DRAGOS

INDUSTRIAL CONTROL SYSTEMS CYBERSECURITY

SAFEGUARDING CIVILIZATION

RIPPLE20: WHAT YOU NEED TO KNOW

REID WIGHTMAN & KATE VAJDA

Vulnerability Researchers

Reid Wightman

- Principal Vulnerability Analyst
- Embedded systems
- Lead exploitation
- Does the hard work
- Fancy home lab

&

Kate Vajda

- Senior Vulnerability Analyst
- Sys & network admin
- Penetration tester
- Tags along
- Growing home lab

What you need to know

JSOF's Findings:

https://www.jsof-tech.com/ripple20/

Key takeaways for the following:

Devices Impacted

Devices that rely on the Treck TCP/IP stack

Prevention Strategy

Strategies for blocking malicious targeting

Mitigation Tactics

Strategies for mitigating the risks associated

Research Implications

What this means for the community

Vulnerabilities identified in Treck TCP/IP Stack

JSOF Research Team White Paper

- Released with demo
- Focused on 2/19 vulns identified
- JSOF talk tomorrow (8/5) at BlackHat

Vendor Coordination

- Several vendors list which devices are affected, more importantly;
- How they are affected
  - DoS or RCE?

Device identification

- Embedded systems with no native IP stack
- Not so easy to detect all vulns (see ‘fixed dates’ in JSOF advisory)
- Many CPU and even ‘overall device’ architectures

DRAGOS

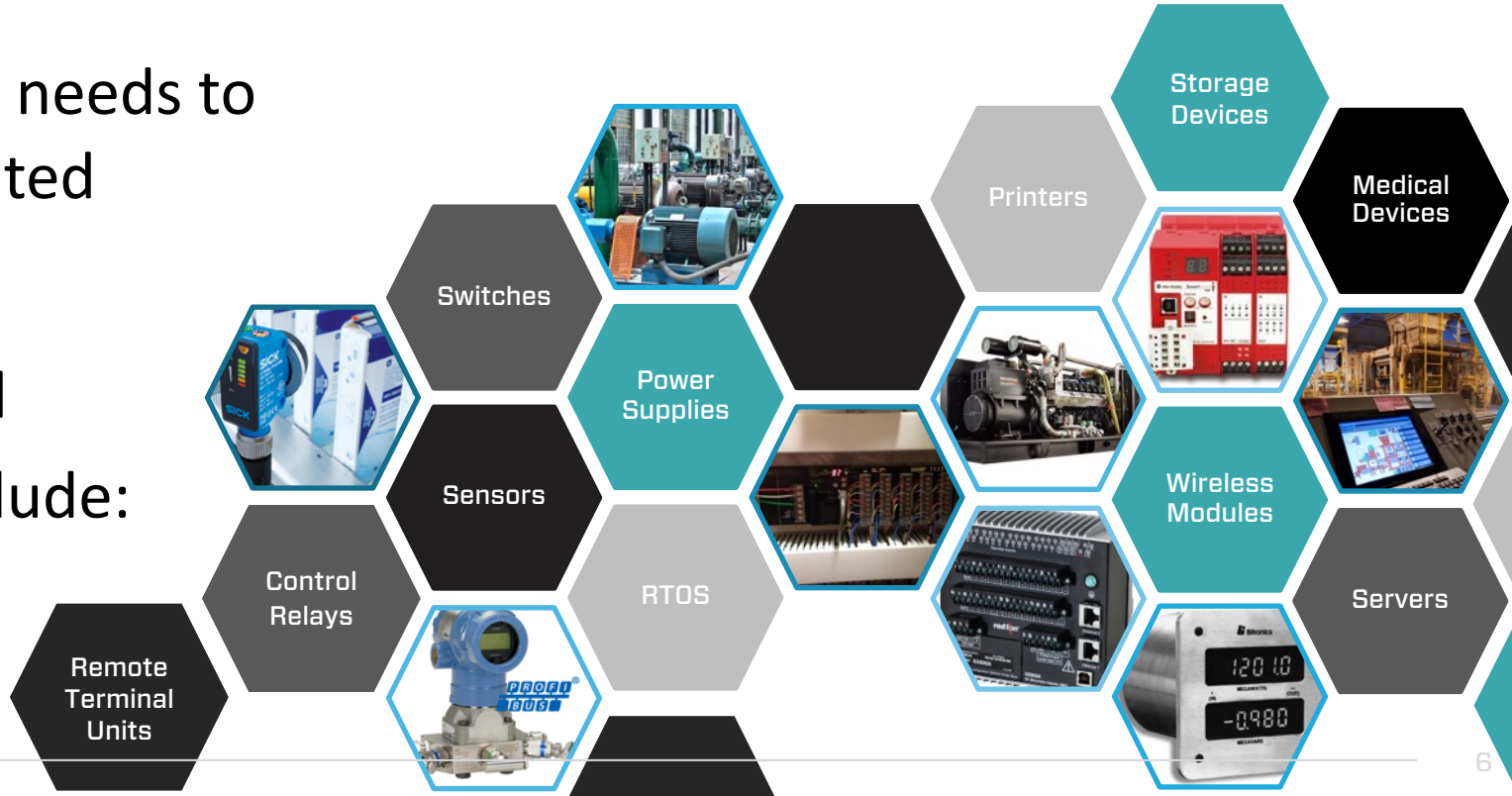
The Bugs

- “Treck TCP/IP” stack is really an IP stack
- The ‘big’ bugs (3) include memory corruption as outcome
- The ‘more minor’ (16) bugs result in out-of-bound reads/memory leaks
- Impacts vary by devices, some bugs fixed/semi-fixed years ago according to JSOF

Devices with Treck TCP/IP stack

Each device needs to be investigated

Devices and systems include:



Devices Tested

A subset of devices affected; these are currently part of our research for Ripple20 vulnerabilities.

These devices live in our home labs

APC SmartUPS

Network-controlled uninterruptable power supply

Digi Connect Wi ME 9210

Embedded wireless network interface

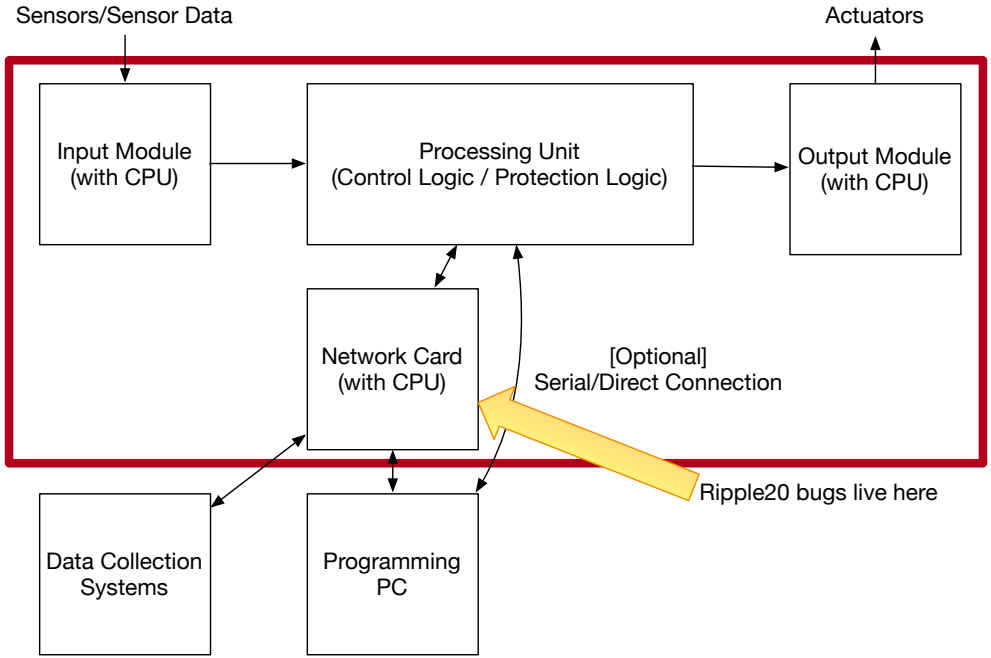
SCADAPACK 32P RTU

Programmable remote terminal unit

ABB REF615

Feeder protection and control device

Generic PLC Architecture

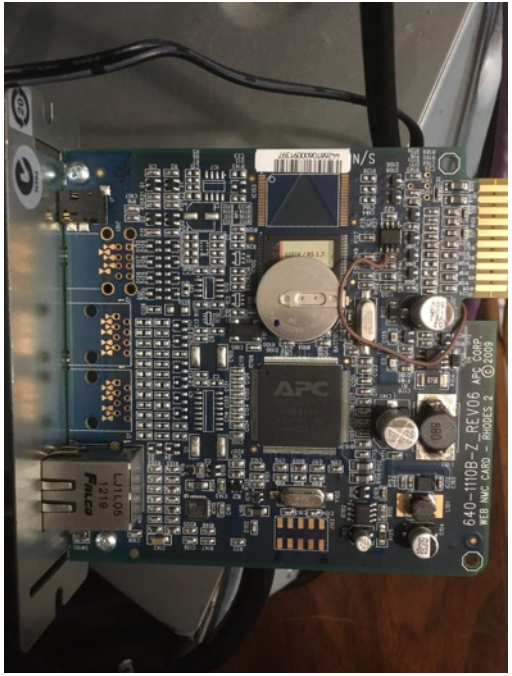


Generally, Ethernet processors

Deep dive into APC SmartUPS

- Devices inspected: Smart-UPS with 963X-series Ethernet cards
- Cards run uC/OS-II operating system with Treck stack on an ASIC (x86-ish – possibly RDC C62xx-series DSP)
- Cards communicate with UPS board, including ‘lights out’ type management (more later)

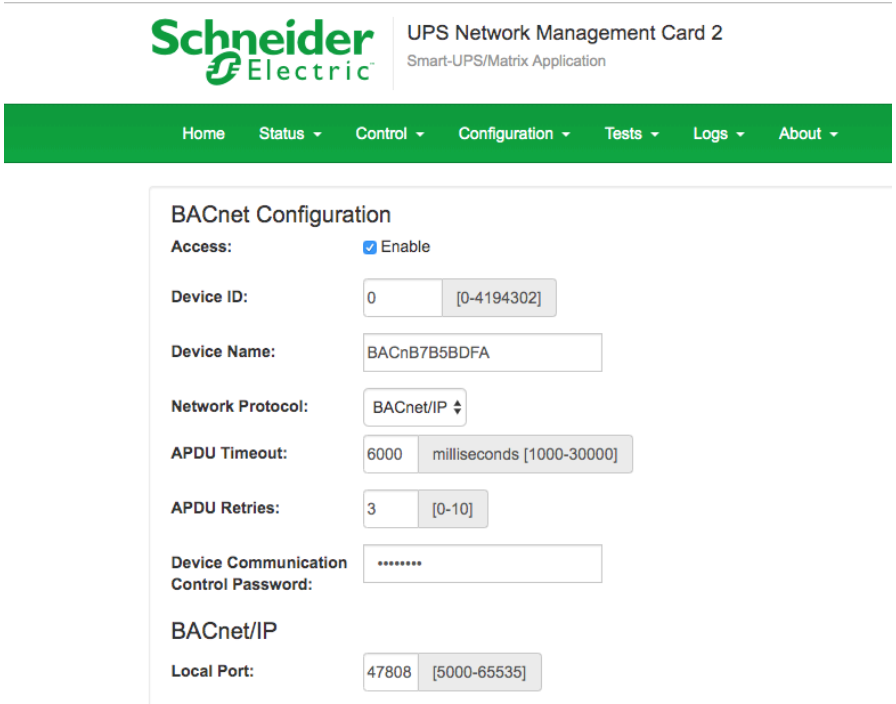
Deep dive into APC SmartUPS



Deep dive into APC SmartUPS

- 963X cards support:
  - SNMP
  - BACnet
  - Modbus/TCP
- BACnet allows ‘shutting the UPS off’ by design
- Modbus/TCP allows poweroff on SOME UPS models (not SmartUPS though)

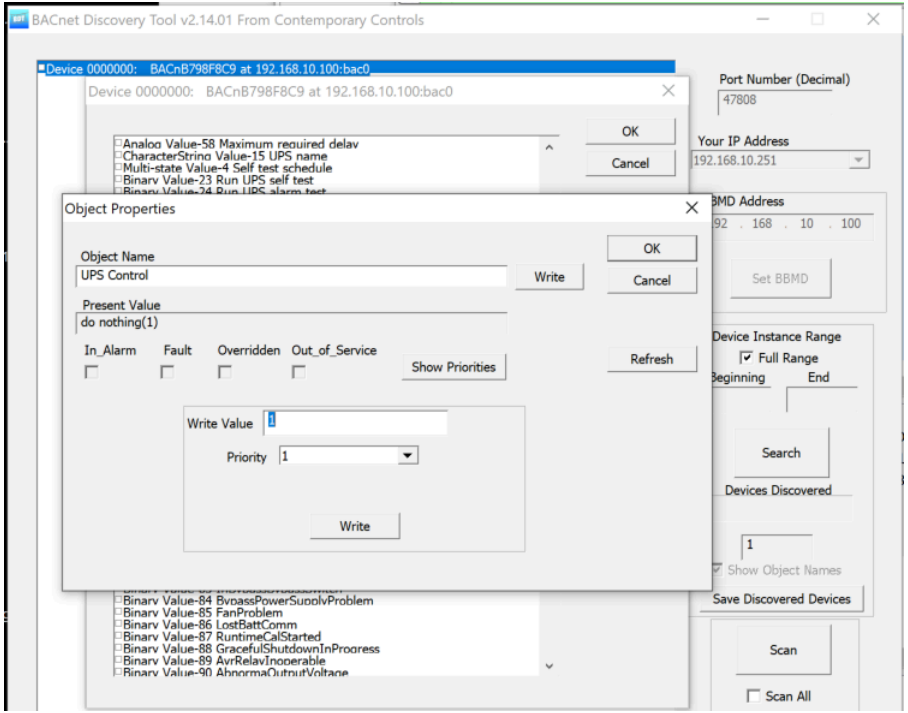
Deep dive into APC SmartUPS



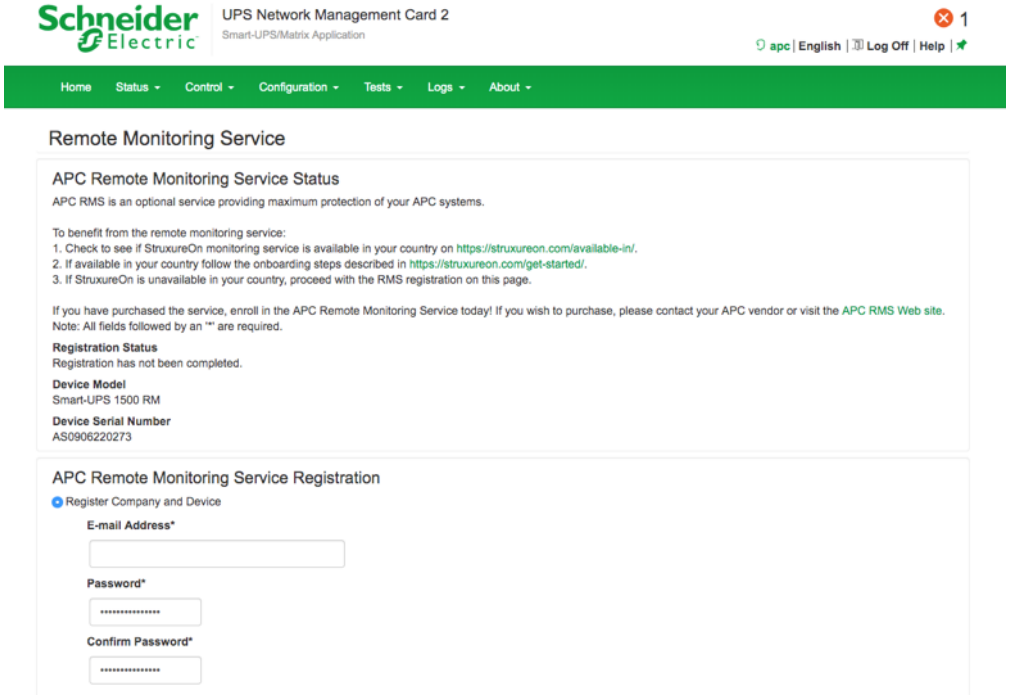
Deep dive into APC SmartUPS

6	79691782	UPS Control	Control the UPS state, based on the options available.	do nothing, on immediately, on delayed, off immediately, off delayed, bypass, return from bypass, reboot immediately, reboot delayed, sleep immediately, sleep delayed	RW	x	x	x	x
7	79691783	Main Outlet Group control	Control the state of each outlet group.	do nothing, cancel, on immediately, on delayed, off immediately, off delayed, reboot immediately, reboot delayed, shutdown immediately with ac restart, shutdown delayed with ac restart	RW	x	x		
8	79691784	Switched Outlet group 1 control	Control the state of each outlet group.	do nothing, cancel, on immediately, on delayed, off immediately, off delayed, reboot immediately, reboot delayed, shutdown immediately with ac restart, shutdown delayed with ac restart	RW	x	x		
9	79691785	Switched Outlet group 2 control	Control the state of each outlet group.	do nothing, cancel, on immediately, on delayed, off immediately, off delayed, reboot immediately, reboot delayed, shutdown immediately with ac restart, shutdown delayed with ac restart	RW	x	x		
10	79691786	Switched Outlet group 3 control	Control the state of each outlet group.	do nothing, cancel, on immediately, on delayed, off immediately, off delayed, reboot immediately, reboot delayed, shutdown immediately with ac restart, shutdown delayed with ac restart	RW	x	x		

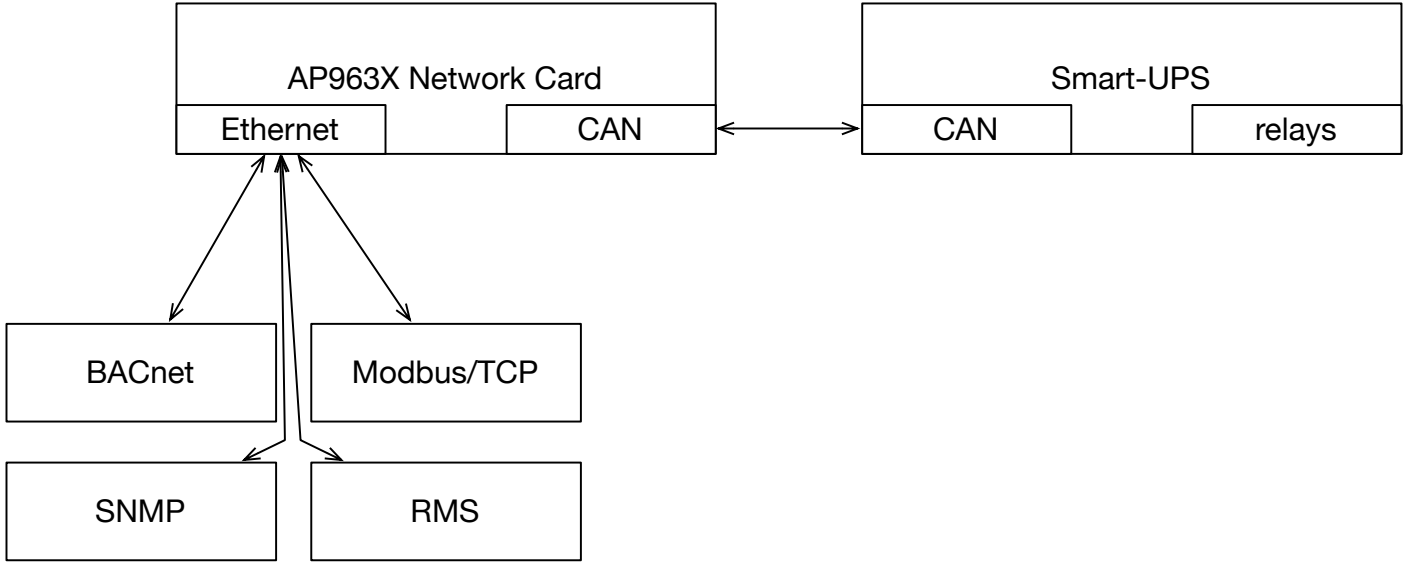
Deep dive into APC SmartUPS



Deep dive into APC SmartUPS



Deep dive into APC SmartUPS



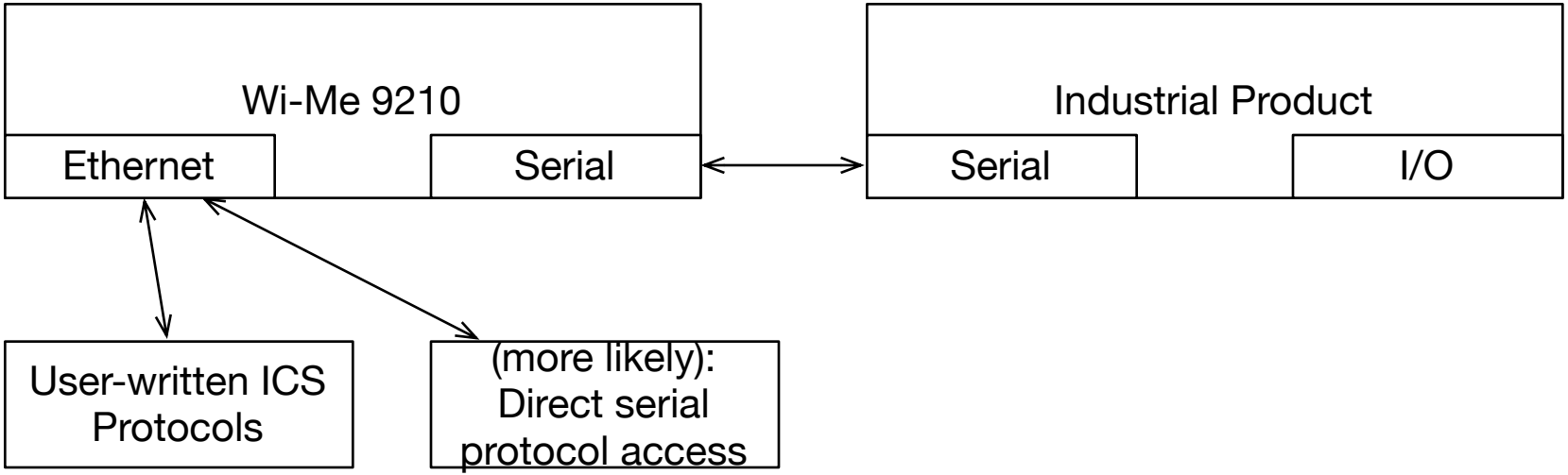
Deep dive into APC SmartUPS

- Summary:
  - Crashing the 963x != disabling UPS protection
  - Remotely powering off/on UPS is a design feature of BACnet (all models) and Modbus (some models)
  - Actually exploiting the bug to poweroff device requires RE, learning CAN commands (or, calling the firmware funcs)
- Before freaking out, keep these things in mind

Shallow dive into Digi Connect Wi ME 9210

- Serial converter requires device maker to modify firmware
- Device has an ARM processor running “NET+OS”
- Default firmware provides access to UDP/2362 (Digi Discovery Protocol)
- Only device (so far) reported to be vulnerable to CVE-2020-11896 for RCE
- h/t to Finite State for walking us through device firmware

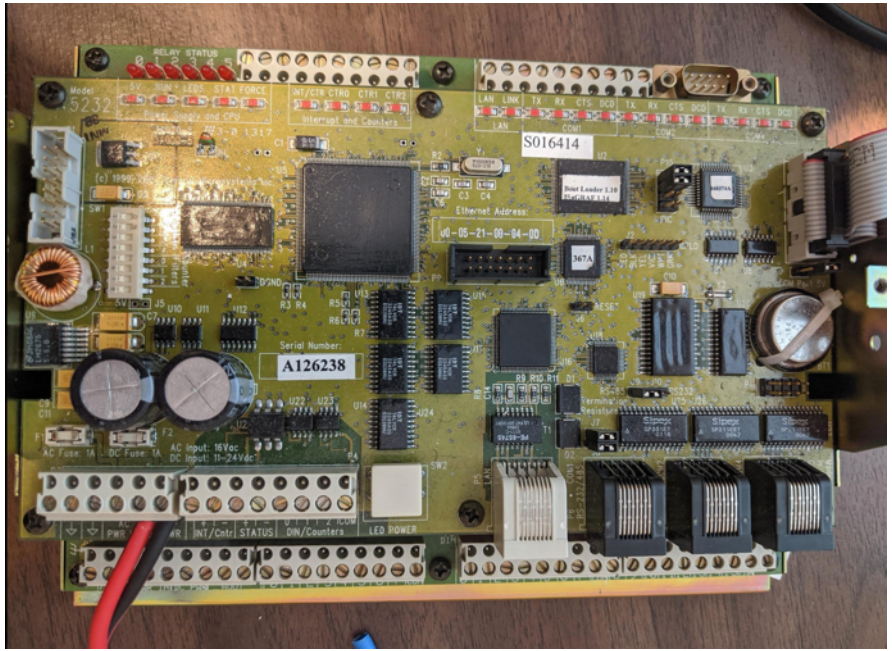
Deep dive into Digi Connect Wi ME 9210



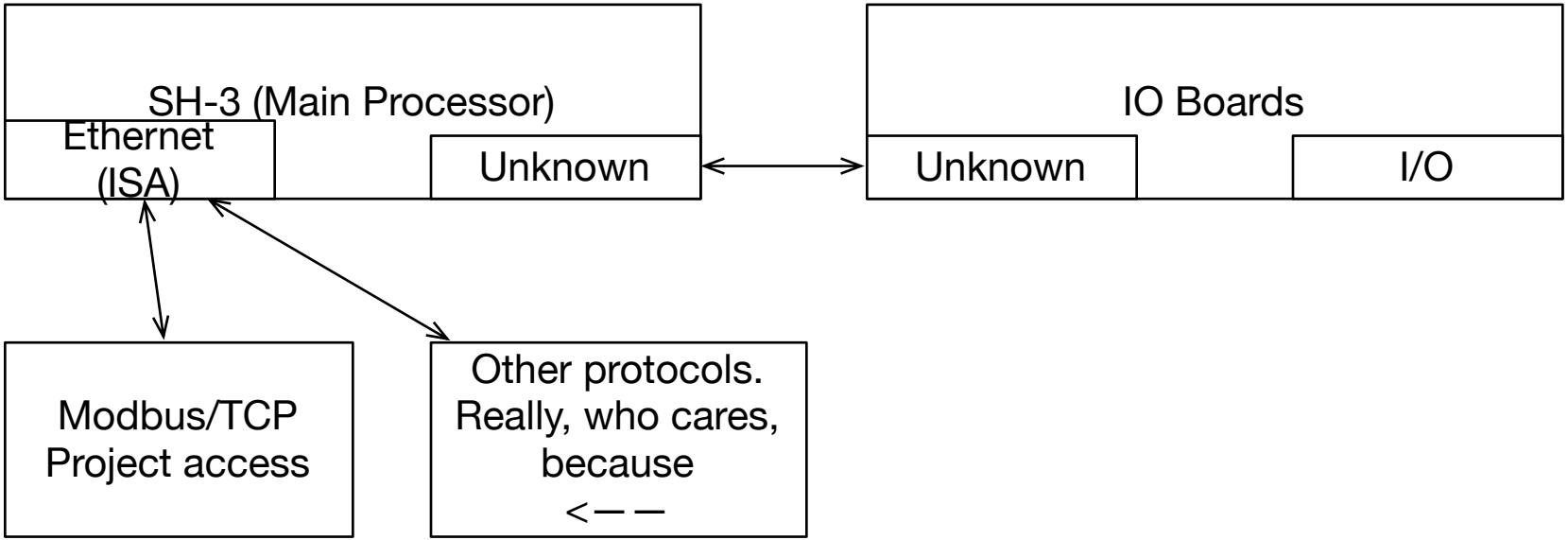
Deep dive into Schneider Electric SCADAPack RTU

- All SCADAPack 32 RTUs affected
- Device inspected: SCADAPack 32 P4
- Runs on SH-3 CPU/Unknown OS
- Logic and Ethernet on one set of firmware

Deep dive into Schneider Electric SCADAPack RTU



Deep dive into Schneider Electric SCADAPack RTU

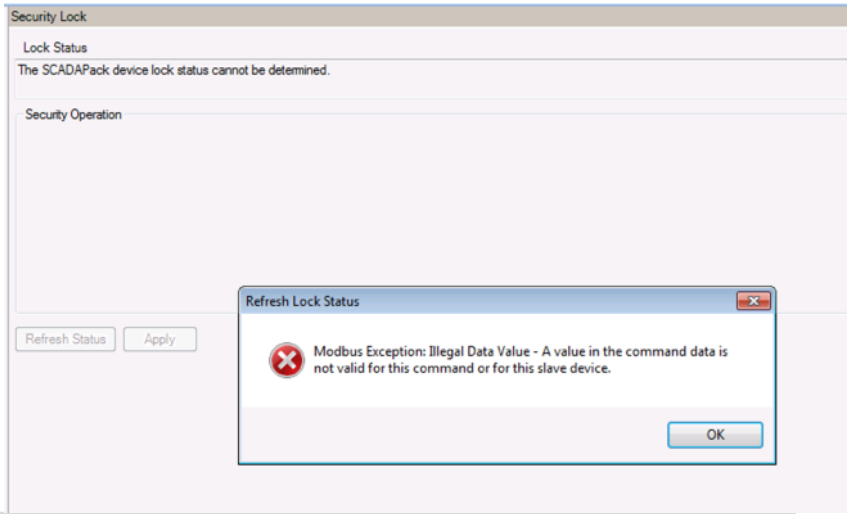


Deep dive into Schneider Electric SCADAPack RTU

Logic transfer done using Modbus/TCP (proprietary function code)

No security on project transfer

Check out our device →



Deep dive into Schneider Electric SCADAPack RTU

Modbus TCP (TCP/502)
Modbus RTU (UDP/49152)
Modbus ASCII (UDP/49153)

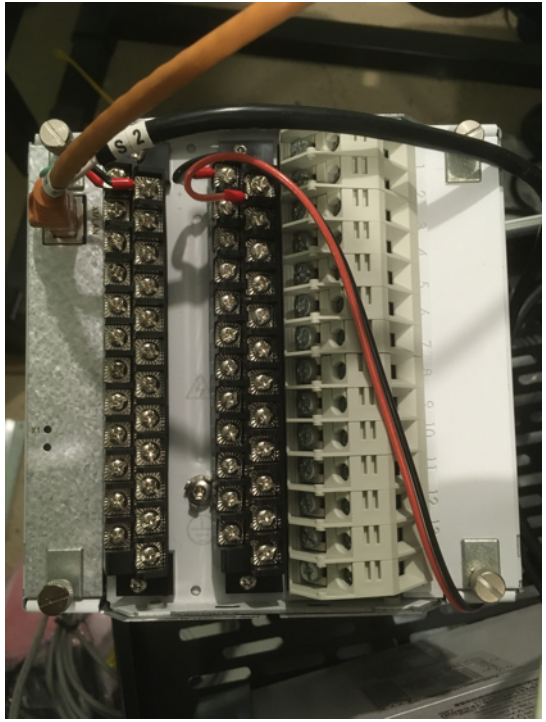
DNP requires activation (TCP/20000 and UDP/20000)

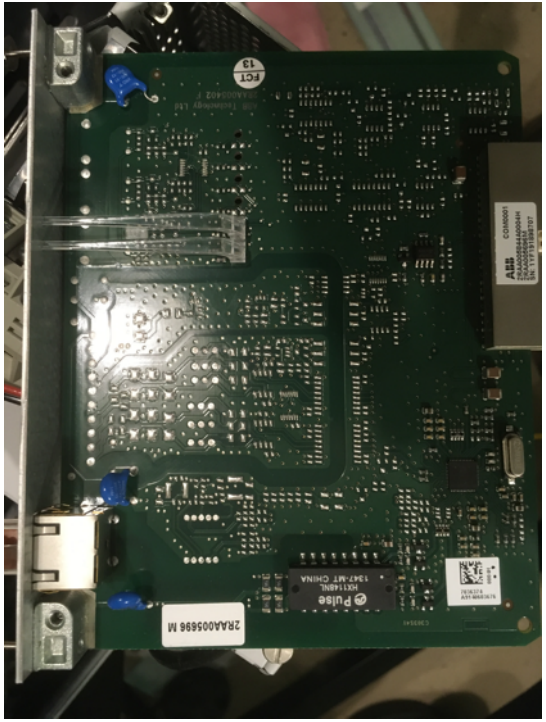
Communications	
Serial Port COM1	Configurable RS-232 or RS-485, 2 wire half duplex or 4 wire full/half duplex
Serial Ports COM2, COM4	• RS-232, DTE, 8 pin modular jack, full or half duplex with RTS/CTS control • Implemented Td, Rd, CTS, RTS, DCD, DTR, +5V
Serial Port COM3	Located on 5604 I/O module. Same specifications as COM2 and COM4
Baud Rates COM1, COM2, COM4	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200
Baud Rate COM3	1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200
Serial Protocols	Modbus RTU, Modbus ASCII, DNP3, DF1, PPP
Ethernet Port	RJ45, 10BaseT
Network Protocols	IP: ARP, TCP, TFTP, UDP, ICMP
Ethernet Port Protocols	Modbus TCP, Modbus RTU in UDP, Modbus ASCII in UDP, DNP in TCP, DNP in UDP
Wireless[1]	Spread spectrum radio at 900MHz[2] and 2.4GHz[2]

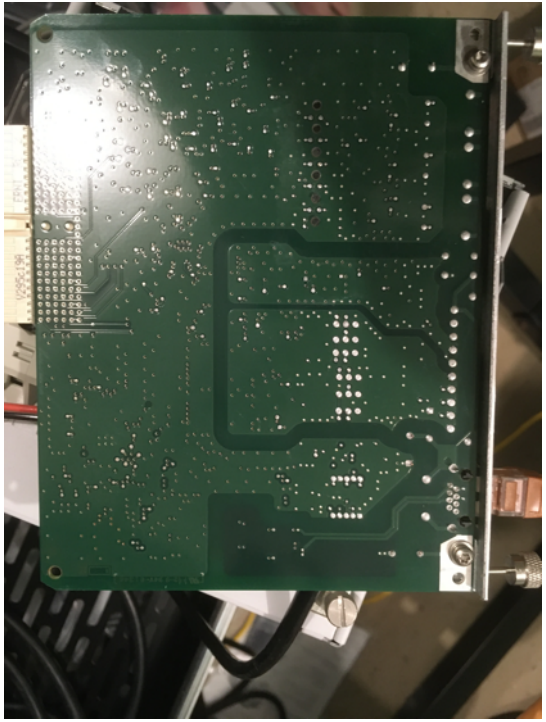
Deep dive into ABB REF615

- Odd hybrid network architecture
- CPU looks like a PowerQUICC (haven’t got the main board out yet though!)
- Ethernet appears to be a separate board, but…

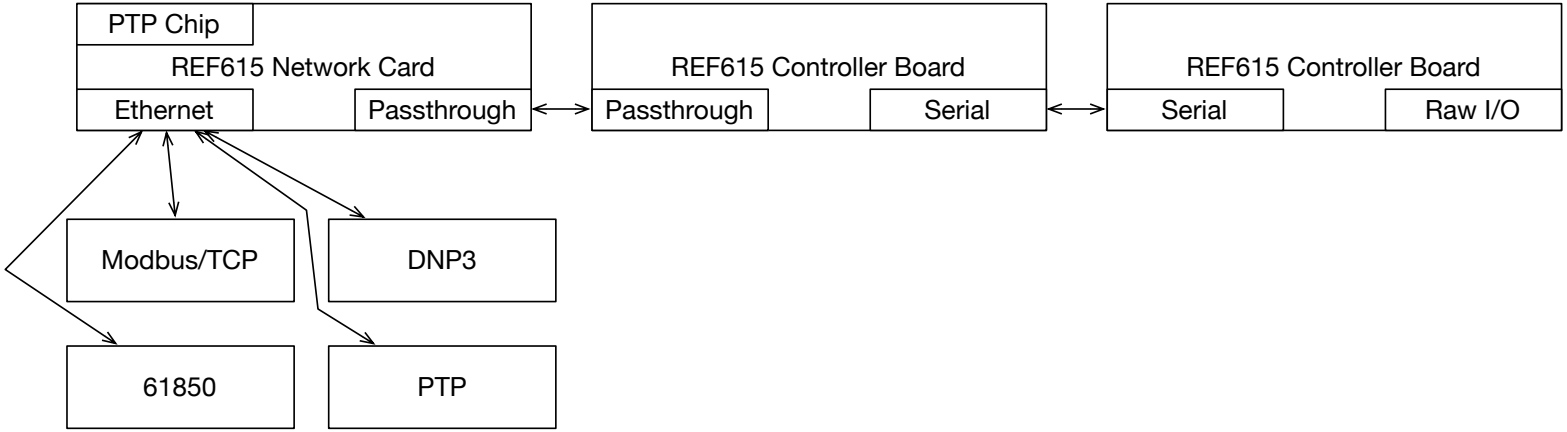
Deep dive into ABB REF615







Deep dive into ABB REF615



Deep dive into ABB REF615

- One of the more worrisome vulnerable products, strangely
- Protection logic is updated via FTP
- Crashing device may impact ability to protect against electrical issues, memory leaks actually useful
- No immediate impact from this loss, but as part of a coordinated effort…could be bad

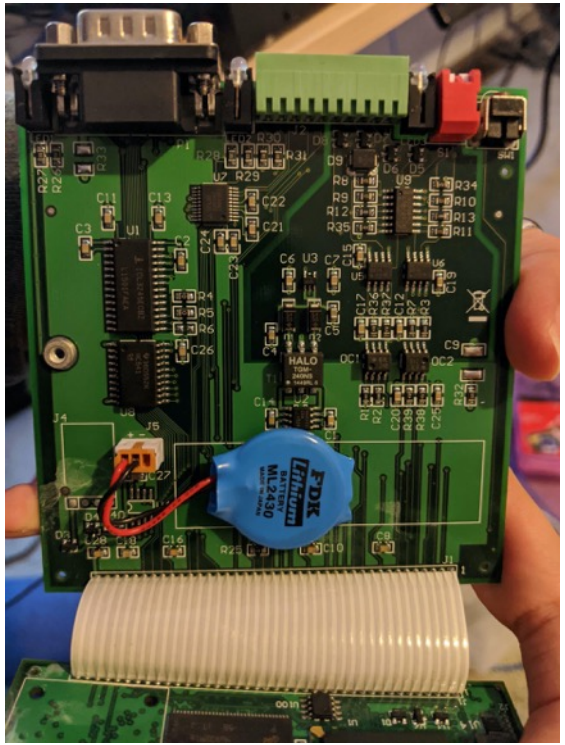
Deep dive into Opto22 SNAP-PAC-S1

Opto22 SNAP-PAC-S1: all versions

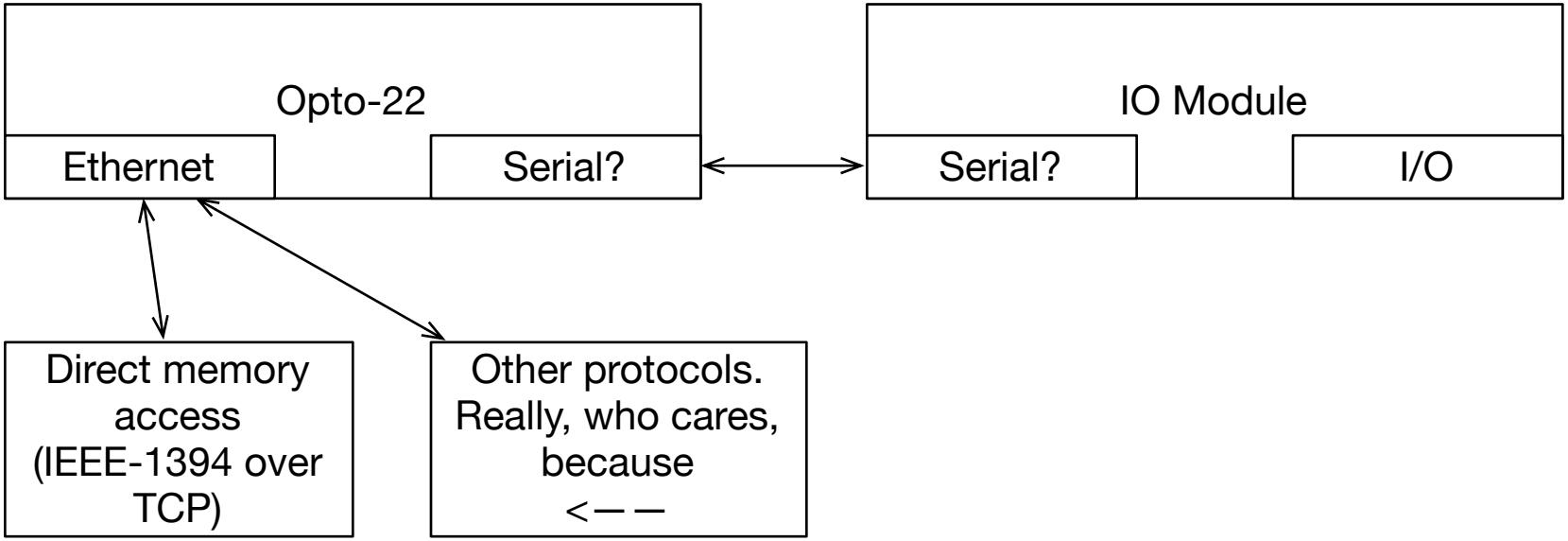
Coldfire / Motorola CPU

Also same processing

Deep dive into Opto22 SNAP-PAC-S1

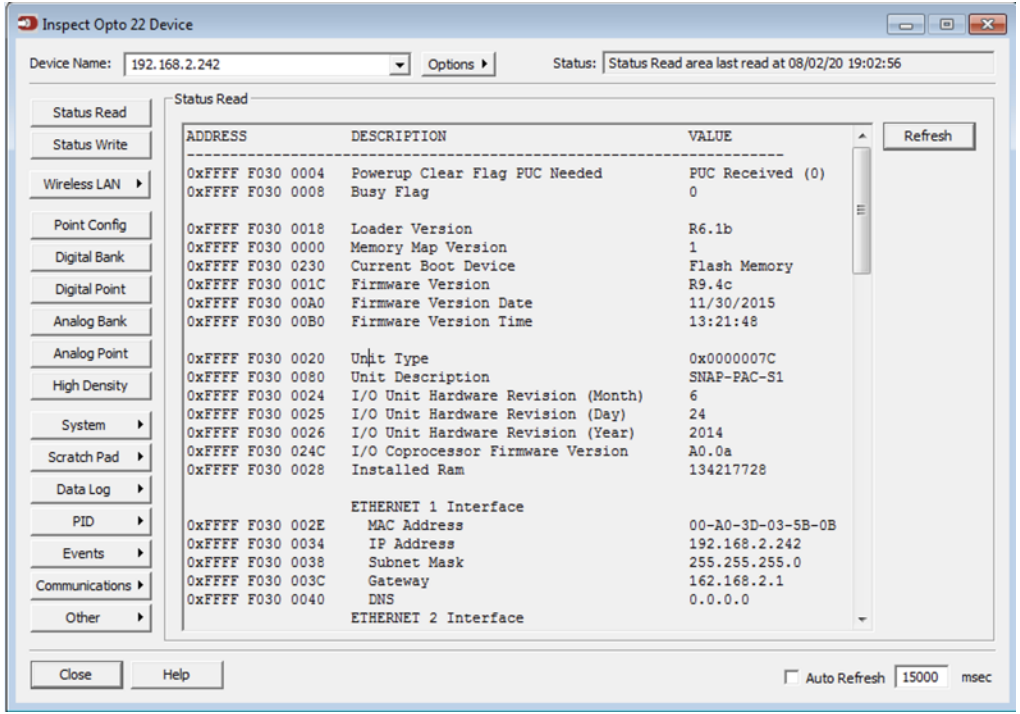


Deep dive into Opto22 SNAP-PAC-S1



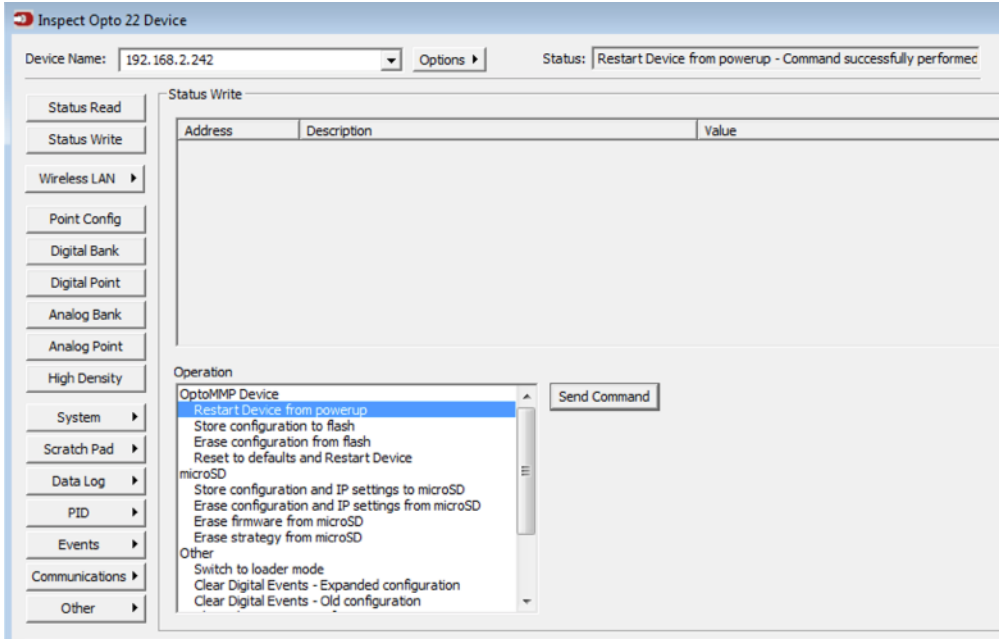
Deep dive into Opto22 SNAP-PAC-S1

Direct memory access, the software even gives you the addresses!



Deep dive into Opto22 SNAP-PAC-S1

Device can be restarted through OptoMMP protocol, no authentication necessary.



Deep dive into Opto22 SNAP-PAC-S1

Ports open :

- FTP (TCP/21)
- OptoMMP (TCP/2001)
- SNMP (UDP/161)

- Use IP filters
- Direct memory access

Impact Summary

Device	Loss of View	Loss of Control	Notes
APC Smart-UPS	Total	‘Soft’ loss	Configurable to allow unauth control
Wi-ME 9210	Total	N/A (in most systems, ‘Soft’)	
SCADAPack	Total	‘Hard’ loss	Insecure by Design
Opto22	Total	‘Hard’ loss	Insecure by Design
REF615	Total	‘Hard’ loss	Device has actual security

Current detection strategy for Ripple20

Active → Passive → Reversing

- Scanning for vulnerable devices (OS fingerprinting) https://github.com/LubyRuffy/FingerPrinting-Ripple20
- Nmap too

- Passive signatures (Dragos Platform!)
- Suricata: https://github.com/CERTCC/PoC-Exploits/blob/master/vu-257161/vu-257161.rules
- Zeek rules: https://github.com/corelight/ripple20

- Firmware static analysis for incorporated functions
- Verify vulnerability

DRAGOS

Prevention Strategy

Three severe vulnerabilities are blocked by preventing IP-over-IP, IPv6, and DNS.

The remaining vulns are less severe, and are blocked by restricting ICMP, 6to4, DHCP; however in most ICS these remaining vulns are not useful to an attacker.

Block IP-over-IP

Restricts the easy ‘denial of service’ vuln

Block or restrict DNS

If absolutely required, configure control systems DNS servers to only allow forwarding of your domain requests

Restrict other services

Majority of vulns are in DNS, DHCP, and ICMP processing. Most are memory leaks, which are not as useful to ICS attackers.

Impacted Devices: Silver Lining

‘Classic’ ICS devices

- Have a separate network cards with separate CPU
- Architecture means vulns have less immediate impact
- Affecting the device function means a lot of research, post-exploitation

VS

‘Newer’ ICS devices

- More likely to cram network stack and logic/protection processing on one CPU
- Ironically /less likely/ to run Treck (Opto22 is a counterexample though!)

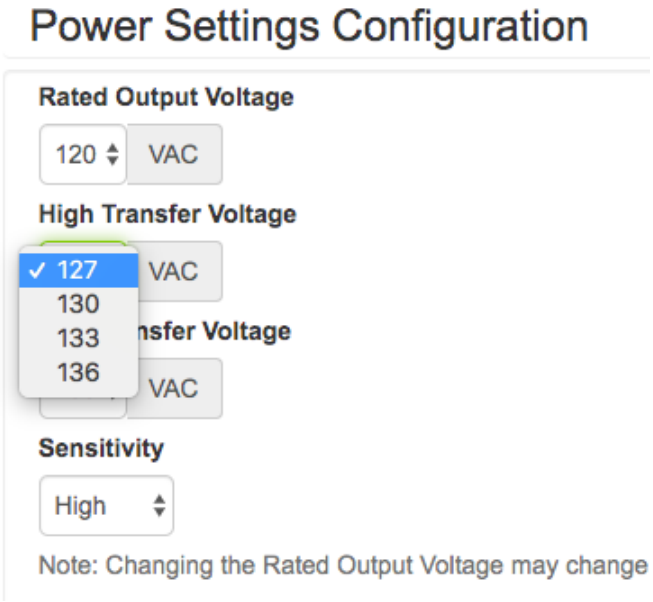
Impacted Devices: Silver Lining

- No active exploitation (yet)
- Bugs similar in nature to URG/11 vulnerabilities (which, is also seeing no active exploitation)
- Achieving RCE requires a lot of effort: CPU architecture, operating system, etc differ widely between devices

Research Implications

- DoS and even RCE are fun, but…
- …what is the impact on an actual process?
- Take APC UPS for example: pwning the card doesn't get us much right away
  - Can power the device off (but so can BACnet or Modbus*)
  - UPS function still works in case of power surge…
  - …unless we learn that CAN commands can change that

Research Implications



Questions?
(intel@dragos.com)

LINKS!

JSOF Paper:

https://www.jsof-tech.com/ripple20/

Fingerprinting (Active):

https://github.com/LubyRuffy/FingerPrinting-Ripple20

Suricata Signatures:

https://github.com/CERTCC/PoC-Exploits/blob/master/vu-257161/vu-257161.rules

Zeek rules:

https://github.com/corelight/ripple20