Testbed Data Collection Capabilities and Implemented Attacks

This writeup serves to give a brief overview/summary about possible data collection and implemented attacks.

Data Collection Capabilities

Currently the testbed has two major ways of collecting data: Wireshark remote capture (sshdump) and the data historian (TICK stack).

Wireshark

We can capture all traffic on each of the testbed hosts via sshdump/remote capture [1].

TICK Stack

The data historian in the testbed is implemented with the TICK stack [2]. Which includes the data collection agent Telegraf. Telegraf has a lot of plugins [4]. In the default configuration we only use the Modbus plugin [5]. However there are a lot of other plugins that could be interesting:

- **network monitoring:** e.g. netstat [6]
- **resource monitoring**
- **output plugins** that determine where it writes collected data to. The testbed currently only uses InfluxDB as output, however other targets might be interesting for your purposes.

The collected data can easily collected from the database or viewed in the dashboard.

Implemented Attacks

For the thesis I implemented a number of attacks, three were covered in detail. Those three attacks all used ARP cache poisoning to obtain a MitM position. The ARP cache poisoning was implemented and is not just an assumption. Meaning it would be included in the network captures.

Two attacks are similar in their effect but have a different technical implementation. The attacks modify the sensor values that reach the PLC. One variant of the attack directly modifies the Modbus packets (with a python program that uses NetfilterQueue + ScaPy). The other one redirects the requests to different Modbus servers than was intended by the victims and chooses how data is passed on. This attack uses iptables to handle the redirection. The same kind of attack can be implemented on the communication of the PLC with the data historian and the HMI, in case a setup does not use Modbus TCP/IP to communicate sensor values.

The third attack is somewhat similar to Stuxnet, as we gain access to the PLC (password sniffing) and then change the PLC's program. This attack in its current variant is recognizable from the measured sensor values (data historian), however it might be possible to construct a more stealthy one.

More details on these attacks are in the thesis.

One attack that is trivial to implement would be abusing the lack of client authentication in Modbus, by writing to any of the two Modbus servers.

References

[1] sshdump: `https://www.wireshark.org/docs/man-pages/sshdump.html`

[2] TICK stack: `https://docs.influxdata.com/platform/#influxdata-1x-tick-stack`

[3] Telegraf: `https://docs.influxdata.com/telegraf/v1.18/`

[4] Telegraf plugins: `https://docs.influxdata.com/telegraf/v1.18/plugins/`

[5] Modbus plugin:

`https://github.com/influxdata/telegraf/blob/release-1.18/plugins/inputs/modbus/README.md`

[6] netstat plugin:

`https://github.com/influxdata/telegraf/blob/master/plugins/inputs/net/NETSTAT_README.md`