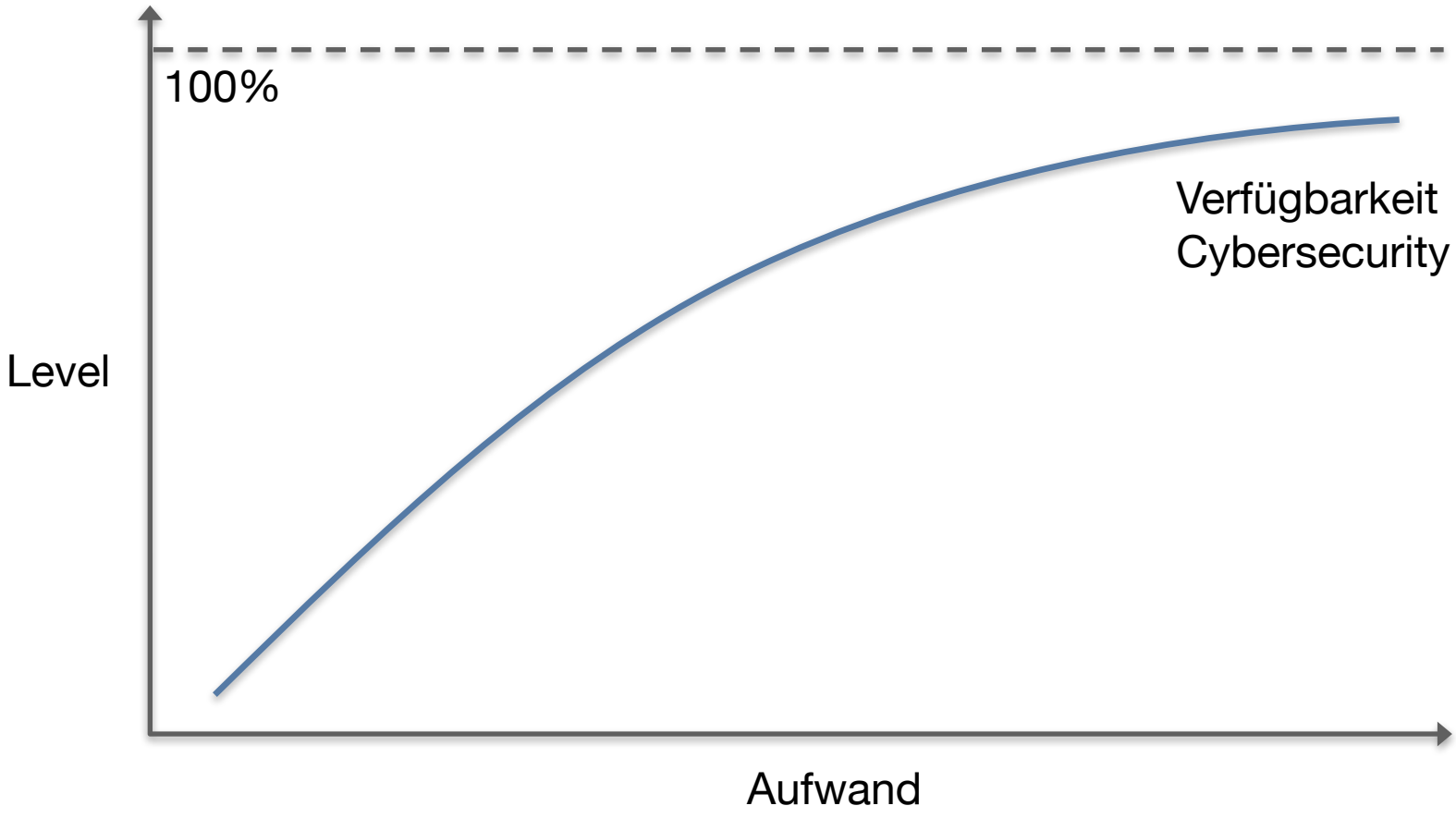
1. Funktionale Sicherheit
2. Verfügbarkeit
3. Integrität
4. Sicherheit vor Angriffen

IT Systeme:

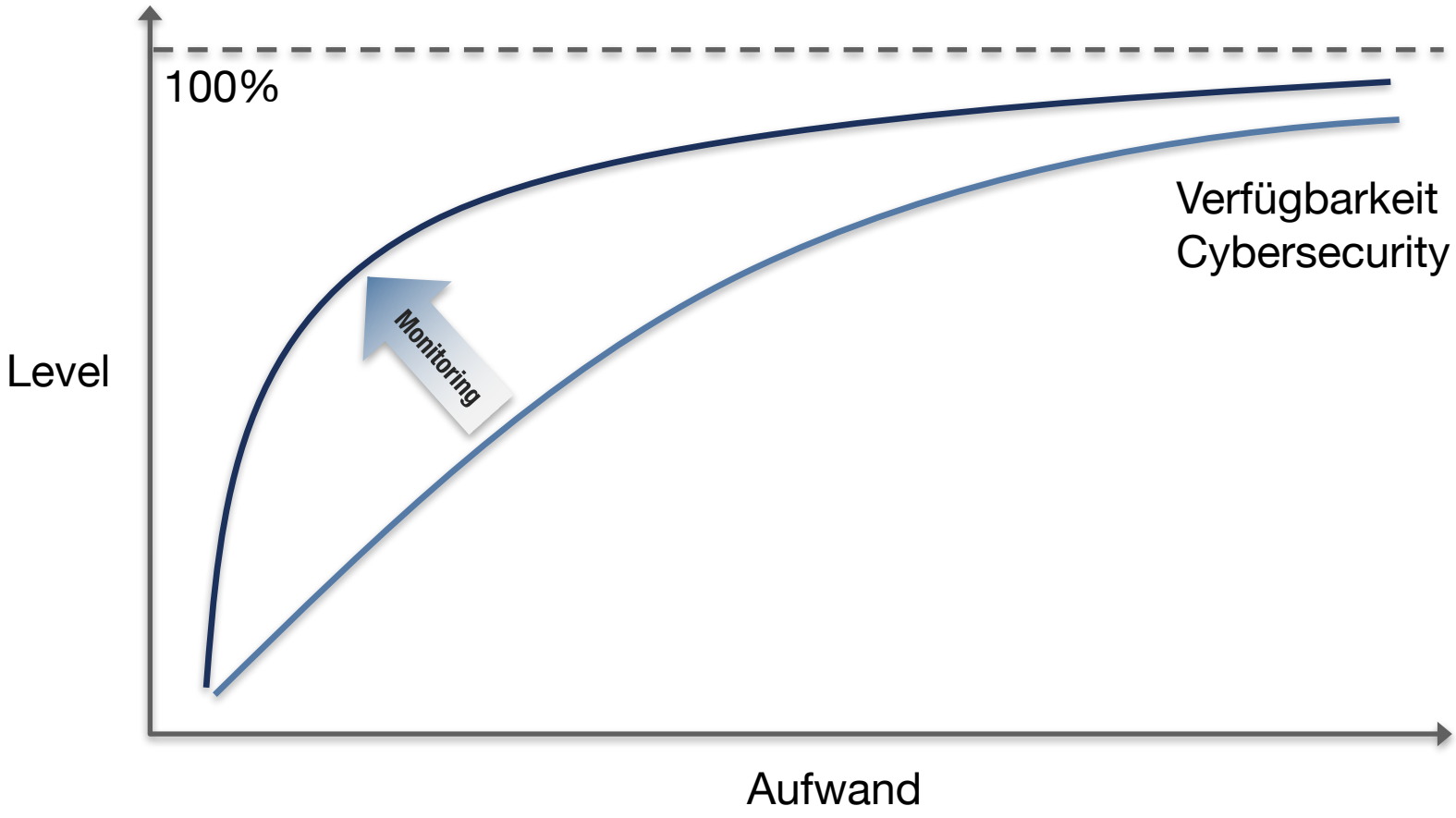
1. Sicherheit vor Angriffen
2. Datenschutz
3. Integrität
4. Verfügbarkeit



VERFÜGBARKEIT UND SICHERHEIT SIND ZWEI SEITEN DERSELBEN MEDAILLE



MONITORING DES NETZWERKVERKEHRS ERMÖGLICHT HÖHERE VERFÜGBARKEIT UND SICHERHEIT



DEFENSE IN DEPTH – SYSTEMATIK DER MASSNAHMEN ZUR SICHERING VON VERFÜGBARKEIT UND SICHERHEIT

Rhebo Industrial Protector v1.4.2-0-g850c37a

Help admin

Notifications (140473/22678)
Dashboards
Trend Analysis
Hosts
Conversations
Events
Administration

Notifications

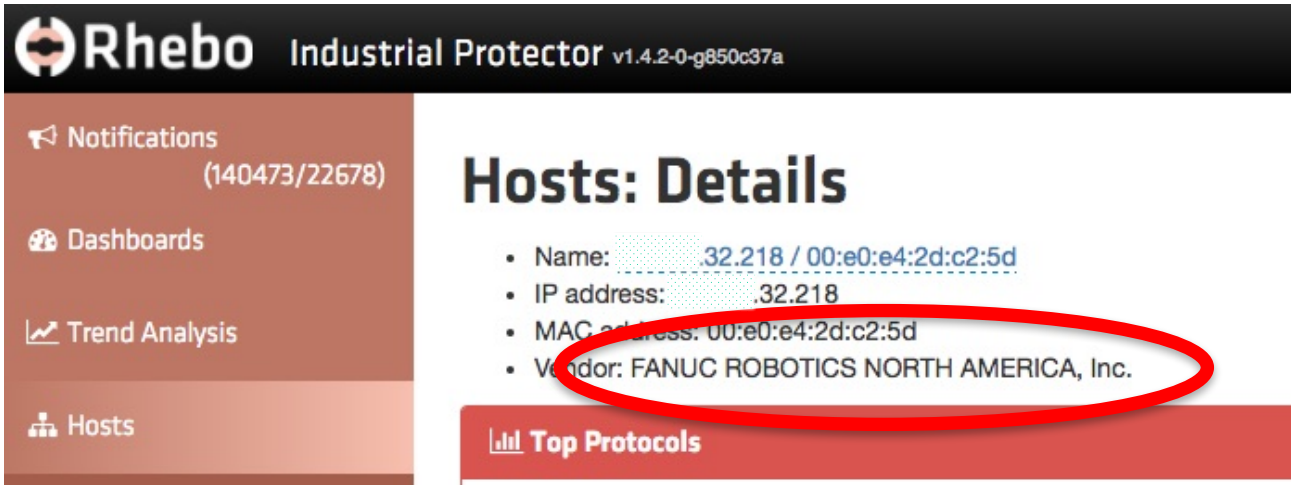
New Notifications (140473/22678) | Monitored (0/0) | Ignored (0/0)

Automatic refresh | Filter: | Cyclic message double, | Monitor | Ignore | Ignore displayed

Timestamp	Value	Hosts		Protocol
2016-08-08 22:54:23	(7)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM
2016-08-08 22:54:23	(7)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUST
2016-08-08 22:54:22	(7)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUS
2016-08-08 22:54:22	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUS
2016-08-08 22:54:22	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUST
2016-08-08 22:54:22	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM
2016-08-08 22:54:22	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP
2016-08-08 22:54:22	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61389
2016-08-08 22:54:05	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61377
		00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61376
		00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61375
		00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61351
		00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61350
		00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61349
2016-08-08 22:53:42		.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61348
2016-08-08 22:53:41	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61347
2016-08-08 22:53:41	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61346
2016-08-08 22:53:41	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61345
2016-08-08 22:53:41	(4)	.32.218 / 00:e0:e4:2d:c2:5d	.32.230 / 00:1b:1b:82:f5:c1	CUSTOM: TCP Port = 61344

= 61395
Port = 61394
Port = 61393
= 61392

Beispiel 1:
"Es schaut aus wie ein Port-Scan, und eventuell ein Angriff oder dessen Vorbereitung."



Beispiel 1:
"Es war aber die regelkonforme Steuerkommunikation eines eingesetzten Roboters."

snort.log.1474300585

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

vnc.client_message_type == 4

No.	Time	Source	Destination	Protocol	Length	Info
3152…	2016-09-19 18:26:37.935079000	.100.163	.100.105	VNC	62	Client key event
3153…	2016-09-19 18:26:38.026987000	.100.163	.100.105	VNC	62	Client key event
3153…	2016-09-19 18:26:38.077585000	.100.163	.100.105	VNC	62	Client key event
3154…	2016-09-19 18:26:38.169245000	.100.163	.100.105	VNC	62	Client key event
3155…	2016-09-19 18:26:38.220034000	.100.163	.100.105	VNC	62	Client key event
3155…	2016-09-19 18:26:38.301800000	.100.163	.100.105	VNC	62	Client key event
3155…	2016-09-19 18:26:38.352407000	.100.163	.100.105	VNC	62	Client key event
3156…	2016-09-19 18:26:38.434028000	.100.163	.100.105	VNC	30	Client key event
3156…	2016-09-19 18:26:38.494818000	.100.163	.100.105	VNC	62	Client key event
3157…	2016-09-19 18:26:38.576246000	.100.163	.100.105	VNC	2	Client key event
3165…	2016-09-19 18:26:40.273524000	.100.163	.100.105	VNC	2	Client key event
3165…	2016-09-19 18:26:40.377180000	.100.163	.100.105	VNC	2	Client key event
3165…	2016-09-19 18:26:40.415667000	.100.163	.100.105	VNC	2	Client key event
3166…	2016-09-19 18:26:40.510036000	.100.163	.100.105	VNC	2	Client key event
3166…	2016-09-19 18:26:40.570061000	.100.163	.100.105	VNC	2	Client key event
3167…	2016-09-19 18:26:40.673487000	.100.163	.100.105	VNC	2	Client key event
3167…	2016-09-19 18:26:40.687564000	.100.163	.100.105	VNC	2	Client key event
3167…	2016-09-19 18:26:40.687571000	.100.163	.100.105	VNC	62	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.687576000	.100.163	.100.105	VNC	30	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.687588000	.100.163	.100.105	VNC	130	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.765625000	.100.163	.100.105	VNC	62	Client key event
3167…	2016-09-19 18:26:40.765634000	.100.163	.100.105	VNC	62	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.765635000	.100.163	.100.105	VNC	130	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.765641000	.100.163	.100.105	VNC	130	[TCP Fast Retransmission] Client key event
3167…	2016-09-19 18:26:40.910464000	.100.163	.100.105	VNC	62	Client key event
3168…	2016-09-19 18:26:40.989762000	.100.163	.100.105	VNC	62	Client key event

```
▶ Frame 316522: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
▶ Ethernet II, Src: Dell_66:01:04 (00:19:b9:66:01:04), Dst: TexasIns_b1:57:ff (50:72:24:b1:57:ff)
▶ Internet Protocol Version 4, Src: .100.163, Dst: .100.105
▶ Transmission Control Protocol, Src Port: 1070 (1070), Dst Port: 5901 (5901), Seq: 17993, Ack: 3191796, Len: 8
▼ Virtual Network Computing
  ▼ Client Message Type: Key Event (4)
      Key down: Yes
      Padding
      Key: s (0x00000073)
```

```
0000  50 72 24 b1 57 ff 00 19  b9 66 01 04 08 00 45 00   Pr$.W... .f....E.
0010  00 30 15 28 40 00 80 06  9b 42 c0 a8 64 a3 c0 a8   .0.(@... .B..d...
0020  64 69 04 2e 17 0d 69 55  03 c8 72 90 16 37 50 18   di....iU ..r..7P.
0030  fc 00 53 d2 00 00 04 01  00 00 00 00 00 73         ..S..... .....s
```

snort Packets: 1211892 · Displayed: 312 (0.0%) · Load time: 0:4.832

Beispiel 2:
“VNC wurde zur Remote-Wartung eingesetzt – durchaus erlaubt.”

```
3167…  2016-09-19 18:26:40.765641000  [redacted].100.163  [redacted].100.105
3167…  2016-09-19 18:26:40.910464000  [redacted].100.163  [redacted].100.105
3168…  2016-09-19 18:26:40.989762000  [redacted].100.163  [redacted].100.105

▶ Frame 316522: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
▶ Ethernet II, Src: Dell_66:01:04 (50:19:b9:66:01:04), Dst: TexasIns_b1:57:ff (50:72:24:b1:57:ff)
▶ Internet Protocol Version 4, Src: [redacted].100.163, Dst: [redacted].100.105
▶ Transmission Control Protocol, Src Port: 1070 (1070), Dst Port: 5901 (5901), Seq: 17993, Ack: 3191796, Len: 8
▼ Virtual Network Computing
  ▼ Client Message Type: Key Event (4)
      Key down: Yes
      Padding
      Key: s (0x00000073)
```

Beispiel 2:
“Es war jedoch ein Sabotage-Angriff eines Innentäters.”

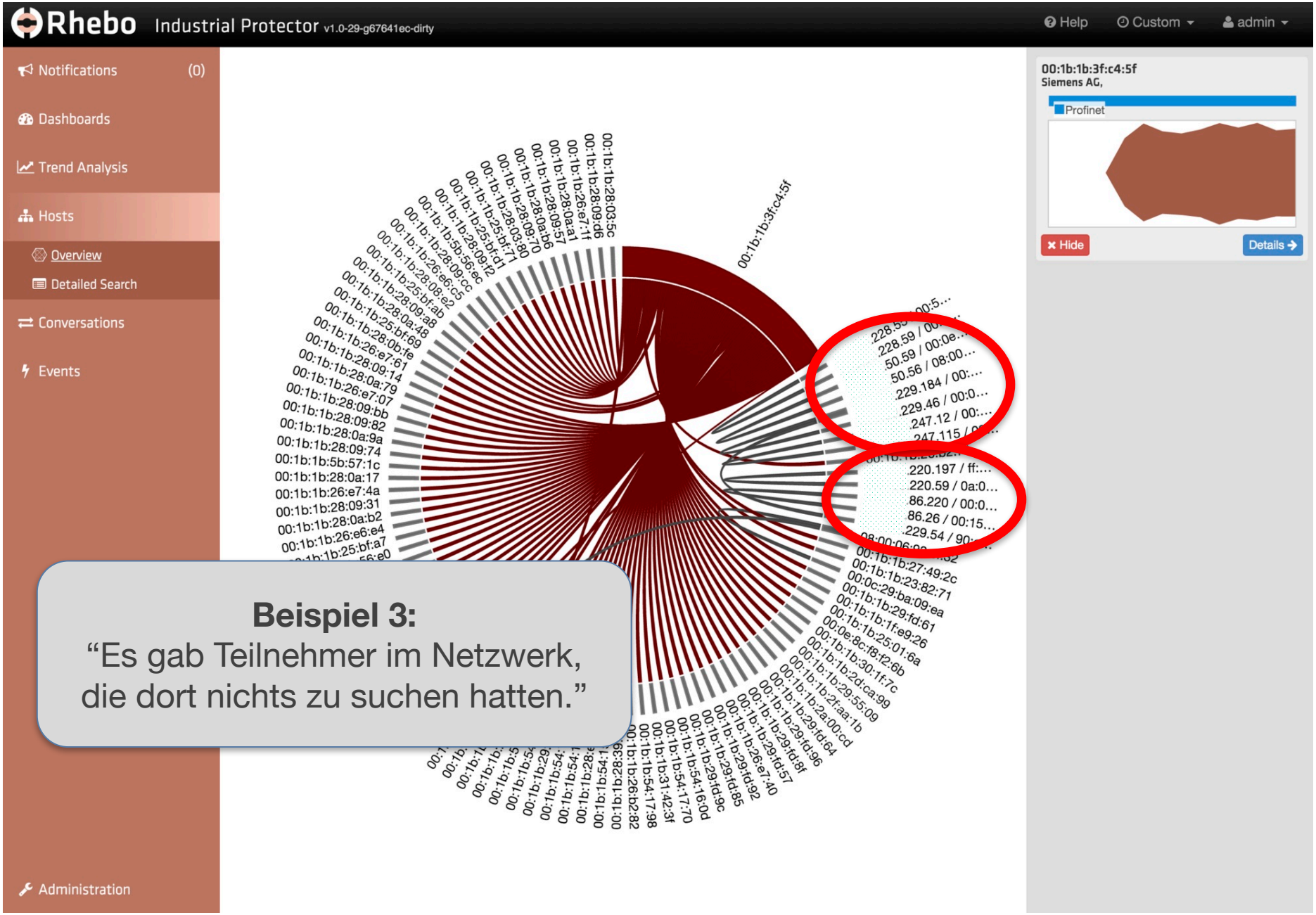
```python
def parse_vnc(raw_payload):

        decoded = binascii.hexlify(raw_payload).decode('latin-1')

        if len(decoded) == 16 and decoded[1] == '4' and decoded[3] == '0':
            try:
                if decoded[-4:] in VNC_CODES:
                    key_stroke = VNC_CODES[decoded[-4:]]
                    print('_{0}'.format(key_stroke))
                else:
                    key_stroke = binascii.unhexlify(decoded[-8:]).decode('utf-8')[-1]
                    print(key_stroke, end='', flush=True)

            except UnicodeDecodeError:
                print('\\n')


def callback(rawPacket):
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
import binascii
from scapy.all import sniff, TCP, Raw


VNC_CODES = {'ff08': 'Backspace',
             'ff09': 'Tab',
             'ff0d': 'Return/Enter',
             'ff1b': 'Escape',
             'ff63': 'Insert',
             'ffff': 'Delete',
             'ff50': 'Home',
             'ff57': 'End',
             'ff55': 'Page-Up',
             'ff56': 'Page-Down',
             'ff51': 'Left',
             'ff52': 'Up',
             'ff53': 'Right',
             'ff54': 'Down'}

PORT_RANGE_TCP = list(range(5901, 5999))
PORT_RANGE_HTTP = list(range(5801, 5899))


def parse_vnc(raw_payload):

        decoded = binascii.hexlify(raw_payload).decode('latin-1')

        if len(decoded) == 16 and decoded[1] == '4' and decoded[3] == '0':
            try:
                if decoded[-4:] in VNC_CODES:
                    key_stroke = VNC_CODES[decoded[-4:]]
                    print('_{0}'.format(key_stroke))
                else:
                    key_stroke = binascii.unhexlify(decoded[-8:]).decode('utf-8')[-1]
                    print(key_stroke, end='', flush=True)

            except UnicodeDecodeError:
                print('--')


def callback(rawPacket):

    if rawPacket.haslayer(TCP):
        srcPort = rawPacket.getlayer(TCP).sport
        dstPort = rawPacket.getlayer(TCP).dport

        if (srcPort in PORT_RANGE_TCP
                or dstPort in PORT_RANGE_TCP):

            if rawPacket.haslayer(Raw):
                parse_vnc(rawPacket.getlayer(Raw).load)


sniff(offline=sys.argv[1], store=0, prn=callback)
```

Beispiel 2:
"… der nachweislich (hier ein Skript zur Analyse) …"

```
➜ ./example_vnc_decoding.py traces/snort.log.1474300585
cc---
---
_Up
_Up
_Return/Enter
_Return/Enter
cc---
---
cc---
---
cccccccccccccccccccccc---
---
pppppp---
---
pppppppppppp---
---
ppss  aauuxx  ---
---
---
---
||---
---
---
---
  ggrreepp  uussttaarrtt_Return/Enter
_Return/Enter
pppppppp---
---
pppppppppppp---
---
sshhuuttddoowwnn  --hh  nooww_Return/Enter
_Return/Enter
```

Beispiel 2:
"… gezielt und konzertiert Geräte heruntergefahren hat."



Beispiel 3:
"Es gab Teilnehmer im Netzwerk, die dort nichts zu suchen hatten."

2015-10-22 11:40:33 (4) (4) .2.22 / 00:07:7c:1a:61:83 ⇄ .88.60 / 00:90:e8:26:40:23 ICMP Download PCAP

IP address: .88.60
Function type: INVALID

IP address: .2.22
IP address: .88.60
Function type: INVALID

2016-05-20 03:33:32 (6) (6) (2) .170.253 / d6:c3:d6:c7:63:f2 ⇄ .111.238 / 00:25:90:0d:23:63 S7 Download PCAP

MAC address: **d6:c3:d6:c7:63:f2**
Message: **TCP checksum error**

MAC address: **d6:c3:d6:c7:63:f2**
Message: **S7 error**

IP address: .170.253
Function type: OPEN CONNECTION

2015-10-22 08:17:01 (6) (6) .2.166 / 00:07:7c:1a:61:83 ⇄ .88.60 / 00:90:e8:26:40:23 Modbus Download PCAP

IP address: .2.166
Function type: W COIL

2015-10-22 12:33:39 (2) (2) .2.2 / 00:07:7c: load PCAP

IP address: .88.55
Function type: TIMEOUT

IP address: .2.2
IP address: .88.55
Function type: TIMEOUT

Weitere Beispiele:
"Invalide Kommunikation, S7-Fehler, nicht gewünschte Modbus-Kommunikation, Timeouts."

BESTEN DANK FÜR IHRE AUFMERKSAMKEIT

Dr. Frank Stummer
Telefon: +49 176 310 461 45
frank.stummer@digitalforensics.de

www.digitalforensics.de

EGAL WIE GROSS DER HEUHAUFEN IST - WIR FINDEN DIE NADEL DARIN.
EFFIZIENT. SCHNELL. BEWEISSICHER.