Serial-Killer

Security Analysis of Industrial Serial Device Servers

Dr. Florian Adamsky

October 18, 2018

University of Luxembourg, SECAN-Lab

Outline

Introduction

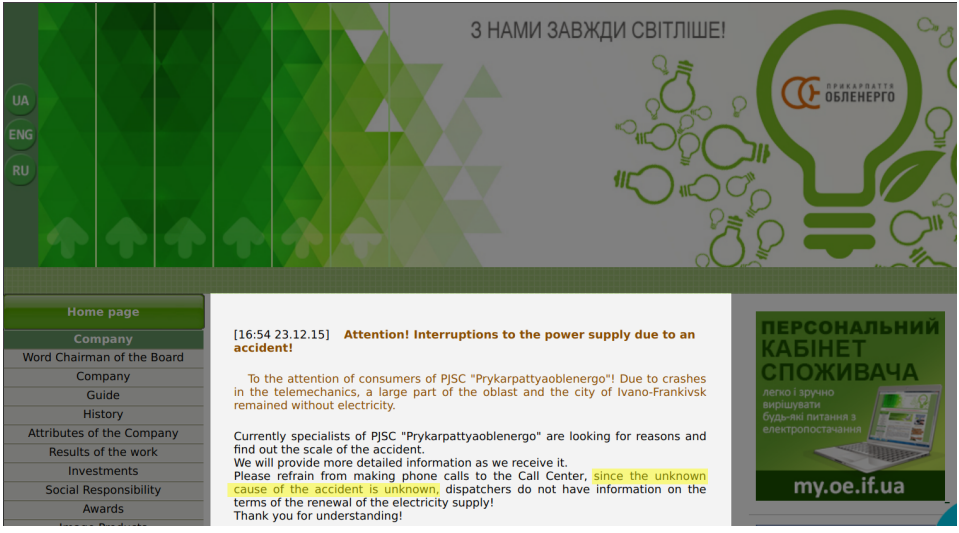
Western Ukraine

16:00 at 23.12.2015 in Western Ukraine

Prykarpattyaoblenergo

Figure 1: Source: derstandard.at

Press Release (Google Translate)



TLP: White

Analysis of the Cyber Attack on the Ukrainian Power Grid

Defense Use Case

March 18, 2016

1325 G Street NW
Suite 600
Washington, DC 20005

Summary what happend

- Outage were due to an attack against the SCADA infrastructure
- Approximately 225.000 customers were without power in three different distribution-level
- These attacks in Ukraine are the first publicly acknowledged incidents to result in power outages

Who was it?

I don’t know

Background on the Attacks

- Attacker send spear phishing emails
- Variants of the BlackEnergy 3 malware
- Manipulation of Microsoft Office documents which contain malware
- Telephone DDoS Attack

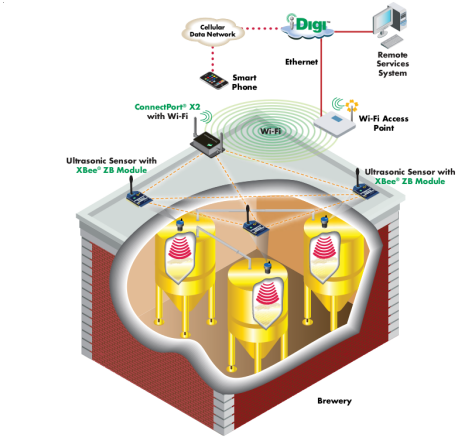
Attacks against substations

Adversaries attacked field devices at substations: *"write custom malicious firmware, and render the devices, such as serial-to-ethernet convertors, inoperable and unrecoverable."*

Research Idea

- Investigate how big the security threat is regarding these devices

Other use cases (Brewery Tank Monitoring)



Industrial Tank Monitoring

Other use cases (Medical Device Monitoring)

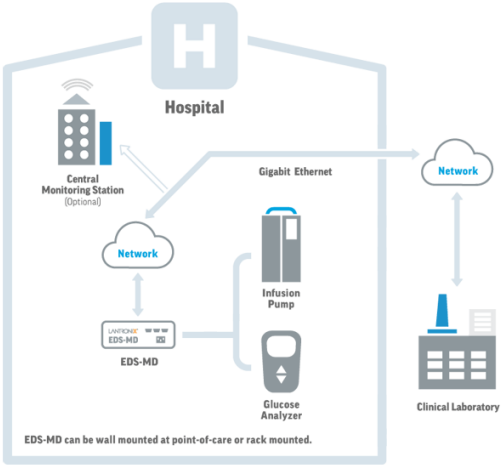


Figure 2: https://www.lantronix.com/products/eds-md/

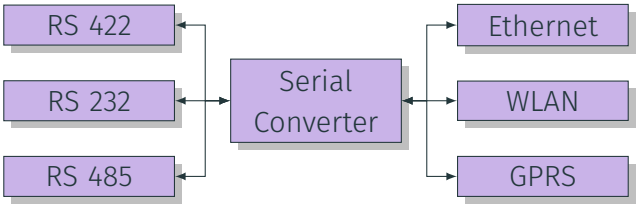


Figure 3: Overview about a generic serial server containing relevant interfaces.

Related Work

2013: HD Moore

- HD Moore gave a talk at InfoSec Southwest 2013
- Mostly focused on the company Digi
- Found scary stuff directly connected on the Internet

2013: HD Moore

- HD M
- Mostl
- Found

Dry Cleaners

National Dry Cleaner Chains

- Full access to PoS systems
- No authentication


```
                                Store Sales Summary
                                                                        Discs/     Cash/
Category  #Tiks  Total Amt    Tax1/2    #Pcs  Upchrgs  Tik Chg  Coupons    A/R Chg
-------------------------------------------------------------------------------------

LEATHER      12     456.58       .00      12      .00      .00      .00     440.18
                                36.52                                .00      52.92

WEDDING       0        .00       .00       0      .00      .00      .00        .00
                                 .00                                .00        .00

FUTURE        0        .00       .00       0      .00      .00      .00        .00
                                 .00                                .00        .00

7 [Hit ANY KEY for More  or VOID to Quit E[Str: 390 [      ] CLEANERS 390
" ["5For the Period: 01/01/12 to 06/30/12
# [#;For Times 00:00 to 24:00

                                Store Sales Summary
                                                                        Discs/     Cash/
```

RAPID7

2013: HD Moore

- HD M
- Mostl
- Found

Traffic Light Control

EDI Traffic Signal Monitors

- Based on Digi development kits, exposes ADDP
  - Default password is “dbps” as a result
  - ~40 or so identified in the Internet Census 2012 data

RAPID7

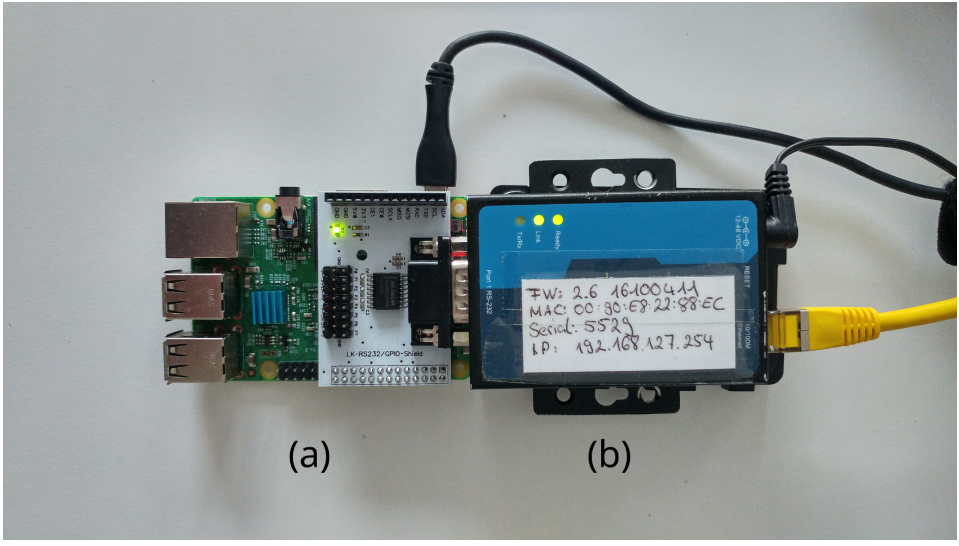
2017: 34c3

- Thomas Roth gave a talk last year on 34c3 with the title ”SCADA - Gateway to (s)hell ”
- Focused on:
  - Advantech EKI-1522
  - Moxa W2150A
  - Lantronix EDS2100

Found Vulnerabilities


80IES ARE CALLING
THEY WANT THEIR EXPLOITS BACK
HACKERMAN

Moxa NPort 5110/5130

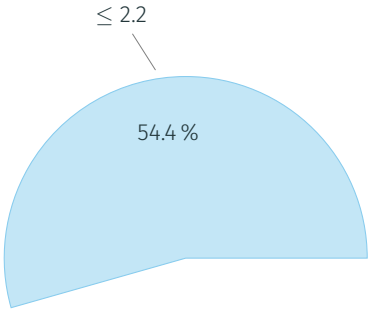


Internet-wide Scans (ZMap)

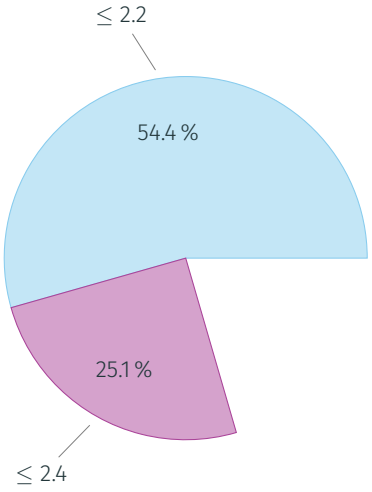
- Daily scans with ZMap on `https://censys.io`
- Analysed telnet scan from 2018-08-21
- 1877 Moxa NPort devices directly connected to the Internet
- 1150 Moxa NPort 5110/5130

Distribution of Firmware (Moxa NPort 5110/5130)

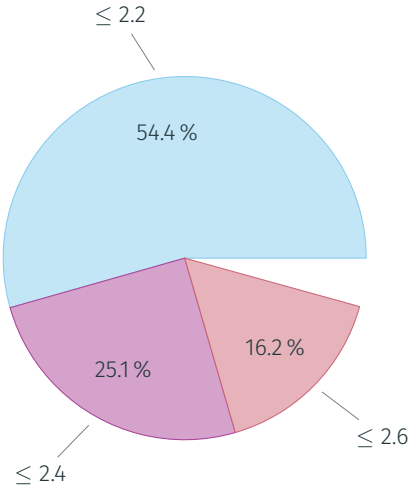
Distribution of Firmware (Moxa NPort 5110/5130)



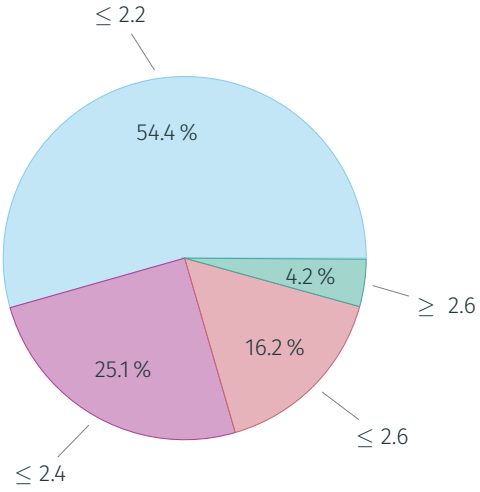
Distribution of Firmware (Moxa NPort 5110/5130)



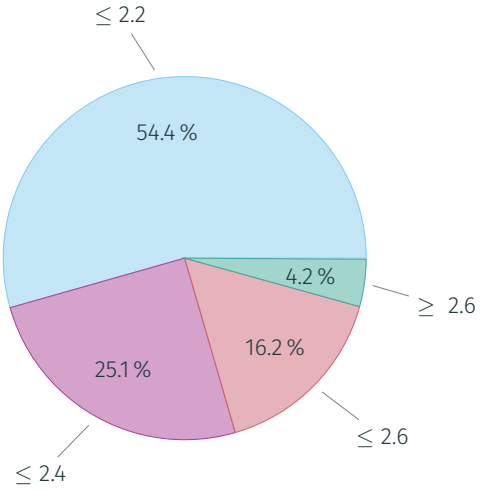
Distribution of Firmware (Moxa NPort 5110/5130)



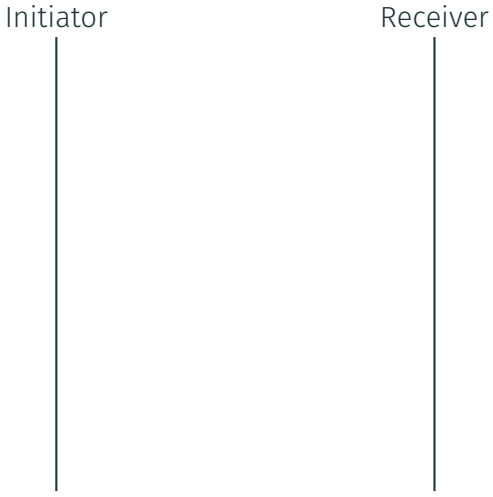
Distribution of Firmware (Moxa NPort 5110/5130)



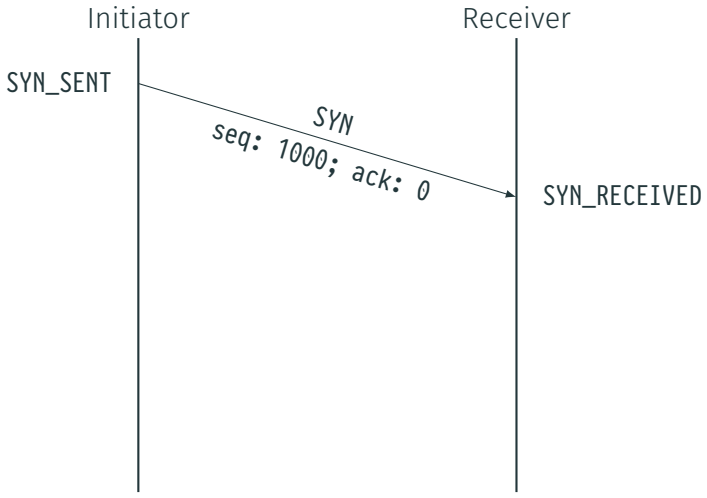
Distribution of Firmware (Moxa NPort 5110/5130)



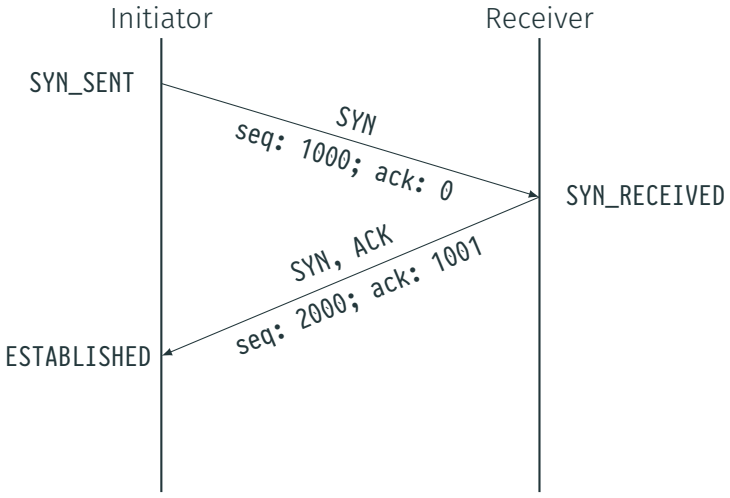
TCP Three-way Handshake—Remember?



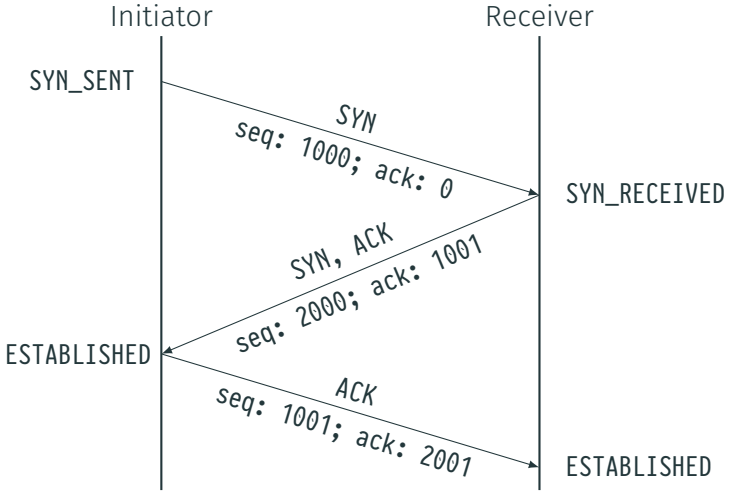
TCP Three-way Handshake—Remember?



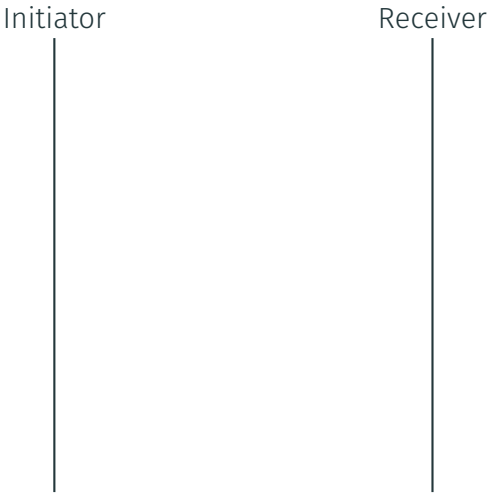
TCP Three-way Handshake—Remember?



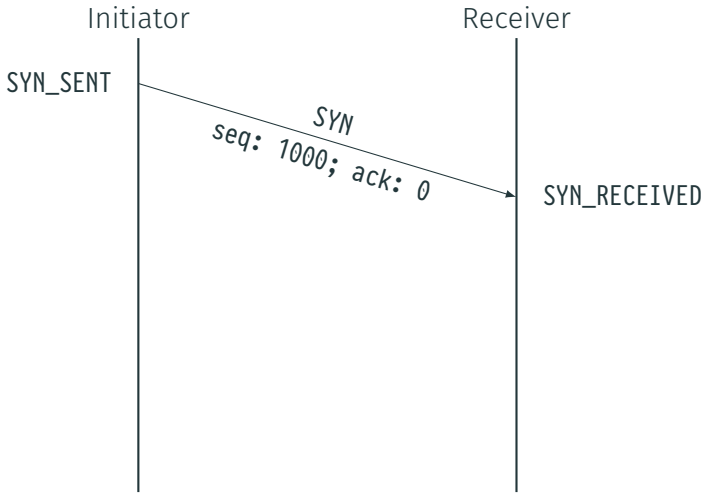
TCP Three-way Handshake—Remember?



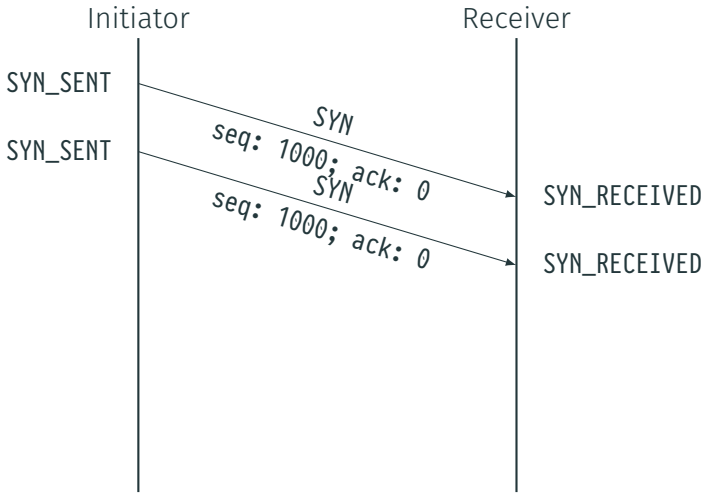
SYN-Flooding Attack (CVE-2017-14028)



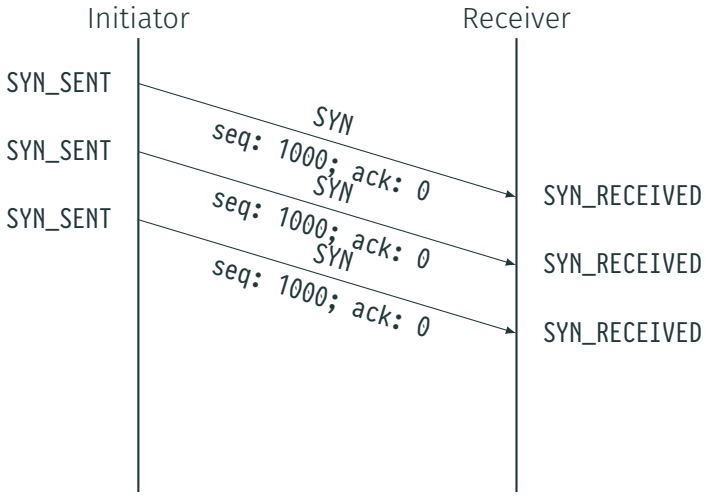
SYN-Flooding Attack (CVE-2017-14028)



SYN-Flooding Attack (CVE-2017-14028)



SYN-Flooding Attack (CVE-2017-14028)



Moxa NPort 5110: Etherleaking (CVE-2017-16715)

```
Welcome to Scapy (unknown.version)
$ sr1(IP(dst="192.168.127.254")/ICMP())
Begin emission:
......Finished to send 1 packets.
```

Moxa NPort 5110: Etherleaking (CVE-2017-16715)

```
Welcome to Scapy (unknown.version)
$ sr1(IP(dst="192.168.127.254")/ICMP())
Begin emission:
......Finished to send 1 packets.

Received 7 packets, got 1 answers, remaining 0 packets
<IP  version=4L ihl=5L tos=0x0 len=28 id=39957 flags=
frag=0L ttl=255 proto=icmp chksum=0x9f7a src=192.168.
127.254 dst=192.168.127.1 options=[] |<ICMP  type=echo
-reply code=0 chksum=0xffff id=0x0 seq=0x0 |<Padding
load='\xf8\x12\xe0\xbfP\x11\x10\x00\xc5\xdf\x00\x00HTTP/1' |>>>
```

Moxa NPort 5110: Etherleaking (CVE-2017-16715)

- Vulnerable to Etherleaking
  - Expose portion of the kernel memory
- According to RFC 894 an Etherframe $\geq$ 46 Bytes
- What if a packet is smaller?

RFC 894:

IP packet "*should be padded (with octet of zero) to meet the Ethernet minimum frame size*"

Problem of Etherleaking (Example)

```
void xmit_frame(char *frame_buf, int frame_len) {
    int length;

    if (frame_len < MIN_FRAME_SZ)
        length = MIN_FRAME_SZ;
    else
        length = frame_len;

    copy_to_tx_buf(frame_buf, length);

    return;
}
```

Problem of Etherleaking (Example)

```
void xmit_frame(char *frame_buf, int frame_len) {
    int length;

    if (frame_len < MIN_FRAME_SZ)
        length = MIN_FRAME_SZ;
    else
        length = frame_len;

    copy_to_tx_buf(frame_buf, length);

    return;
}
```

Problem of Etherleaking (Example)

```
void xmit_frame(char *frame_buf, int frame_len) {
    int length;

    if (frame_len < MIN_FRAME_SZ)
        length = MIN_FRAME_SZ;
    else
        length = frame_len;

    copy_to_tx_buf(frame_buf, length);

    return;
}
```

Using uninitialized memory as padding.

How to fix it?

```
void xmit_frame(char *frame_buf, int frame_len) {
    int length;

    if (frame_len < MIN_FRAME_SZ) {
        length = MIN_FRAME_SZ;
        memset(frame_buf + frame_len, 0, length — frame_len);
    } else
        length = frame_len;

    copy_to_tx_buf(frame_buf, length);

    return;
}
```

Moxa NPort 5110: TCP ISN prediction 1/2 (CVE-2017-16719)

RFC 1948

The initial sequence numbers are intended to be more or less random. More precisely, RFC 793 specifies that the 32-bit counter be incremented by 1 in the low-order position about every 4 microseconds.

- TCP Initial Sequence Number (ISN) should be competently random
- If not $\rightarrow$ Attacker can predict the next SN and inject arbitrary packets into an established TCP connection
- We found out that Moxa Embedded uses the uptime of the device as ISN

Moxa NPort 5110: TCP ISN prediction 2/2 (CVE-2017-16719)

```
$ sudo python2.7 tcp-isn.py
   ISN    Difference
 1903050    +95
 1903125    +75
 1903220    +95
 1903310    +90
 [...]
```

- An attacker can get the uptime of the device via SNMP by requesting the OID `sysUpTimeInstance`

Moxa NPort 5110: Connection Blocking

- NPort 5110 uses TCP port 950/966 for the serial connection
  - TCP Port 966: signaling
  - TCP Port 950: data
- Both port accept connections without authentication
- Only 1 connection is allowed $\rightarrow$ all other connections will be blocked
- An attacker needs to connect to both ports and hold the connection open
  - Legitimate users cannot connect to the device anymore

Moxa NPort 5110: No firmware verification

- If you have access to web GUI you can upload firmware images
- Moxa NPort 5110 does not support firmware verification
- You can upload a malicious firmware and the device will write it to memory
- How did we found out?

Moxa NPort 5110: No firmware verification

- If you have access to web GUI you can upload firmware images
- Moxa NPort 5110 does not support firmware verification
- You can upload a malicious firmware and the device will write it to memory
- How did we found out?
  - We tested it and bricked one device ;-)

Hi-Fly DTU-E100

Hi-Fly DTU-E100

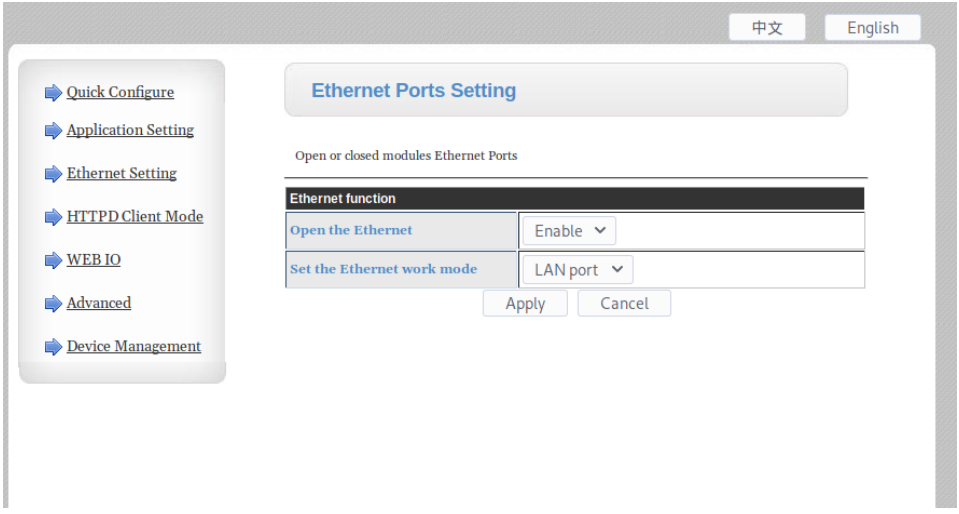
- Sold under different cases
- Support TCP/UDP/HTTP/TLS/Modbus Network Protocols
- Support RS232/RS485/Ethernet Data Interface

Hi-Fly DTU-E100

- Sold under different c
- Support TCP/UDP/HTTP/TLS/M Network Protocols
- Support RS232/RS485/ Interface

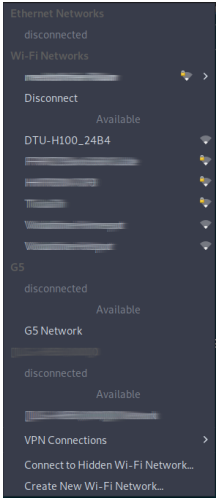
Available with Wireless LAN

Hi-Fly DTU-E100 Web Interface



Hi-Fly DTU-E100: Open Wireless LAN

- According to the manual this device should not have a wireless LAN
- According to the web GUI, this device should not have a wireless LAN
- We found out this device has an un-encrypted and open wireless LAN with the essid `DTU-H100_24B4`
- Can only deactivated via telnet with AT+T commands



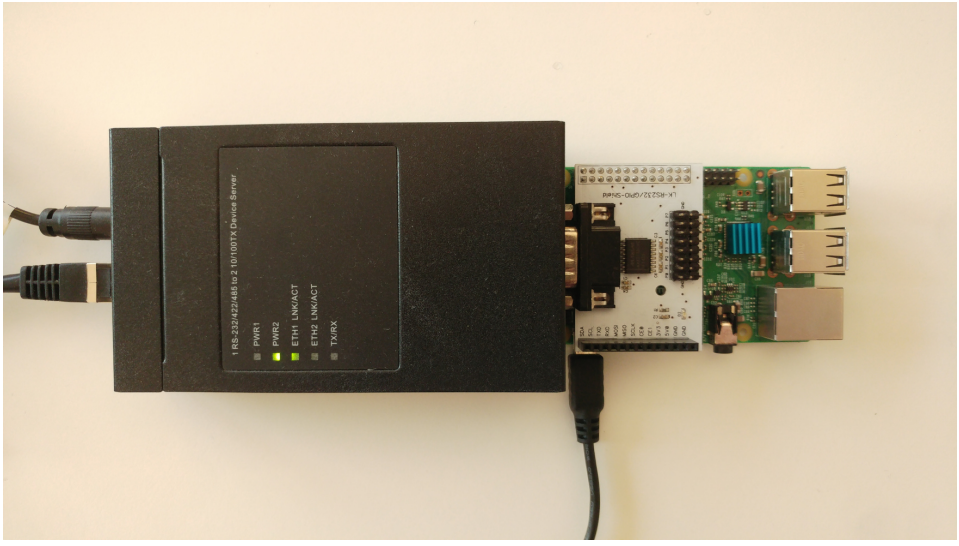
Hi-Fly DTU-E100: Connection Blocking

- Similar to Moxa NPort device
- This device allows 32 TCP connection simultaneously
- An attacker just has to start 32 TCP connections on that port to block any legitimate user to reach this device
- This is also possible when password authentication is enabled

Hi-Fly DTU-E100: Flooding Attack

- Default TCP window size is very low
- A couple of larger TCP packets are enough to exhausted the window size and the device cannot receive any packets

Lantech IDS-2102



Web Interface

MENU

Serial Device Server

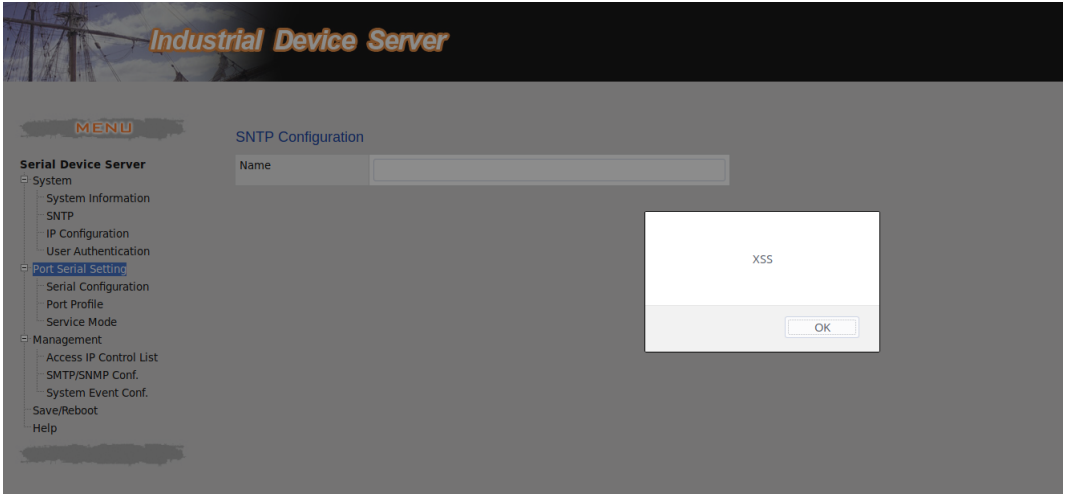
- System
  - System Information
  - SNTP
  - IP Configuration
  - User Authentication
- Port Serial Setting
  - Serial Configuration
  - Port Profile
  - Service Mode
- Management
  - Access IP Control List
  - SMTP/SNMP Conf.
  - System Event Conf.
- Save/Reboot
- Help

IP Configuration

IP Configuration	Static
IP Address	192.168.10.2
Netmask	255.255.255.0
Gateway	192.168.10.1
DNS Server 1	192.168.10.1
DNS Server 2	
Auto IP Report	
Auto Report to IP	
Auto Report to TCP Port	0
Auto Report Interval	0 seconds
Ethernet Mode	
Ethernet Mode	Redundant / Switch

Apply

XSS (CVE-2018-8869)



Firmware extraction

```
$ binwalk -e IDS-2102_V1.1h.bin

DECIMAL    HEXADECIMAL   DESCRIPTION
--------------------------------------------------------------------------------
84600      0x14A78       CRC32 polynomial table, little endian
143308     0x22FCC       gzip compressed data, maximum compression
1171828    0x11E174      gzip compressed data, has file name: "ramdisk"
```

Comparision between ser2net

Radare2 Analyse of the Binary

```
[0x0000a688]> afl~sel_
0x00017240 sym.sel_set_fd_handlers
0x0001728c sym.sel_clear_fd_handlers
0x00017344 sym.sel_set_fd_read_handler
0x0001739c sym.sel_set_fd_write_handler
0x00017400 sym.sel_set_fd_except_handler
```

Comparision between ser2net

Radare2 Analyse of the Binary

```
[0x0000a688]> afl~sel_
0x00017240 sym.sel_set_fd_handlers
0x0001728c sym.sel_clear_fd_handlers
0x00017344 sym.sel_set_fd_read_handler
0x0001739c sym.sel_set_fd_write_handler
0x00017400 sym.sel_set_fd_except_handler
```

Ser2net (Open Source)

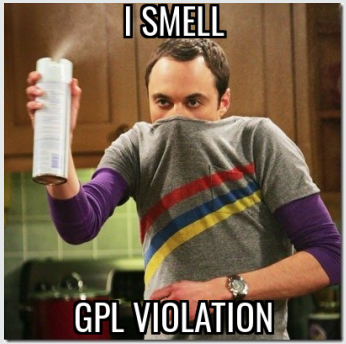
```
$ cat selector.h | grep "^void_sel"
void sel_set_fd_handlers(...)
void sel_clear_fd_handlers(...)
void sel_set_fd_read_handler(...);
void sel_set_fd_write_handler(...);
void sel_set_fd_except_handler(...);
```

Comparision between ser2net

Radare2 Analyse of the Bi

```
[0x0000a688]> afl~sel_
0x00017240 sym.sel_set_fd_ha
0x0001728c sym.sel_clear_fd_
0x00017344 sym.sel_set_fd_re
0x0001739c sym.sel_set_fd_wr
0x00017400 sym.sel_set_fd_ex
```

GPL Violation?



n Source)

```
n | grep "^void_sel"
_handlers(...)
fd_handlers(...)
_read_handler(...);
_write_handler(...);
_except_handler(...);
```

How to examine the binary

1. Run binary with chroot and qemu-arm-static

```
sudo chroot . qemu—arm—static —g 1337 /usr/bin/ser2net —p 600 —c /etc/com2ip.conf
```

1. gdb-multiarch with gef

```
$ gdb—multiarch
gdb> set gnutarget elf32—littlearm
gdb> file /mnt/disk/usr/bin/ser2net
gdb> target remote localhost:1337
```

Configuration File

```
version:
model:DS-12
name:DeviceServer-DEFAULT
serialno:123456789-12-456
password:
network:static:192.168.10.2:255.255.255.0:AAAAAAAAAAAAAAAA:192.168.10.1
snmp:empty:empty:empty:empty:empty:empty:empty
email:::::::::
management:web:telnet:
[...]
tty:port1:mapped:38400 1STOPBIT 8DATABITS NONE -RXFAST -RXPROBE:RS232:NONE: :
vir4000:raw:0,0,0,0,40:port1:38400 1STOPBIT 8DATABITS NONE -RXFAST -RXPROBE -R
```

Buffer 0verflow (CVE-2018-8865)

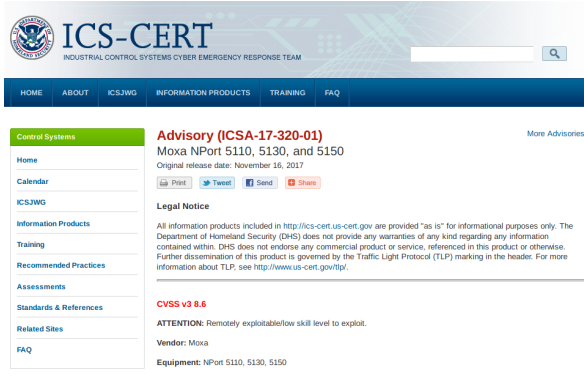
```
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
GEF for linux ready, type `gef' to start, `gef config' to configure
60 commands loaded for GDB 7.11.1 using Python engine 3.5
[*] 6 commands could not be loaded, run `gef missing` to know why.
gef➤  set gnutarget elf32-littlearm
gef➤  target remote localhost:1337
Remote debugging using localhost:1337
warning: Architecture rejected target-supplied description
0x00000000 in ?? ()
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────────────────────────────────────[ registers ]────
[!] Command 'context' failed to execute properly, reason: 'NoneType' object has no attribute 'all_registers'
gef➤  file /mnt/disk/usr/bin/ser2net
Reading symbols from /mnt/disk/usr/bin/ser2net...(no debugging symbols found)...done.
gef➤  continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x41414140 in ?? ()
[ Legend: Modified register | Code | Heap | Stack | String ]
──────────────────────────────────────────────────────────────────────────────────────────────────────────────[ registers ]────
$r0   : 0x0002f4c6  →  0x35322e30  →  0x35322e30
$r1   : 0xf6fff01c  →  0x00000000  →  0x00000000
$r2   : 0x090304b5  →  0x090304b5
$r3   : 0x00000000  →  0x00000000
$r4   : 0x300e4678  →  0x300e4678
$r5   : 0x1a499001  →  0x1a499001
$r6   : 0x27081a92  →  0x27081a92
$r7   : 0x370351c2  →  0x370351c2
```

Conclusion

Responsible Disclosure

- All vulnerabilities have been sent to ICS-CERT
- They contacted the companies
- After $\pm$ 3 months $\rightarrow$ public advisory



Moxa

Moxa

Example (Solved in firmware 2.8)

- SYN Flooding (CVE-2017-14028)
- Etherleaking (CVE-2017-16715)
- TCP ISN Prediction (CVE-2017-16715)

Moxa

Example (Solved in firmware 2.8)

- SYN Flooding (CVE-2017-14028)
- Etherleaking (CVE-2017-16715)
- TCP ISN Prediction (CVE-2017-16715)

Still vulnerable

- Firmware Verification
- Connection blocking

Hi-Fly

Example (Solved)

Example (Solved)

Still vulnerable (No Answer)

- Open WLAN
- Connection blocking
- Small TCP window size

Lantech

Example (Solved)

Example (Solved)

Still Vulnerable

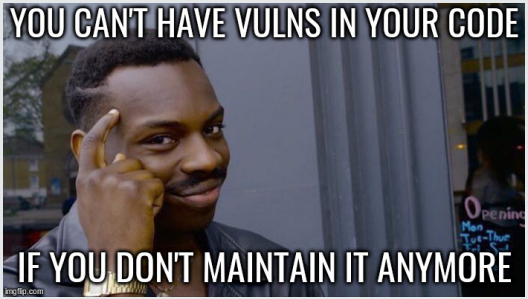
- XSS (CVE-2018-8869)
- Buffer Overflow (CVE-2018-8865)

Example (Solved)

Still Vuln

- XSS (
- Buffe

Think about it



Conclusion

- Found several severe security vulnerabilities on industrial serial device servers
- Often used in critical infrastructure
- If you use them, take extra care to protect these devices

Questions?

florian-hacklu@adamsky.it
https://florian.adamsky.it
Twitter: **@c1t**