Fuzzing and protocol analysis case-study of DNP3

Adam Crain, Automatak
HMI/SCADA Master

External Control Points

SCADA Master Station/Control Center

Comm. Links

Remote Substation

1200 bps +
(down to 300 bps in actual installations)

Radio Microwave Spread-spectrum

Twisted-pair Fiber-optics Dial-up Leased line

Remote Terminal Unit (RTU)

Intelligent Electronic Devices

Actuator

Meter

Accumulator

Programmable Logic Controller (PLC)
Developed by Harris Corp, handed over to a vendor-neutral User Group in **1993**.

Many features have been “bolted on”, including security.
Layered Architecture

- **Application Layer**
  - Application Service Data Unit (ASDU)
  - Typical max size of 2KB
  - Semantics == functions + objects

- **Transport Layer**
  - Tx segmentation
  - Rx re-assembly of APDUs

- **Link Layer**
  - Adds CRCs and addressing. Error checking and (de) multiplexing.

- **User code**
  - IED/RTU or your SCADA master
Application layer messages

| function | Header + data (1) | ... | Header + data (N) |
Application-layer semantics

- **FUNCTION CODES**
  - READ
  - WRITE
  - OPERATE
  - CONFIRM
  - ..... 
  - RESPONSE
  - UNSOLICITED

- **OBJECTS**
  - Measurements, time sync, file transfer, controls, etc, etc

- **FUNCTION CODES**
  - \( \infty \) combinations
  - multiple types per message
  - Some function codes are “function only”
Project Robus

- Started in April 2013
- 30+ CVEs found via fuzzing
- Deep study of failure modes in one protocol
- [automatak.com/robus](http://automatak.com/robus)
Focus on serial / masters

A. Commonly expected direction of attack

B. Direction of newly discovered vulnerabilities

C. This direction of attack has always been possible
DNP3 Fuzzing

Num Test Cases

Test DNP3 Message (DL, TL, or AL)

Request Link States

Request

Response

Link Status

x Num Retry (10)
Common Faults

\[
\text{uint32_t count} = \text{stop} - \text{start} + 1; \quad \text{// \leftarrow \text{integer overflow}}
\]
Less Common Faults

Unexpected function code / object combinations

- Unsolicited Response
- Control Relay Output Block
- 1 byte start/stop
- buffer overrun
- not malformed!
- unexpected objects
- accepts broadcast
DNP3 Security

- Tightly coupled to the DNP3 application layer
- Auth-only
- New functions
- New objects
- 2 modes of authentication
Porous Trust Boundary

- Data is dangerous, intended function matters not.
- Every time you extend DNP3, you make it less secure.
- Optional challenges make security state machine overly complex
2 modes of authentication

Challenge-response – 2 pass authentication

"Aggressive mode" – 1 pass authentication
Aggressive mode message

| normal function | User ID & CSQ | Payload objects ... | HMAC |
Issue #1: Aggressive-mode ambiguity

You can only tell if this is an aggressive mode request by speculatively parsing the 1st object header. Ambiguity is dangerous.
Issue #2: Lack of an envelope for HMAC

DNP3 headers cannot be “skipped”. They must be parsed sequentially (at least lightly), so that you known where the next one starts.
“Session key status object”

- Total size framed by TLV in wrapping header
- Composed of fixed-size and variable-length subfields
- Final v-length field is the remainder of the encapsulation.
“Update key change reply”

- Total size framed by TLV in wrapping header
- Composed of fixed-size and variable-length subfields
- Final v-length field is the remainder of the encapsulation AND a length prefix.
A.45.12.2.3 Notes

This object shall always be used with a Qualifier of 0x5B, indicating that the object is of variable length up to 65535 octets, specified in the Object Prefix. The length of the Challenge Data may therefore be either calculated from the qualifier or read from the corresponding field of the object.
SA Conclusions

• Prefer a layered approach to SCADA security to that decouples legacy protocol encodings/semantics from security.

• Design security to address both function and implementation attack surface.
How can langsec help?

• Critical infrastructure vendors need better tools besides hand-rolled parsers.

• Standards bodies need the theory/guidance to produce better designs.

• Protocols need reference implementations to guide their evolution.
Questions?