# NMAP-Erkennung von SPS

**Philipp Wins**

# Gliederung

- Einleitung
- Nmap
- Analyse
- Entwicklung eigener Skripte
- Entwicklung GUI
- Zusammenfassung

# Einleitung

- Zunehmende Vernetzung (Industrie 4.0)
- Netzwerke werden komplexer, Verwaltungsaufwand steigt
- Immer häufiger Angriffe auf IT-Systeme
- Administrator muss sein Netzwerk kennen

# Nmap

- Kurzform für „Network Mapper“
- Ursprünglich einfacher Portscanner (1997)
- Stetige Weiterentwicklung: Version-Detection, OS-Fingerprinting, Schwachstellenerkennung…
- Kann große Netzwerke in kurzer Zeit scannen oder gezielt einzelne Hosts analysieren

# Nmap - NSE

- NSE = Nmap Scripting Enginge
- Bietet die Möglichkeit Nmap durch Lua-Skripte zu erweitern
- Spezielle Anwendungsfälle können so abgedeckt werden

# Analyse

- Untersucht wurden drei verschiedene Hersteller:
  - Schneider Electric  HMI (STU655) & SPS (M238)
  - Siemens SIMATIC S7-1516
  - Bosch Rexroth IndraControl L25
- Portscan mit Nmap
- Analyse des Datenverkehrs mit Wireshark

# Analyse - Portscan

- Beispiel: Schneider Electric HMI
- TCP-SYN-Scan:

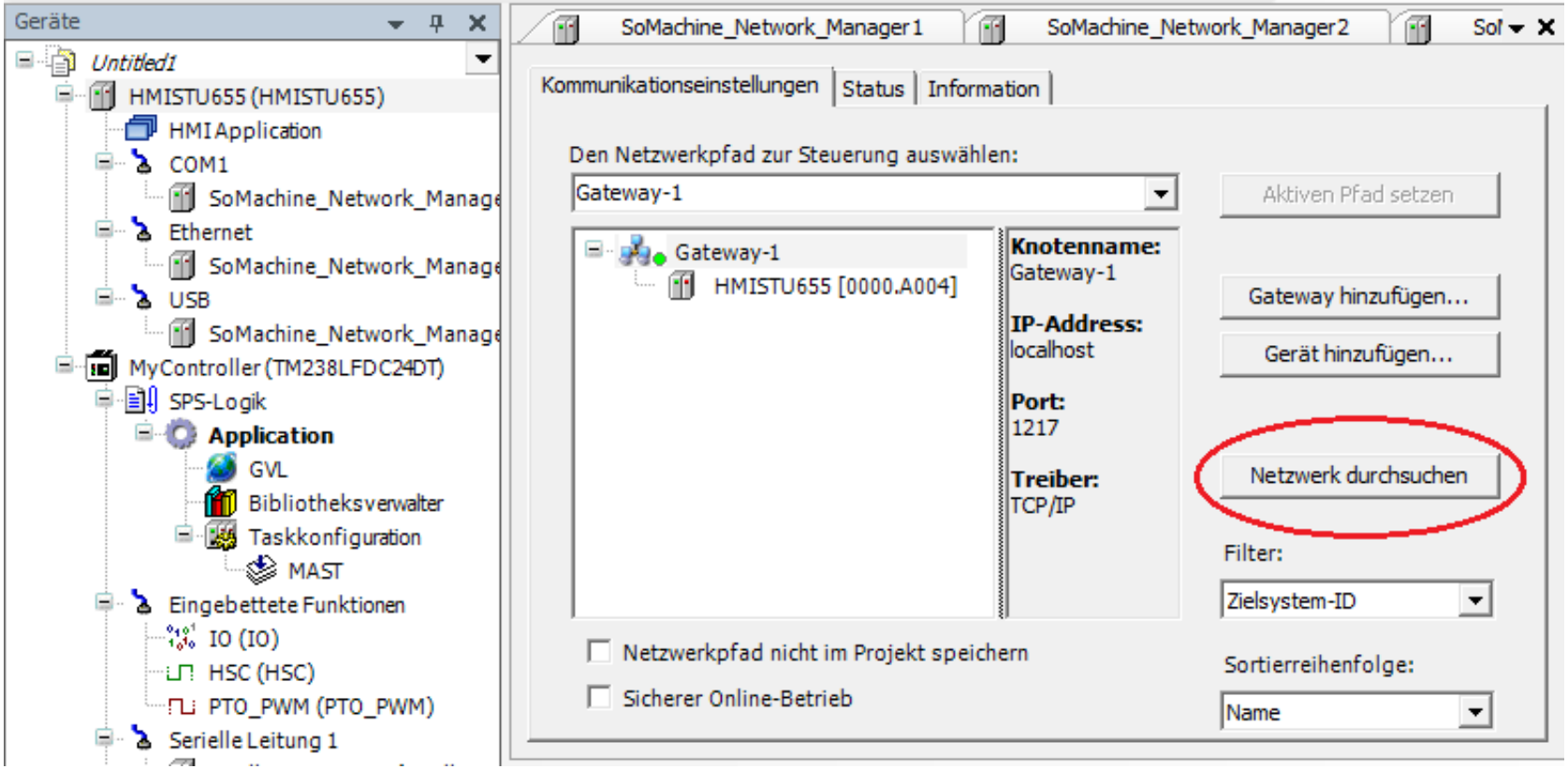
```
Host is up (0.0018s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE
1105/tcp open  ftranhc
1217/tcp open  hpss-ndapi
6001/tcp open  X11:1
MAC Address: 00:01:23:17:F7:9B (Digital Electronics)
```

- UDP-Scan:

```
Host is up (0.0014s latency).
Not shown: 65534 closed ports
PORT     STATE         SERVICE
1740/udp open|filtered encore
MAC Address: 00:01:23:17:F7:9B (Digital Electronics)
```

# Analyse – SoMachine Software

- SoMachine = Konfigurationssoftware des Geräts
- Ermöglicht das Auffinden und Erkennen:



# Analyse – Datenverkehr

- Wireshark Capture:

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1444 | 528.153141 | 192.168.0.109 | 192.168.0.255 | UDP | 62 | 1740 → 1740 Len=20 |
| 1445 | 528.153229 | 192.168.0.109 | 192.168.0.255 | UDP | 62 | 1740 → 1741 Len=20 |
| 1446 | 528.153255 | 192.168.0.109 | 192.168.0.255 | UDP | 62 | 1740 → 1742 Len=20 |
| 1447 | 528.153276 | 192.168.0.109 | 192.168.0.255 | UDP | 62 | 1740 → 1743 Len=20 |
| 1448 | 528.162496 | 192.168.0.4 | 192.168.0.109 | UDP | 170 | 1740 → 1740 Len=128 |
| 1449 | 528.244779 | 192.168.0.4 | 192.168.0.109 | UDP | 188 | 1740 → 1740 Len=146 |

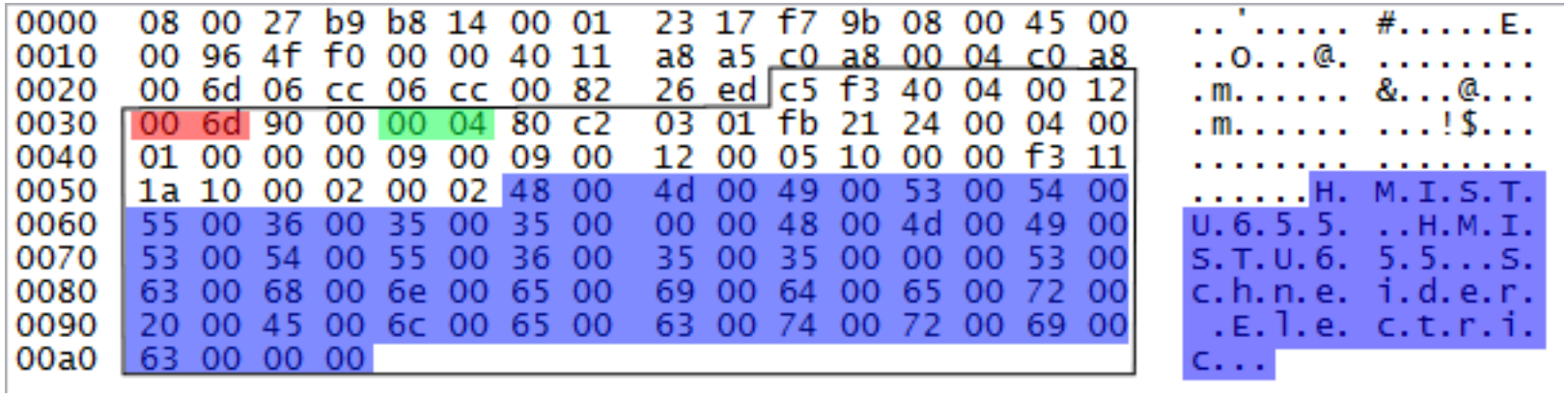
- Gesendeter Broadcast enthält Teile der IP-Adresse:

```
0000  ff ff ff ff ff ff 08 00  27 b9 b8 14 08 00 45 00   ........ '.....E.
0010  00 30 2b e7 00 00 80 11  8c 19 c0 a8 00 6d c0 a8   .0+..... .....m..
0020  00 ff 06 cc 06 cc 00 1c  82 ea c5 f4 40 03 00 20   ........ ....@..
0030  a2 d4 00 6d 90 00 02 c2  03 01 fa 21 24 00        ...m.... ...!$.
```

- 00 6D = 0 109
- Gerät sendet Antworten an diese IP-Adresse

# Analyse – Datenverkehr

- Die Antworten enthalten Informationen über Hersteller und Modell des Geräts und weitere angeschlossene Geräte
- Beispiels: Schneider Electric HMI (STU655)



# Analyse – Datenverkehr

- Beispiels: Siemens SIMATIC S7-1516

```
0000  08 00 27 00 bf 63 28 63  36 92 ba 2a 08 00 45 00   ..'..c(c 6..*..E.
0010  00 ef 00 3b 00 00 1e 06  1a 5c c0 a8 00 13 c0 a8   ...;.... .\......
0020  00 0f 00 66 c0 1c 00 06  05 91 d8 5b 14 0b 50 18   ...f.... ...[..P.
0030  10 00 1c e0 00 00 03 00  00 c7 02 f0 80 72 01 00   ........ .....r..
0040  b8 32 00 00 04 ca 00 00  00 01 36 00 02 87 05 87   .2...... ..6.....
0050  35 a1 00 00 01 20 82 1f  01 00 a3 81 69 00 15 02   5.... .. ....i...
0060  30 30 a3 82 2b 00 04 82  80 80 80 00 a3 82 2d 00   00..+... ......-.
0070  15 10 4f 4d 53 50 2e 52  45 4c 2e 37 30 37 30 2e   ..OMSP.R EL.7070.
0080  31 37 a3 82 2f 10 02 14  76 83 3d 64 ed 87 e8 48   17../... v.=d...H
0090  d9 0e e4 8b 43 d6 9a 42  a3 4d 32 96 a3 82 32 00   ....C..B .M2...2.
00a0  17 00 00 01 3a 82 3b 00  04 83 40 82 3c 00 04 83   ....:.;. ..@.<...
00b0  40 82 3d 00 04 84 81 82  00 82 3e 00 04 84 81 81   @.=..... ..>.....
00c0  40 82 3f 00 15 1a 31 3b  36 45 53 37 20 35 31 36   @.?...1; 6ES7 516
00d0  2d 33 41 4e 30 31 2d 30  41 42 30 3b 56 31 2e 38   -3AN01-0 AB0;V1.8
00e0  82 40 00 15 08 32 3b 32  38 37 34 39 31 82 41 00   .@...2;2 87491.A.
00f0  03 00 03 00 a2 00 00 00  00 72 01 00 00           ........ .r...
```

# Entwicklung eigener Skripte

- Mit der NSE lässt sich die Geräteanalyse automatisieren
- 3 NSE Skripte entwickelt um das Modell der Geräte bestimmen zu können
- Jedes Skript besteht aus dem Skript selbst und einer Library

# Entwicklung eigener Skripte

Funktionen:

- Protokollanalyse
- Scan typischerweise verwendeter Ports
- Ausgabe der Ergebnisse

# Entwicklung eigener Skripte



Protokollanalyse:

- Library wird aufgerufen
  - Library sendet  Datagramm an Ziel und analysiert die Antworten
- Beispiel: Schneider Electric
- Antworten enthalten Modell- und Herstellerbezeichnung in ASCII kodiert
- ASCII-Zeichen werden in einem String aufgereiht
- String wird nach bekannten Bezeichnungen durchsucht

# Entwicklung eigener Skripte



## Protokollanalyse:

- Datagramm wird gesendet

```
0000   ff ff ff ff ff ff 08 00   27 b9 b8 14 08 00 45 00   ........ '.....E.
0010   00 30 2b e7 00 00 80 11   8c 19 c0 a8 00 6d c0 a8   .0+..... .....m..
0020   00 ff 06 cc 06 cc 00 1c   82 ea c5 f4 40 03 00 20   ........ ....@..
0030   a2 d4 00 6d 90 00 02 c2   03 01 fa 21 24 00        ...m.... ...!$.
```

- Antwort enthält Details

```
0000   08 00 27 b9 b8 14 00 01   23 17 f7 9b 08 00 45 00   ..'..... #.....E.
0010   00 96 4f f0 00 00 40 11   a8 a5 c0 a8 00 04 c0 a8   ..O...@. ........
0020   00 6d 06 cc 06 cc 00 82   26 ed c5 f3 40 04 00 12   .m...... &...@...
0030   00 6d 90 00 00 04 80 c2   03 01 fb 21 24 00 04 00   .m...... ...!$...
0040   01 00 00 00 09 00 09 00   12 00 05 10 00 00 f3 11   ........ ........
0050   1a 10 00 02 00 02 48 00   4d 00 49 00 53 00 54 00   ......H. M.I.S.T.
0060   55 00 36 00 35 00 35 00   00 00 48 00 4d 00 49 00   U.6.5.5. ..H.M.I.
0070   53 00 54 00 55 00 36 00   35 00 35 00 00 00 53 00   S.T.U.6. 5.5...S.
0080   63 00 68 00 6e 00 65 00   69 00 64 00 65 00 72 00   c.h.n.e. i.d.e.r.
0090   20 00 45 00 6c 00 65 00   63 00 74 00 72 00 69 00    .E.l.e. c.t.r.i.
00a0   63 00 00 00                                        c...
```

eHMISTU655HMI
STU655Schneider
Electric

eHMISTU655HMI
STU655Schneider
Electric

```
8   manufacturers = {
9     ["SchneiderElectric"] = "Schneider Electric"
10  }
11
12  models = {
13    ["HMISTU655"] = "Magelis HMI STU655",
14    ["TM238LFDC24DT"] = "Modicon M238 SPS TM238LFDC24DT"
15  }
16
```

# Entwicklung eigener Skripte

Portscan:

- Abfrage von Ports, die typischerweise von Geräten des Herstellers benutzt werden

```
local services = {
  ["FTRANHC"] = {
    ["port"] = 1105,
    ["protocol"] = "tcp"
  },
  ["HPSS-NDAPI"] = {
    ["port"] = 1217,
    ["protocol"] = "tcp"
  },
  ["X11:1"] = {
    ["port"] = 6001,
    ["protocol"] = "tcp"
  },
  ["SoMachine"] = {
    ["port"] = 1740,
    ["protocol"] = "udp"
  }
}
```

# Entwicklung eigener Skripte

Ausgabe der Ergebnisse:

- Konsolenausgabe
- XML-Ausgabe
  - Eigenes Skript zur XML-Ausgabe
  - Kein Befehlsparameter benötigt
  - Nimmt Ergebnisse in bestimmten Format entgegen
  - Zur Weiterverarbeitung der Ergebnisse (z.B. UI)

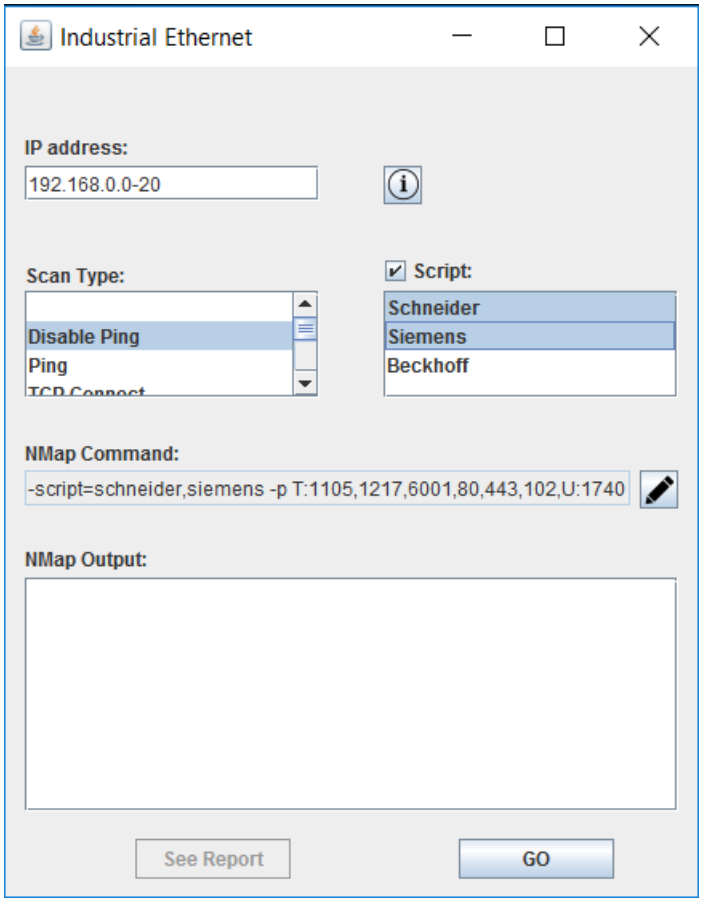
# Entwicklung GUI

- GUI = Graphical User Interface
- In Java geschrieben
- Basiert auf Nmap und den angefertigten Skripten
- Erleichtert die Bedienung der Skripte
- Strukturierte Anzeige der Skriptergebnisse
- Erweiterbar

# Entwicklung GUI

Funktionen:

- Nimmt IP-Adresse entgegen
- Bietet Auswahl für Scan-Typ und Skripte
- Stellt Nmap-Befehl zusammen
- Nmap Konsolenausgabe
- Report für Skriptergebnisse



# Zusammenfassung

- Was habe ich gemacht?
- Fazit
  - Erkennung möglich
  - Viele Protokolle, wenige Programme