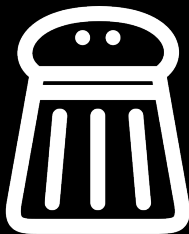


IoT Honeypot, new types of attacks



Sebastien Tricaud

July 2, 2018

IoT, Internet of Things

- Industrial Control System (ICS)
 - Supervisory Control and Data Acquisition (SCADA)
 - Distributed Control Systems (DCS)
 - Programmable Logic Controllers (PLC)
- It is an old industry
 - Ever heard of Industrial Real-Time Fortran? https://en.wikipedia.org/wiki/Industrial_Real-Time_Fortran
 - Modbus was published in 1979

Common IoT problems

- Asset Inventory and Discovery
- Patch Management, Firmware update, Offline Vessel
- Remote Access to third party vendors

Top 5 Areas of Weaknesses by ICS US CERT

- 1 Boundary Protection
- 2 Least Functionality
- 3 Identification and Authentication (Org Users)
- 4 Physical Access Control
- 5 Audit, Review, Analysis and Reporting

NIST 800-82 Methodology

- Step 1: Categorize Information System
- Step 2: Select Security Controls
- Step 3: Implement Security Controls
- Step 4: Assess Security Controls
- Step 5: Authorize Information System
- Step 6: Monitor Security Controls

Example, Gas Station Honeygot

- The GasPot Experiment: Unexamined Perils in Using Gas-Tank-Monitoring Systems
- By Kyle Wilhoit and Stephen Hilt
- Talk given at BlackHat in 2015
- https://documents.trendmicro.com/assets/wp/wp_the_gaspot_experiment.pdf

Conpot, the ICS Honeypot

- Written by Lukas Rist in 2013
- Works with a templating system
- Creates `/var/log/conpot/conpot.log`
- Simulates Siemens S7-200 PLC, Guardian AST tank monitors etc.

Conpot Template for Guardian AST devices

```
<guardian_ast enabled="True" host="0.0.0.0" port="10001">  
  <device_info>  
    <vendor_name>Guardian</vendor_name>  
    <product_code>Guardian AST</product_code>  
  </device_info>  
</guardian_ast>
```


Conpot Template for Guardian AST devices

```
<core>
  [..]
  <databus>
<!-- Core value that can be retrieved from the databus by key -->
<key_value_mappings>
  <key name="product1">
<value type="value">"SUPER"</value>
  </key>
  <key name="product2">
<value type="value">"UNLEAD"</value>
  </key>
  <key name="product3">
<value type="value">"DIESEL"</value>
  </key>
  <key name="product4">
<value type="value">"PREMIUM"</value>
  </key>
  <key name="station_name">
<value type="value">"STATOIL STATION"</value>
  </key>
  <key name="vol1">
<value type="value">random.randint(1000, 9050)</value>
  </key>
  <key name="vol2">
<value type="value">random.randint(1000, 9050)</value>
  </key>
  </databus>
</core>
```

Nmap already provides a Guardian AST scanner!

```
$ nmap --script atg-info -p 10001 <host>
10001/tcp open  Guardian AST reset
| atg-info:
| I20100
| SEP 19, 2015  5:33 PM
|
| Fuel Company
| 12 Fake St
| Anytown, USA 12345
|
|
| IN-TANK INVENTORY
|
| TANK PRODUCT          VOLUME TC VOLUME    ULLAGE  HEIGHT  WATER  TEMP
|  1 UNLEADED           5135   0     6647    42.71   0.00  72.01
|  2 UNLEADED           5135   0     6647    42.70   0.00  71.55
|  3 PREMIUM UNLEADED   5135   0     5350    19.27   0.00  72.52
|_
```

Nmap script

```
action = function(host, port)
  local command = "I20100"
  local arguments = stdnse.get_script_args('command')
  if (arguments ~= "I20100" and arguments ~= nil) then
command = arguments
  end
  local sock = nmap.new_socket()
  sock:set_timeout(1000)
  local tank_command = "\x01" .. command .. "\n"
  [..]
local sendstatus, senderr = sock:send(tank_command)
local rcvstatus, response = sock:receive_bytes(1024)
if (response == "TIMEOUT" or response == "EOF") then
  sock:close()
  return "TIMEOUT: No response from query"
end
if(string.byte(response,1) == 0x01 or string.byte(response,1) == 0x0a) then
  local inventory_output = string.sub(response,2,-2)
  set_nmap(host, port)
  sock:close()
  return inventory_output
end
end
```

Metasploit Module

- Goes further and implemented most of Guardian AST Commands
- One can set a tank name...

Metasploit Module

- Goes further and implemented most of Guardian AST Commands
- One can set a tank name...

```
| IN-TANK INVENTORY
|
| TANK PRODUCT          VOLUME TC VOLUME  ULLAGE  HEIGHT  WATER  TEMP
|  1  SPLUNK IS         5135   0         6647   42.71   0.00  72.01
|  2  HIRING            5135   0         6647   42.70   0.00  71.55
|  3  SECURITY PEEPS    5135   0         5350   19.27   0.00  72.52
|_
```

Conpot Results on a Guardian AST device


- 3 months (April-June 2018)
- Total of 5 unique IP connecting on a 2 months period
 - Seychelles (2 sessions)
 - Russia (1 session)
 - Latvia (1 session)
 - USA (3 sessions)
 - Hong-Kong (4 sessions)

`/var/log/conpot/conpot.log`

```
2018-05-26 05:21:6,2 New guardian_ast session from host (c21d562e-7d68-4492-95c0-4e4d477a6a7c)
```

Siemens Simatic S7-200

- Emulated by Conpot (default template)
- Reliable, fast and modular controller that can be used in lots of area
- Programmable
- Enables automation of various devices



Siemens Simatic S7-200 CPU 214 PLC 6ES7214-1AC00-0XB0

\$35.76
Buy it Now
78 watching | 🔥 68 sold

[View Details](#)

Condition: Used

Time left: 28d 15h 33m

Item location: Michigan

Sold by: [cncsalvage11 \(3570★\)](#)

Communication Interfaces

- Profinet / Industrial Ethernet
- Industrial Wireless LAN
- Profibus
- AS-Interface
- WAN
- Multi-Point Interface (MPI)
- Point-to-Point Interface (PPI)
- KNX/EIB (KONNEX)

Communication Services

- FTP
- Email
- SNMP
- OPC
- Profinet IO
- Profinet CBA
- S7
- PG/OP

Exploitation

```
2018-03-28 05:06:52,792 New IPMI traffic from ('184.105.247.214', 3863)
2018-03-28 05:06:52,793 New IPMI session initialized for client (('184.105.247.214', 3863))
2018-03-28 05:06:52,793 Connection established with ('184.105.247.214', 3863)
2018-03-28 05:06:52,793 IPMI response sent to ('184.105.247.214', 3863)
2018-03-28 05:48:40,823 New snmp session from 209.126.136.2 (7537b1b0-b091-4228-b521-6722fe22c9a3)
2018-03-28 05:48:40,823 SNMPv2 Get request from ('209.126.136.2', 51493): 1.3.6.1.2.1.1.1.0
2018-03-28 05:48:40,823 SNMPv2 Get response to ('209.126.136.2', 51493): 1.3.6.1.2.1.1.1.0 Siemens, SIMATIC
2018-03-28 05:51:47,563 handle server TID [16431/3783674803216] starting on ('196.52.43.57', 6666)
2018-03-28 05:52:18,463 ( ENIP_6666.( header.(empty)) recv: 0: ''
2018-03-28 05:52:18,464 Transaction parsed after 30.899s
2018-03-28 05:52:18,464 Class 1/0x0001, Instance 1, Attribute None ==> None
2018-03-28 05:52:18,464 Identity, Class ID 0x0001, Instance ID 1 created
2018-03-28 05:52:18,464 Class 1/0x0001, Instance 1, Attribute None ==> None
2018-03-28 05:52:18,465 Class 1/0x0001, Instance 0, Attribute None ==> None
2018-03-28 05:52:18,465 meta-Identity, Class ID 0x0001, Instance ID 0 created
2018-03-28 22:17:09,123 Modbus traffic from 66.240.219.146: {'function_code': None, 'slave_id': 255, 'req
2018-03-28 22:17:09,123 Modbus response sent to 66.240.219.146
2018-03-28 22:17:09,400 New Modbus connection from 66.240.219.146:40756. (d389cf90-2505-41af-bc09-253b471
2018-03-28 22:17:09,401 Modbus traffic from 66.240.219.146: {'function_code': None, 'slave_id': 255, 'req
2018-03-28 22:17:09,401 Modbus response sent to 66.240.219.146
```

Results

- Access to the emulated PLC device was really quick: First S7 frame after 3 hours
- If we remove HTTP and SNMP to get people really targeting IoT devices
 - S7: 10 established connections
 - Modbus: 518 established connections
 - IPMI: 47



Questions?

Thank you!

stricaud@splunk.com, [@tricaud](https://twitter.com/tricaud)