

Praxisseminar SS 2020

Angriffssimulation und strukturierte Datenerfassung

Johannes Seitz

Lehrstuhl für Wirtschaftsinformatik I

Fakultät für Wirtschaftswissenschaften



Universität Regensburg



COVID-19 Cyber Threat Assessment

Cyberangriffe folgen der Entwicklung von COVID-19

[Schmitz2020]



Während Coronavirus-Pandemie: Cyberangriff legt tschechisches Krankenhaus lahm

Regierung und Industrie betroffen

Australien meldet massiven Cyberangriff

Stand: 19.06.2020 09:24 Uhr

Australien ist laut Premier Morrison zur Zielscheibe eines groß angelegten Cyberangriffs durch einen anderen Staat geworden. Wer hinter den Attacken steckt, sagte er nicht - doch es gebe nicht allzu viele Länder, die infrage kämen.

[Holland2020]

[Bodewein2020]

Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

4 Aufarbeitung der Daten

5 Zusammenfassung und Fazit

6 Quellenverzeichnis

- Aufbau eines Digitalen Zwillings als Abbild eines realistischen Industriesettings
- Angriffssimulation auf den Digitalen Zwilling und Aufzeichnen des angefallenen Netzwerkverkehrs
- Strukturierte Aufarbeitung der Daten des Mitschnitts

Ablauf des Seminars

Aufgaben	Zeit geplant in h:	Mrz 20		Apr 20					Mai 20				Jun 20				Jul 20			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Literaturrecherche	10																			
Einarbeitung Digital Twin	20																			
Einarbeitung Angriffe	5																			
Einarbeitung STIX/ JSON	15																			
Entwicklung Digital Twin	40																			
Simulation Angriff	20																			
Aufarbeitung der Daten	40																			
Vorbereitung der Präsentation	10																			
Textbearbeitung	20																			
Gesamt	180																			

- Start im März mit Literaturrecherche
- Anhaltspunkt für Zeitplanung war die Anzahl der Credits (30 x 6 ECTS)
- Einarbeitung hat viel Zeit beansprucht

- Versionierung: Github
- Programmierung: PyCharm, Notepad++
- Programmiersprache: Python, JSON (STIX)
- Virtualisierung: Oracle Virtualbox, Ubuntu 18.04



Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

4 Aufarbeitung der Daten

5 Zusammenfassung und Fazit

6 Quellenverzeichnis

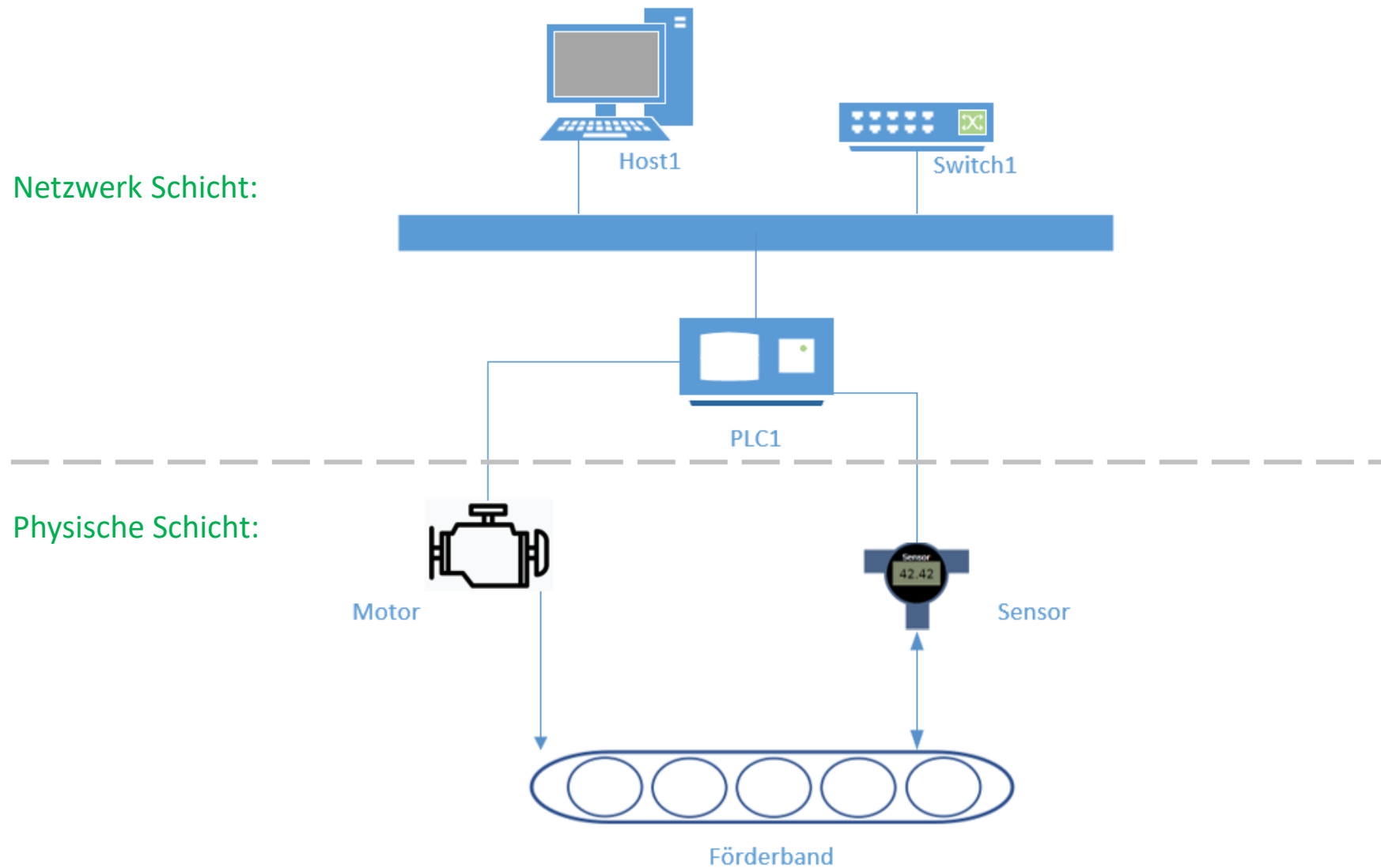
„Digital Twin“ – was ist das und wie wird es erzeugt

- Digital Twin (DT, auch Digitaler Zwilling): Die digitale Spiegelung eines Realwelt-Systems (z.B. Wasseraufbereitungsanlage)
 - Ermöglichen das Überwachen, Simulieren, Optimieren und die Prognose von CPS (cyber physical systems) ohne auf das eigentliche System einzugreifen [Eckhart2018, 1]
 - Meistens sind echte Systeme für Security-Forscher nicht offen zugänglich [Antonioli2015, 1]
 - Für IT-Security relevant: Man kann das digitale Spiegelbild unter realen Bedingungen testen, ohne das reale System zu gefährden („Spielwiese für Security-Experten“)
- Wie wird ein DT erstellt:
 - Entscheidung für die Entwicklung eines DT, der ein ICS System darstellen soll
 - Auswahl des Frameworks (z.B. <https://github.com/hslatman/awesome-industrial-control-system-security>)
 - Entscheidung für MiniCPS, da zu diesem Framework ein Paper der Entwickler existiert, dass das System relativ ausführlich erklärt und weil Code in Python geschrieben wird [Antonioli2015]
 - MiniCPS arbeitet auf Basis von Mininet
 - Netzwerk Emulator, wodurch man ein realistisches virtuelles Netzwerk mit Switches und Hosts erzeugen kann (lässt sich mit Python entwickeln)

Idee zum Digital Twin – Teil I

- Industrial Control System (ICS) auf Basis von Mininet und MiniCPS
- Das Industrial Control System ist Bestandteil eines Abfüllanlagensystems, dass bei Auslieferung an den Kunden bei einer Drittfirma eingerichtet wird
- Um die Komplexität für das Praxisseminar einzuschränken => Konzentration auf ein Teilsystem
- Teilsystem stellt die Bedienung eines Förderbandes dar
- Förderband verfügt über einen Motor und die Geschwindigkeit wird durch einen Sensor überwacht
- Sensor und Motor sind über einen PLC (programmable logic controller) mit dem Netzwerk verbunden
- Ein HMI (human machine interface) kann auf die Daten des PLCs zugreifen und diese auch verändern
- Das Netzwerk besteht aus einem Switch, dem PLC und einem Host auf dem das HMI-Programm läuft

Idee zum Digital Twin – Teil II



- Präsentation – aus Zeitgründen nur Einzelausschnitte
 - Topologie (mininet-Code)
 - HMI
- Gesamter Code unter <https://github.com/EnjoyFitness92/Praxisseminar-SS2020> einsehbar

Code Digital Twin I – Mininet Topologie

```
class CbTopo(Topo):  
  
    """Fließband 1 plc + 1 Host mit HMI + 1 Angreifer + 1 Switch"""  
  
    def build(self):  
  
        switch = self.addSwitch('s1')  
  
        plc1 = self.addHost(  
            'plc1',  
            ip=IP['plc1'] + NETMASK,  
            mac=MAC['plc1'])  
        self.addLink(plc1, switch)  
  
        host1 = self.addHost(  
            'host1',  
            ip=IP['host1'] + NETMASK,  
            mac=MAC['host1'])  
        self.addLink(host1, switch)  
  
        attacker = self.addHost(  
            'attacker',  
            ip=IP['attacker'] + NETMASK,  
            mac=MAC['attacker'])  
        self.addLink(attacker, switch)  
        Praxisseminar_test_logger.info('PLC1, Host1 und Attacker')
```

Code Digital Twin II – HMI Geschwindigkeit

```
# Geschwindigkeit einstellen
elif eingabe == 2:
    Praxisseminar_test_logger.debug("User befindet sich in der zweiten if-Abfrage")
    motor = self.receive(MOTOR, PLC1_ADDR)
    print "DEBUG plc1 erhaelt motor: " + motor
    Praxisseminar_test_logger.info('Motor erhaelt von PLC1_ADDR: ' + motor)

    # siehe Eingabe '1'

    if motor == '1':
        Praxisseminar_test_logger.info("Der Motor ist an")
        sensor = self.receive(SENSOR, PLC1_ADDR)
        print 'DEBUG plc1 motor: An mit der Geschwindigkeit' + sensor
        Praxisseminar_test_logger.info('Sensor erhaelt von PLC1_ADDR: ' + sensor)

        # Wollen Sie die Geschwindigkeit veraendern? Wie hoch soll die Geschwindigkeit sein (Rahmen der Geschwindigkeit anpassen)
        change = raw_input("Wollen Sie die Geschwindigkeit veraendern? J/N")
        Praxisseminar_test_logger.debug("Der User hat folgendes eingegeben: %s" % change)

        if change == "J" or change == "j":
            new_vel = float(raw_input("Geben Sie die neue Geschwindigkeit ein: "))
            Praxisseminar_test_logger.debug("Der User hat folgendes eingegeben: %s" % str(new_vel))
            self.send(SENSOR, new_vel, PLC1_ADDR)
            print 'DEBUG plc1 motor: An mit neuer Geschwindigkeit' + str(new_vel)
            Praxisseminar_test_logger.info('HMI sendet folgende SENSOR-Daten an PLC1_ADDR: ' + str(new_vel))

        elif change == "N" or change == "n":
            Praxisseminar_test_logger.debug("Elif-Abfrage wurde erreicht weil User ein N/n eingegeben hat")
            continue

    elif motor == '0':
        print 'DEBUG plc1 motor: Aus'
        Praxisseminar_test_logger.info("Der Motor ist aus")
    print
```

Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

4 Aufarbeitung der Daten

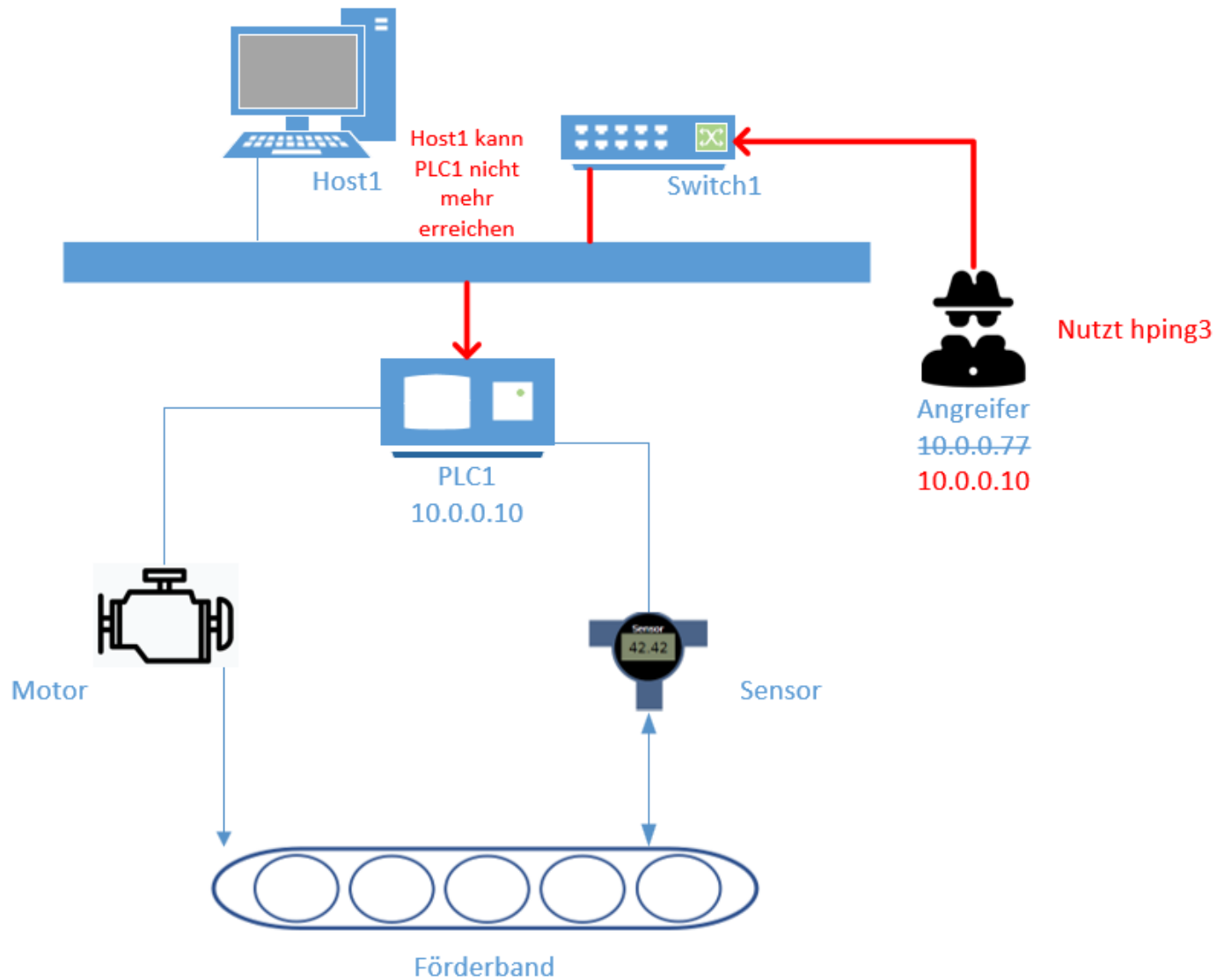
5 Zusammenfassung und Fazit

6 Quellenverzeichnis

Annahmen zum Angriff

- Angreifer ist ein Innentäter und verfügt über Detailwissen zum System
- Weiß um die Schwachstellen des Systems und umgeht Hindernisse
- Verfügt über Expertenwissen und weiß wie er sich unerkannt im System aufhalten kann
- Motivation: Rache wegen einer anstehenden Entlassung oder Bestechung durch eine konkurrierende Firma

Erklärung zum Angriff – DOS Attacke



Live-Simulation eines DOS- Angriffs auf den Digital-Twin

Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

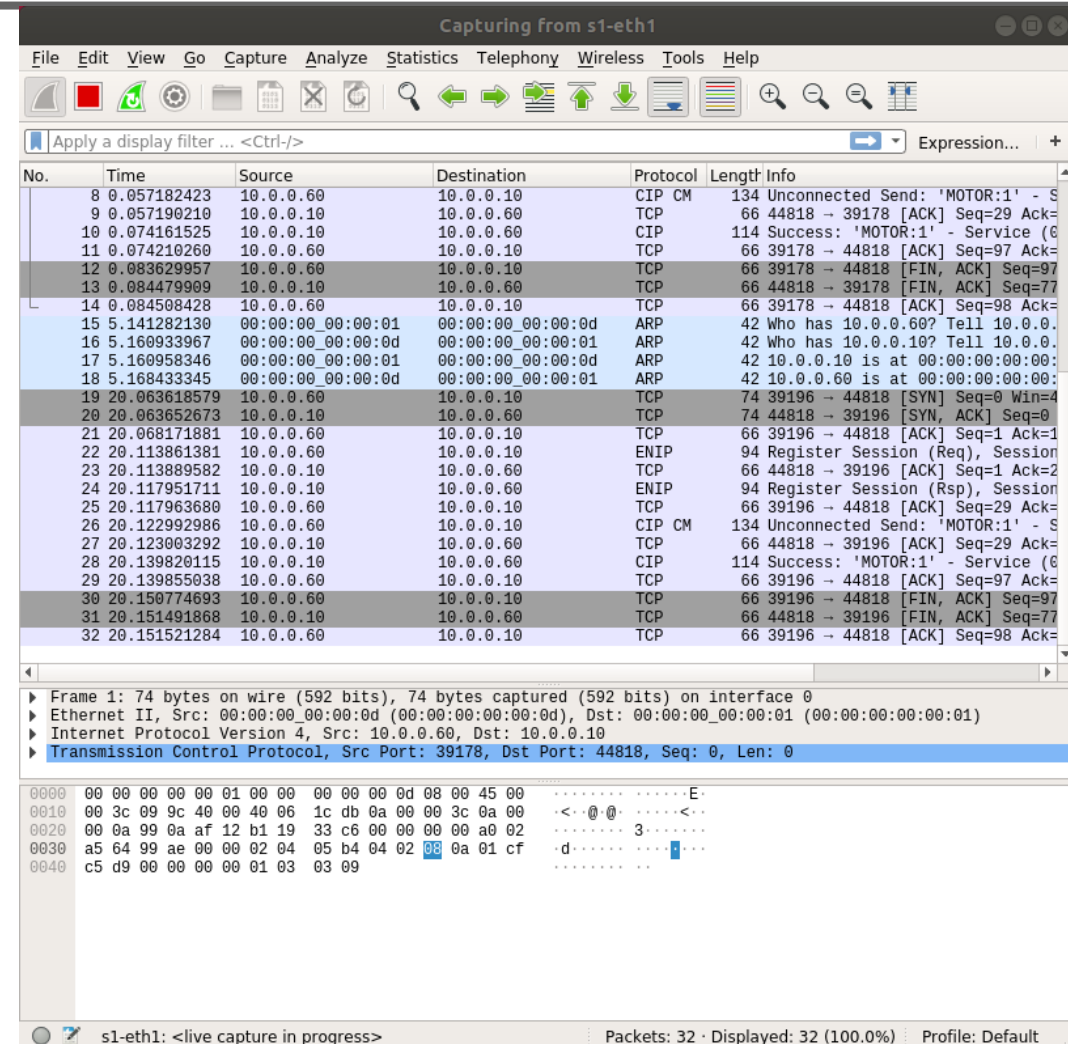
4 Aufarbeitung der Daten

5 Zusammenfassung und Fazit

6 Quellenverzeichnis

Woher kommen die Daten?

- Aufzeichnung der Daten mittels Wireshark
- Wireshark zeichnet alle Pakete auf, die über den Switch kommuniziert werden
- Daten werden in JSON-Format abgespeichert
- Was passiert dann?



Capturing from s1-eth1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
8	0.057182423	10.0.0.60	10.0.0.10	CIP CM	134	Unconnected Send: 'MOTOR:1' - S
9	0.057190210	10.0.0.10	10.0.0.60	TCP	66	44818 → 39178 [ACK] Seq=29 Ack=
10	0.074161525	10.0.0.10	10.0.0.60	CIP	114	Success: 'MOTOR:1' - Service (6
11	0.074210260	10.0.0.60	10.0.0.10	TCP	66	39178 → 44818 [ACK] Seq=97 Ack=
12	0.083629957	10.0.0.60	10.0.0.10	TCP	66	39178 → 44818 [FIN, ACK] Seq=97
13	0.084479909	10.0.0.10	10.0.0.60	TCP	66	44818 → 39178 [FIN, ACK] Seq=77
14	0.084508428	10.0.0.60	10.0.0.10	TCP	66	39178 → 44818 [ACK] Seq=98 Ack=
15	5.141282130	00:00:00_00:00:01	00:00:00_00:00:0d	ARP	42	Who has 10.0.0.60? Tell 10.0.0.
16	5.160933967	00:00:00_00:00:0d	00:00:00_00:00:01	ARP	42	Who has 10.0.0.10? Tell 10.0.0.
17	5.160958346	00:00:00_00:00:01	00:00:00_00:00:0d	ARP	42	10.0.0.10 is at 00:00:00:00:00:
18	5.168433345	00:00:00_00:00:0d	00:00:00_00:00:01	ARP	42	10.0.0.60 is at 00:00:00:00:00:
19	20.063618579	10.0.0.60	10.0.0.10	TCP	74	39196 → 44818 [SYN] Seq=0 Win=4
20	20.063652673	10.0.0.10	10.0.0.60	TCP	74	44818 → 39196 [SYN, ACK] Seq=0
21	20.068171881	10.0.0.60	10.0.0.10	TCP	66	39196 → 44818 [ACK] Seq=1 Ack=1
22	20.113861381	10.0.0.60	10.0.0.10	ENIP	94	Register Session (Req), Session
23	20.113889582	10.0.0.10	10.0.0.60	TCP	66	44818 → 39196 [ACK] Seq=1 Ack=2
24	20.117951711	10.0.0.10	10.0.0.60	ENIP	94	Register Session (Rsp), Session
25	20.117963680	10.0.0.60	10.0.0.10	TCP	66	39196 → 44818 [ACK] Seq=29 Ack=
26	20.122992986	10.0.0.60	10.0.0.10	CIP CM	134	Unconnected Send: 'MOTOR:1' - S
27	20.123003292	10.0.0.10	10.0.0.60	TCP	66	44818 → 39196 [ACK] Seq=29 Ack=
28	20.139820115	10.0.0.10	10.0.0.60	CIP	114	Success: 'MOTOR:1' - Service (6
29	20.139855038	10.0.0.60	10.0.0.10	TCP	66	39196 → 44818 [ACK] Seq=97 Ack=
30	20.150774693	10.0.0.60	10.0.0.10	TCP	66	39196 → 44818 [FIN, ACK] Seq=97
31	20.151491868	10.0.0.10	10.0.0.60	TCP	66	44818 → 39196 [FIN, ACK] Seq=77
32	20.151521284	10.0.0.60	10.0.0.10	TCP	66	39196 → 44818 [ACK] Seq=98 Ack=

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

Ethernet II, Src: 00:00:00_00:00:0d (00:00:00:00:00:0d), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)

Internet Protocol Version 4, Src: 10.0.0.60, Dst: 10.0.0.10

Transmission Control Protocol, Src Port: 39178, Dst Port: 44818, Seq: 0, Len: 0

0000 00 00 00 00 00 01 00 00 00 00 0d 08 00 45 00E.

0010 00 3c 09 9c 40 00 40 06 1c db 0a 00 00 3c 0a 00 ..<..@..<..

0020 00 0a 99 0a af 12 b1 19 33 c6 00 00 00 a0 023.....

0030 a5 64 99 ae 00 00 02 04 05 b4 04 02 00 0a 01 cf ..d.....

0040 c5 d9 00 00 00 00 01 03 03 093.....

s1-eth1: <live capture in progress> Packets: 32 · Displayed: 32 (100.0%) Profile: Default

Was ist STIX?

- STIX (auch Structured Threat Information Expression) eine standardisierte Sprache, um Bedrohungen im Cyber-Umfeld zu beschreiben [Jordan2020, 14]
- Informationen lassen sich einfach teilen, speichern, analysieren oder automatisiert verarbeiten
- STIX 2.x verwendet JSON als Serialisierungssprache
- Graph-basiertes Modell: Domain Objekte als Knoten und Beziehungs-Objekte als Kanten
- Besteht aus: Domain Objects, Relationship Objects, Cyber-observable Objects, Meta Objects, Bundle Objects

- SDO: z.B. „Threat Actor“



Threat Actor

- Beispiel Code für eine SRO „Relationship“:

```
{  
    "type": "relationship",  
    "spec_version": "2.1",  
    "id": "relationship--13",  
    "created": "2020-07-09T10:40:00.000Z",  
    "modified": "2020-07-09T10:40:00.000Z",  
    "relationship_type": "uses",  
    "source_ref": "threat-actor--1",  
    "target_ref": "infrastructure--1"  
},
```

Mapping der Daten

- Um den Traffic genauer unter die Lupe zu nehmen ist ein Parser hilfreich
- Wichtig: welche Daten sind für mich relevant (Bspw. Verbindungsdaten)
- Beispiel: Codeausschnitt Python Parser

```
for d in data:
    protocol = d['_source']['layers']['frame']['frame.protocols']

    if TCP_PROTO in protocol:
        proto = 'TCP'
        time = forttime(d['_source']['layers']['frame']['frame.time'])
        eth_dst = d['_source']['layers']['eth']['eth.dst']
        eth_src = d['_source']['layers']['eth']['eth.src']
        ip_src = d['_source']['layers']['ip']['ip.src']
        ip_dst = d['_source']['layers']['ip']['ip.dst']
        tcp_src_port = d['_source']['layers']['tcp']['tcp.srcport']
        tcp_dst_port = d['_source']['layers']['tcp']['tcp.dstport']

        # Daten in ein Liste schreiben
        liste.append([time, proto, eth_dst, eth_src, ip_src, ip_dst, tcp_src_port, tcp_dst_port])

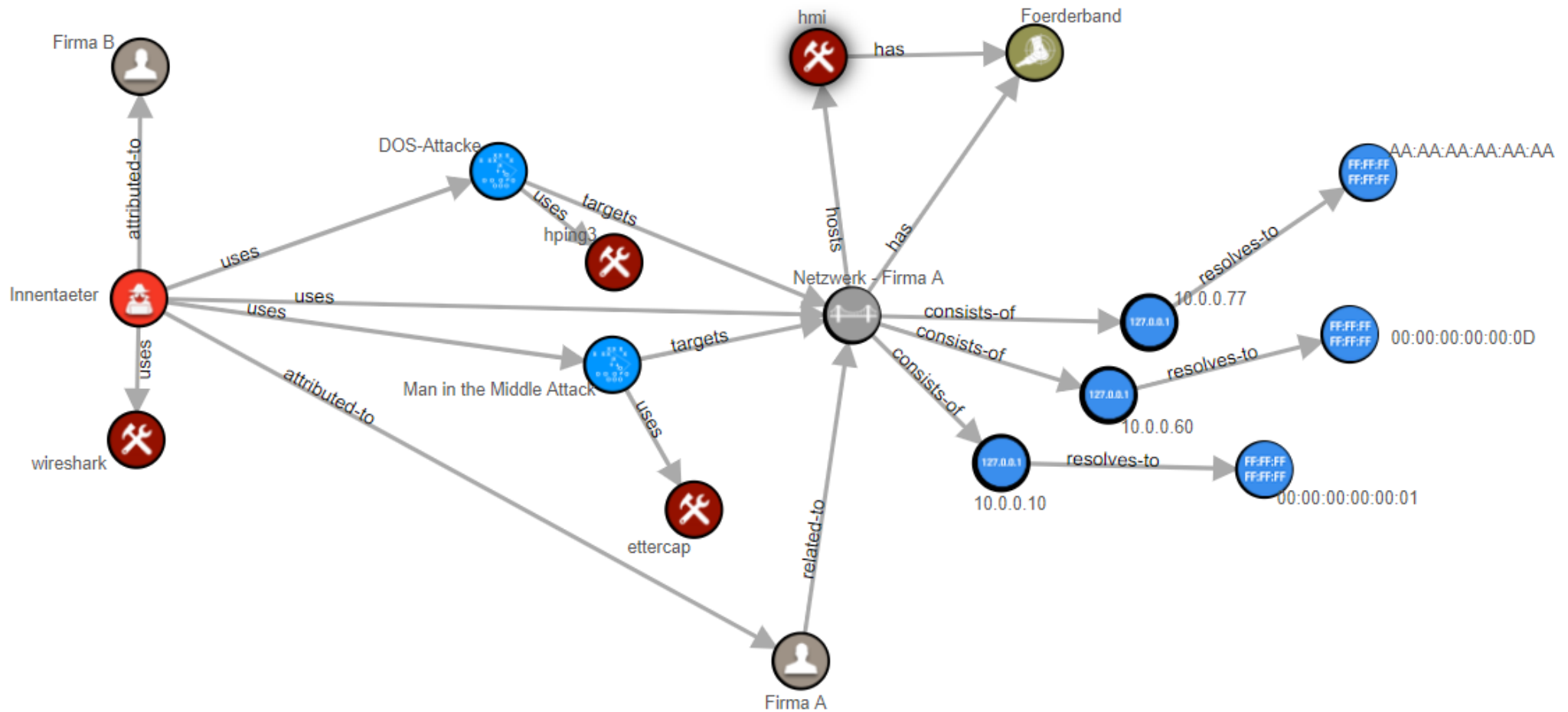
    elif ARP_PROTO in protocol:
        proto = 'ARP'
        time = forttime(d['_source']['layers']['frame']['frame.time'])
        eth_dst = d['_source']['layers']['eth']['eth.dst']
        eth_src = d['_source']['layers']['eth']['eth.src']

        # Daten in ein Liste schreiben
        liste.append([time, proto, eth_dst, eth_src])

    else:
        continue
```

```
{
  "type": "bundle",
  "id": "bundle--1",
  "objects": [
    {
      "type": "threat-actor",
      "spec_version": "2.1",
      "id": "threat-actor--1",
      "created": "2020-07-08T23:39:03.893Z",
      "modified": "2020-07-08T23:39:03.893Z",
      "name": "Innentaeter",
      "description": "Ein Innentaeter versucht den aktiven Prozess mitzuhoeren, Informationen zu gewinnen und Daten zu manipulieren.",
      "threat_actor_types": [
        "insider-disgruntled"
      ],
      "roles": [
        "agent"
      ],
      "goals": [
        "Informationen ueber den Netzwerkverkehr im Unternehmen gewinnen",
        "Daten manipulieren"
      ],
      "sophistication": "expert",
      "resource_level": "organization",
      "primary_motivation": "personal-gain",
      "secondary_motivations": [
        "dominance"
      ]
    },
    {
      "type": "infrastructure",
      "spec_version": "2.1",
      "id": "infrastructure--1",
      "created": "2020-07-08T23:39:03.893Z",
      "modified": "2020-07-08T23:39:03.893Z",
      "name": "Netzwerk - Firma A",
      "description": "Das Netzwerk der Firma A, welches aus mehreren Teilnehmern besteht."
    },
    {
      "type": "identity",
      "spec_version": "2.1",
      "id": "identity--1",
      "created": "2020-07-08T23:39:03.893Z",
      "modified": "2020-07-08T23:39:03.893Z",
      "name": "Firma B",
      "description": "Firma B versucht Innentaeter anzuwerben und dadurch der Konkurrenz zu schaden.",
      "identity_class": "organization"
    }
  ],
}
```

Visualisierung (bspw. mit STIX Visualizer)



Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

4 Aufarbeitung der Daten

5 Zusammenfassung und Fazit

6 Quellenverzeichnis

Zusammenfassung und Fazit

■ Zusammenfassung:

- Im Rahmen des Seminars wurde ein DT mittels miniCPS-Framework erstellt.
- Der DT simulierte ein einfaches Förderbandsystem, dass über einen PLC mit dem Netzwerk verbunden wurde und über eine HMI bedient werden konnte
- Auf den DT wurde ein DOS-Angriff simuliert und der Netzwerkverkehr mit Wireshark aufgezeichnet
- Mit Stix 2.x wurden die Daten des Netzwerkmitschnitts strukturiert aufgearbeitet

■ Fazit:

- Thema bietet noch viel mehr Spielraum als gezeigt – ABER: limitiert durch Zeit nur Konzentration auf das Wichtigste
- Durch einen DT kann die Security einer ICS auch während des Betriebs getestet werden
- Durch die Verwendung von STIX können Security-Experten strukturiert miteinander zusammenarbeiten
- Sehr intensiv in neue Themen eingearbeitet
- Vielen Dank an die Unterstützung durch die Betreuer Marietheres Dietz und Daniel Schlette

Gliederung der Präsentation

1 Aufbau und Ziel des Praxisseminars

2 Entwicklung des „Digital Twins“

3 Angriffssimulation

4 Aufarbeitung der Daten

5 Zusammenfassung und Fazit

6 Quellenverzeichnis

[Antonioli2015]

Antonioli, D. & Tippenhauer, N. O.: MiniCPS: A toolkit for security research on CPS Networks

[Bodewein2020]

Bodewein, L.: Australien meldet massiven Cyberangriff. <https://www.tagesschau.de/australien-cyberangriffe-101.html>, Abruf am 2020-07-13.

[Eckhart2018]

Eckhart, M. & Ekelhart, A.: Towards Security-Aware Virtual Environments for Digital Twins. Proceedings of the 4th ACM Workshop on Cyber-Physical System Security - CPSS '18, ACM Press, 2018

[Holland2020]

Holland, M.: Während Coronavirus-Pandemie: Cyberangriff legt tschechisches Krankenhaus lahm. <https://www.heise.de/security/meldung/Waehrend-Coronavirus-Pandemie-Cyberangriff-legt-tschechisches-Krankenhaus-lahm-4683370.html>, Abruf am 2020-07-13.

[Jordan2020]

Jordan, B.; Piazza, R. & Darley, T.: StixTM version 2.1. Committee Specification 01

[Schmitz2020]

Schmitz, P.: COVID-19 Cyber Threat Assessment. Cyberangriffe folgen der Entwicklung von COVID-19. <https://www.security-insider.de/cyberangriffe-folgen-der-entwicklung-von-covid-19-a-926037/>, Abruf am 2020-07-13.

Fragen / Diskussion