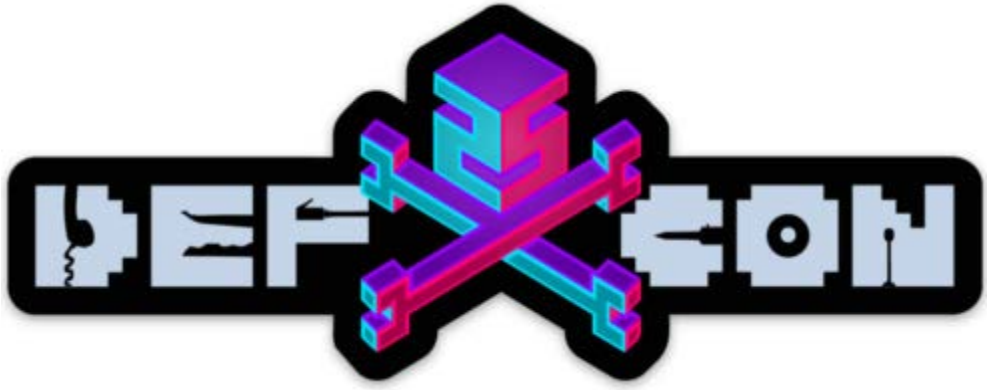


Breaking Wind: Adventures in Hacking Wind Farm Control Networks

Jason Staggs, Ph.D.
University of Tulsa
Tulsa, Oklahoma

whoami

- Security researcher
  - Focused on control systems and network security
- PhD in Computer Science from The University of Tulsa
  - Cellular networks, security engineering and forensics
- Presented “How to Hack your Mini Cooper”
  - @ DEFCON 21
  - CAN bus attacks and trickery
- I enjoy trying to break things…
  - Sometimes I try to fix them ☺
  - Sometimes people don’t listen ☹

Disclaimer

- All affected parties have been notified of identified security issues that are about to be presented
- I am NOT a power grid engineer!
- Don't try this at home (without permission)…



Why Hack a Wind Farm?

- Wind energy
  - Becoming the predominant source of renewable energy
  - 4.7% of electricity generated in the United States in 2015
  - Contribution expected to climb to 20% by 2030
- Increased reliance on wind energy draws increased attention to attackers of all shapes and sizes
- Modern wind farms are operated by computers and networks
- Why should we care?
  - https://www.youtube.com/watch?v=wfzgIxMEo8g
  - Mechanical failures can be influenced by targeting insecure control networks!
- But most importantly…

To Prevent Attackers from Turning these Peaceful Symbiotic Systems…

Into Targets of Ransomware… or

Into Massive Burning Wastelands…

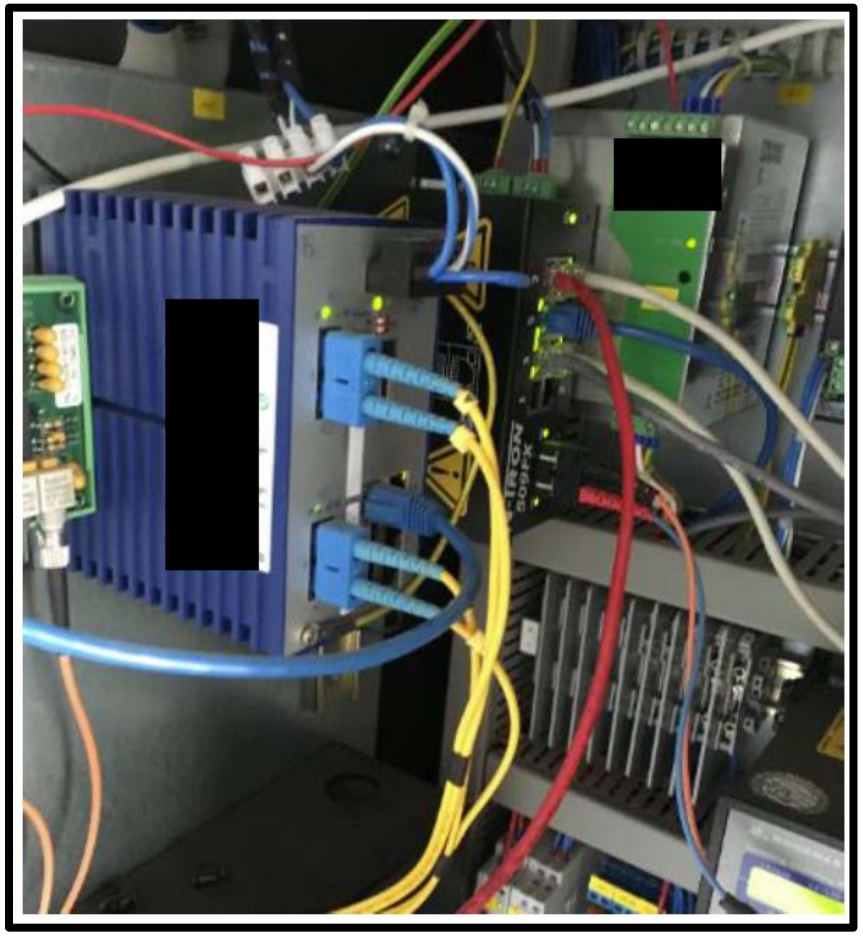
What is a Wind Farm?

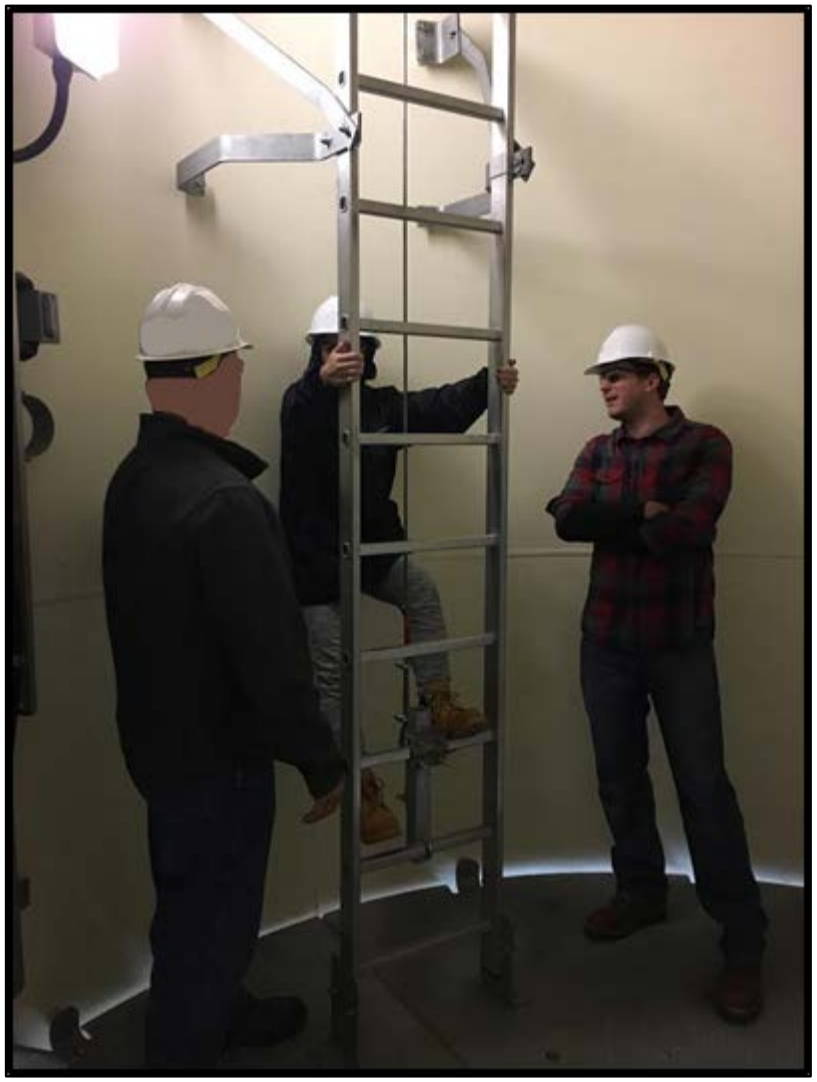
- Power plant that converts wind into electricity
- Wind turbines
  - Variable power source that generates energy from wind
- Substations
  - Collects the energy produced by wind turbines and feeds it into the power grid
- SCADA systems/networks
  - Controls the wind turbines and substations
  - Mix of ICS and IT
- IEC-61400-* specifications
  - Defines design, operations and communications requirements

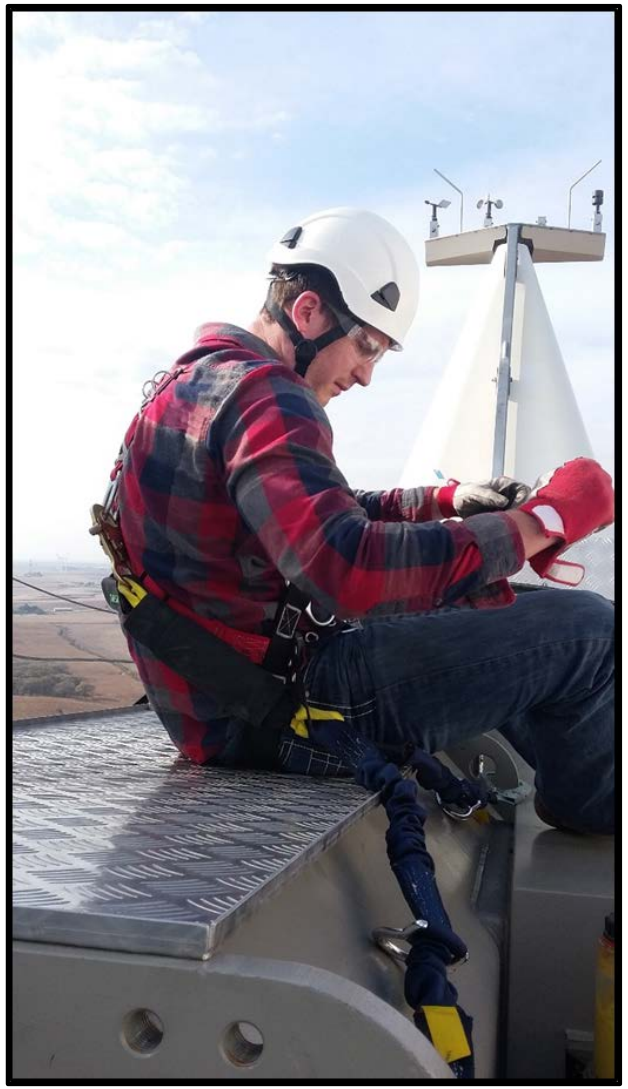
Wind Farm Case Study

- Exclusive access to five U.S. based wind farms
  - 1000+ wind turbines
  - Multiple equipment models from five major vendors
- Research spanning nearly 2 years
- Assessed control systems/networks, IT systems and physical security for vulnerabilities
- Gained valuable insights into how vendors actually implement and configure wind turbine and substation control systems

Red Teaming a Wind Farm

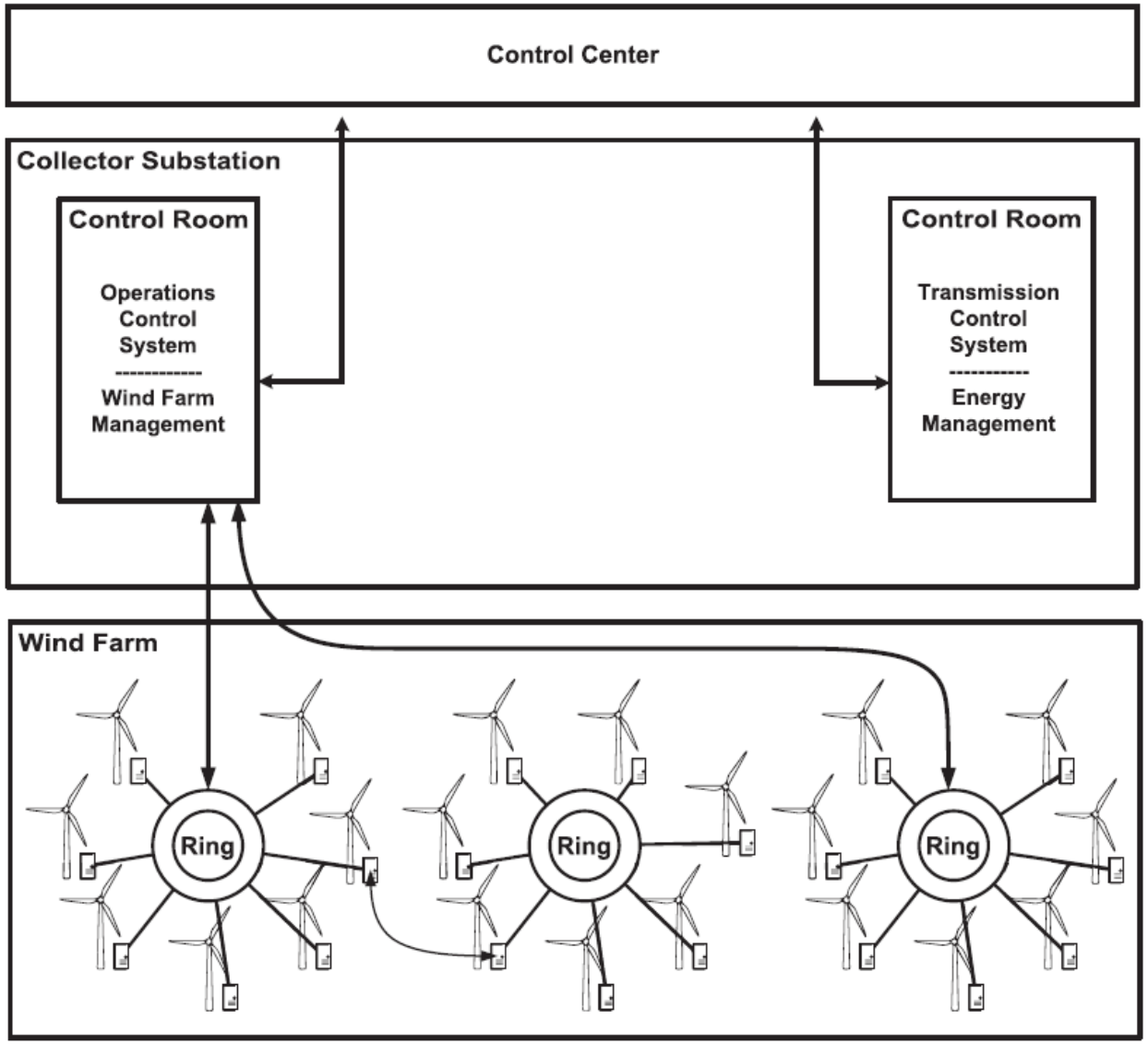




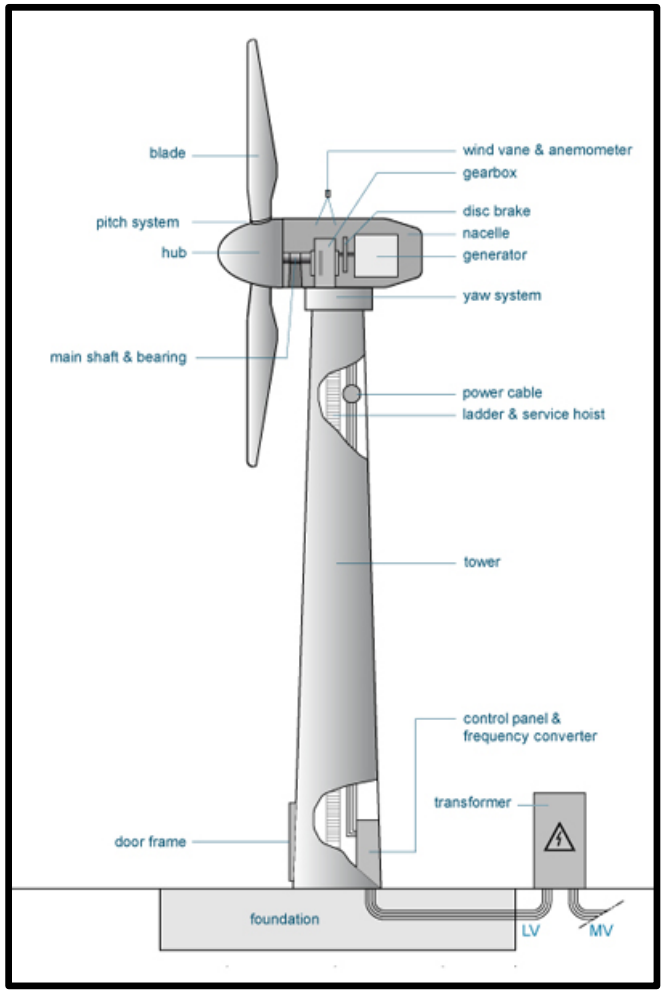


Red Teaming at Over 300 Feet…

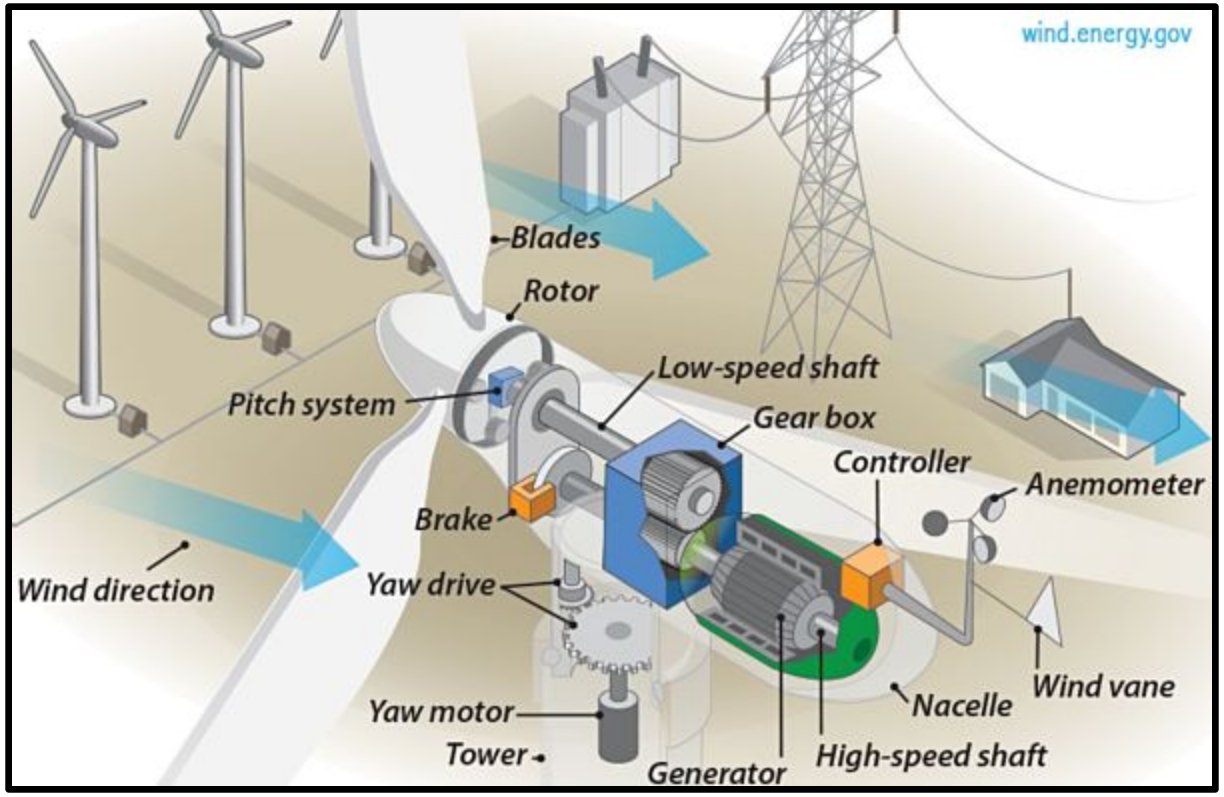
Communications Infrastructure



Wind Turbine Anatomy

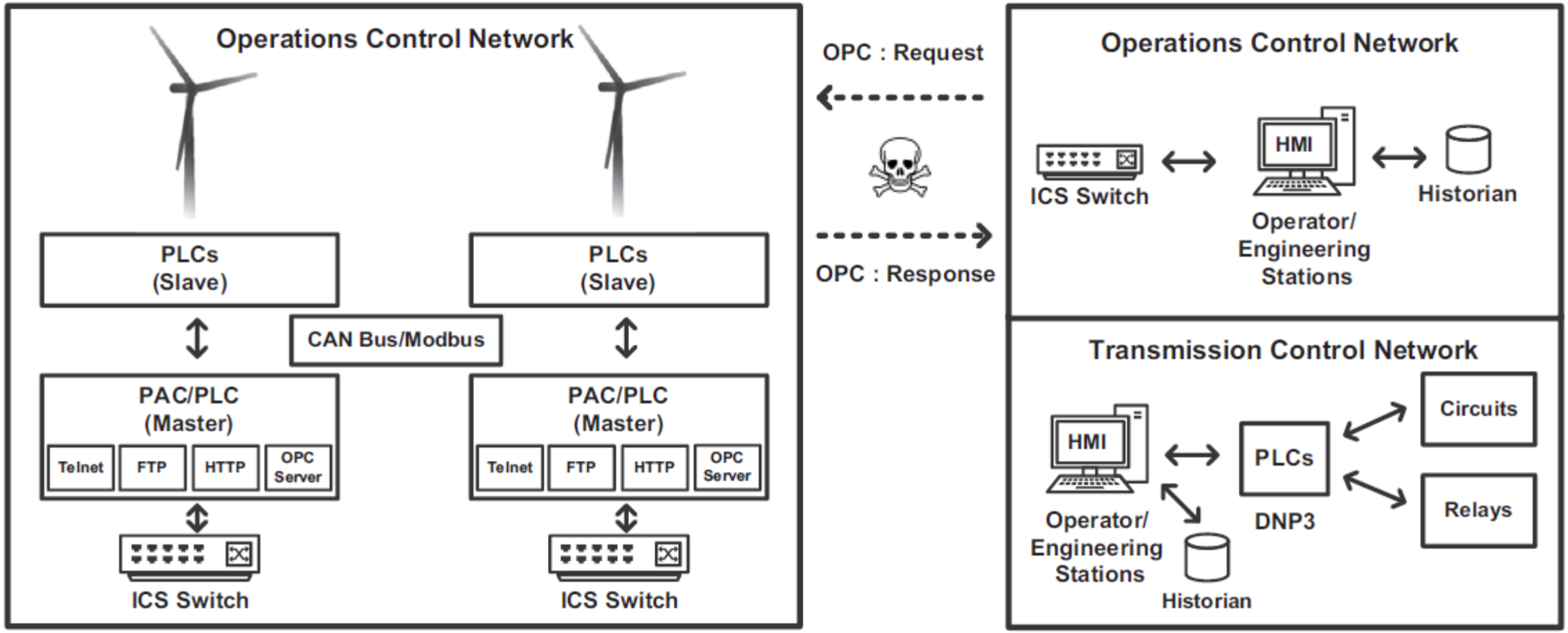


Turbine



Nacelle

Wind Farm Operations Control Network

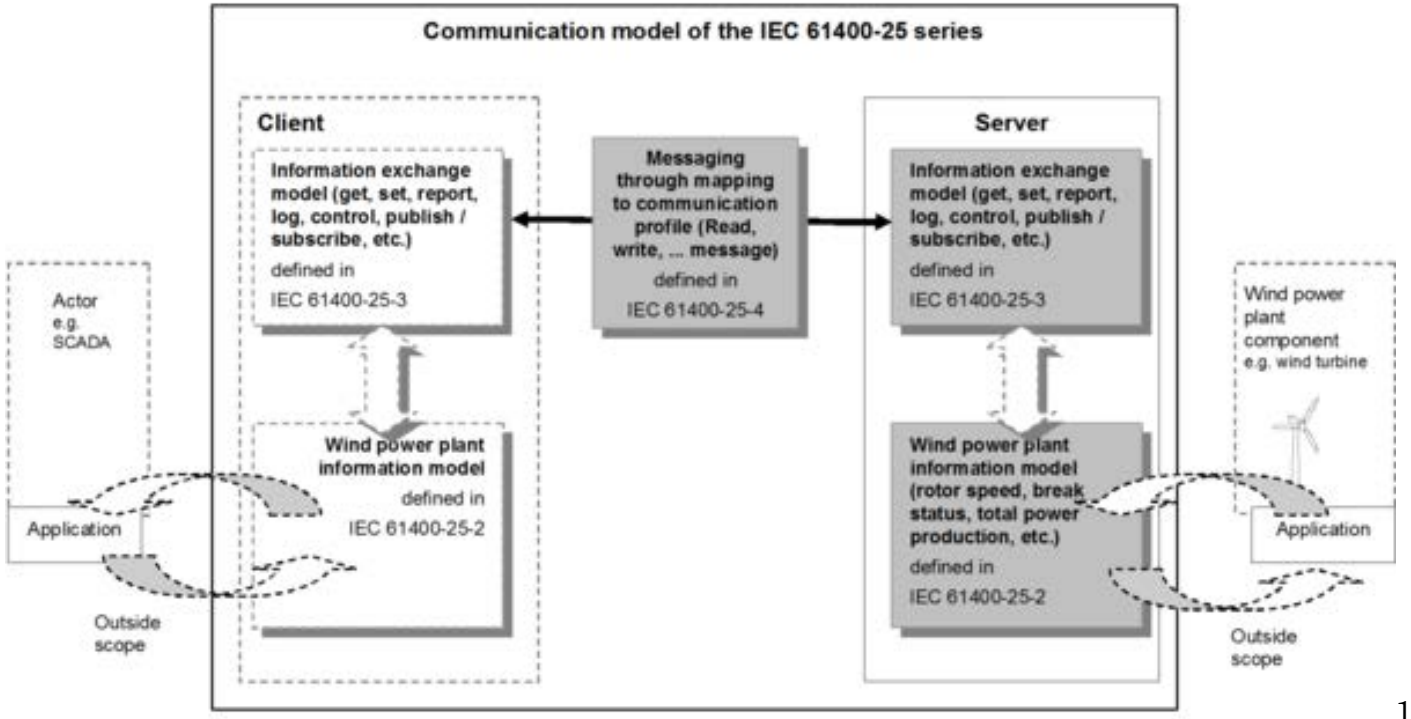


Overview of Vulnerabilities

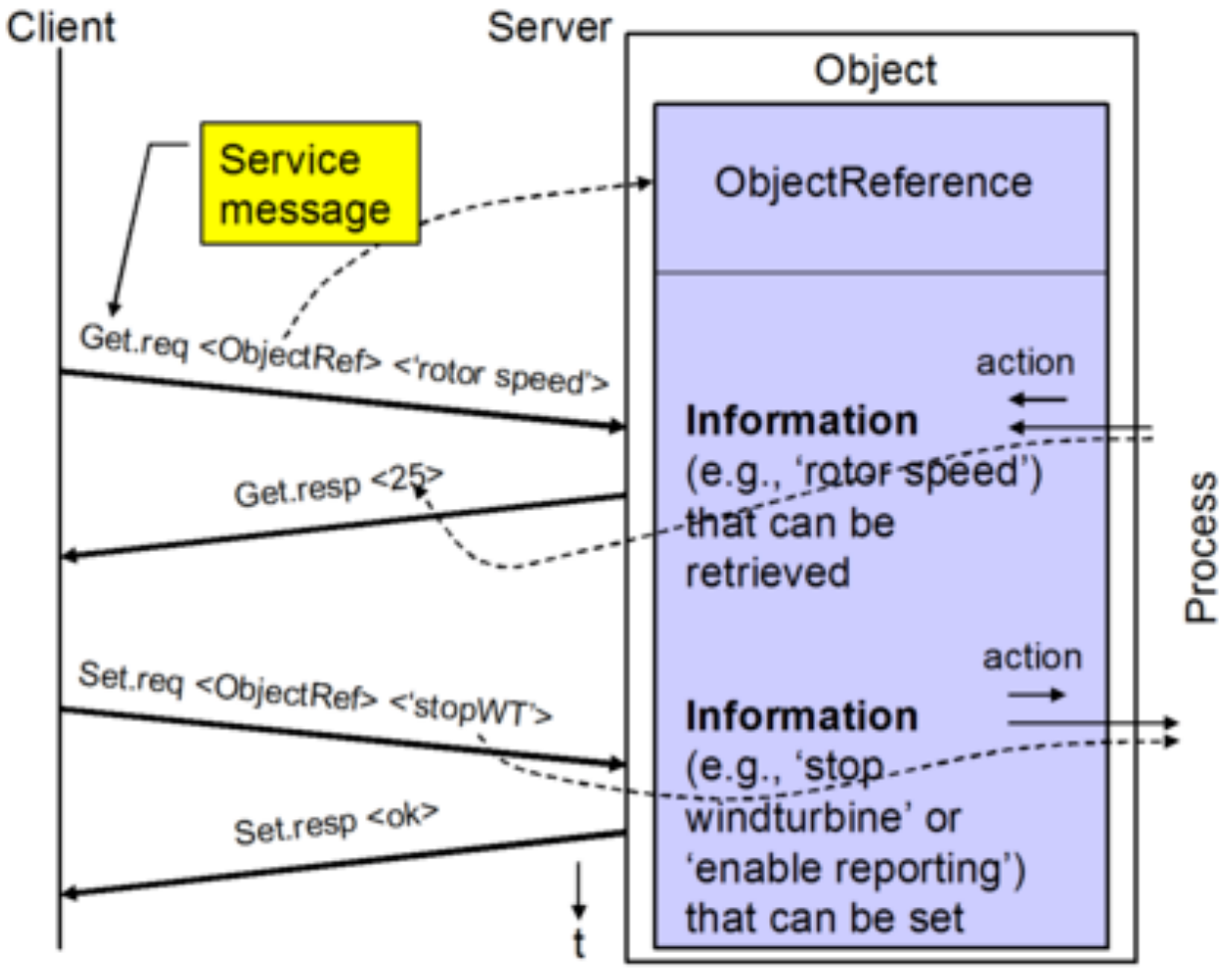
- No authentication or encryption of control messages
- Use of insecure remote management services
  - Telnet, FTP, SNMP, etc.
- Easy to guess or vendor default passwords
- No network segmentation between wind turbines
- Extremely weak physical security
- Exactly what we would expect from an ICS
- And now the fun begins… ☺

IEC-61400-25

- Defines uniform communications requirements for wind power plants
- Support for a handful of protocols
  - Web services
  - DNP3
  - OPC XML-DA
  - IEC 60870-5-104
  - IEC 61850-8-1 MMS



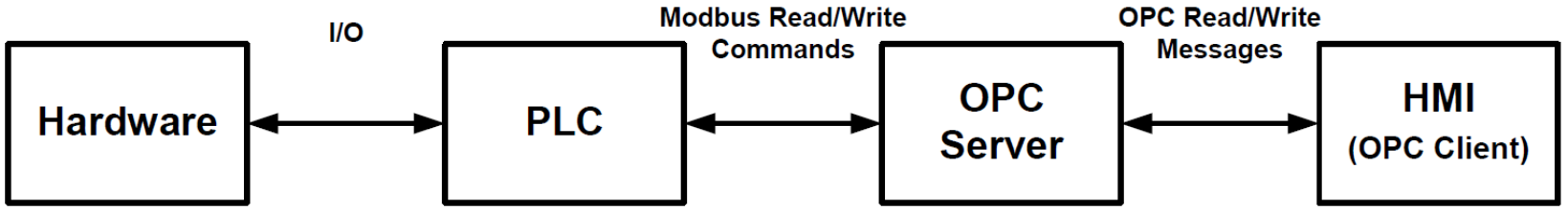
IEC-61400-25



IEC 2150/06

What is OPC?

- First released in 1996 with the goal of abstracting PLC specific protocols into a standardized and generic interface
  - (OPC) Object linking and embedding for Process Control
  - Different variations have been developed over the years
    - OPC UA, OPC XML, OPC XML-DA, etc…
- Used to exchange real-time data, monitoring of alarms/events, and… set/update control values ☺
- Client/server architecture
  - Client = Issues the OPC read/write request message
  - Server = Translates request into appropriate field bus command



OPC XML-DA Specification

- Uses SOAP (HTTP, XML) to exchange data
- OPC XML-DA message services
  - Status
  - Read
  - Write ← ☺
  - Subscription
  - Browse
  - …
  - Get Properties

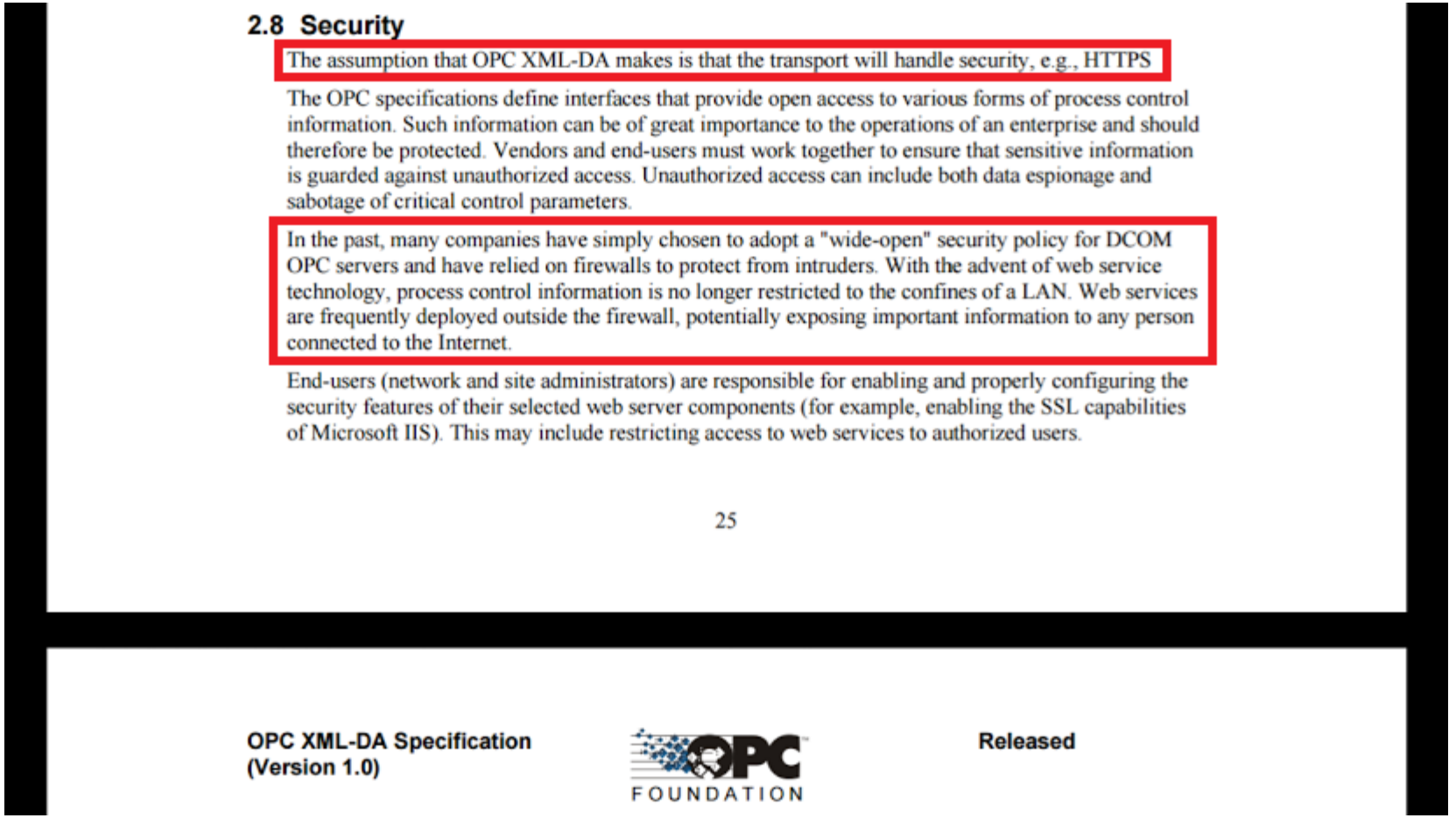
Example:

```
<soap:Body>
  <Write xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <Options
      ReturnErrorText="false"
      ReturnItemName="true"
      LocaleID="en"
    />
    <ItemList>
      <Items ItemName="Simple Types/UInt">
        <Value xsi:type="xsd:unsignedInt">4294967295</Value>
      </Items>
      <Items ItemName="Simple Types/Int">
        <Value xsi:type="xsd:int">2147483647</Value>
      </Items>
      <Items ItemName="Simple Types/Float">
        <Value xsi:type="xsd:float">3.402823E+38</Value>
      </Items>
    </ItemList>
  </Write>
</soap:Body>
```

Vendor Implementation != Specification

- OPC XML-DA messages are sent in the clear by default!
- Technical specification assumes this…
- Sometimes people don't follow instructions ☹
- OPC XML-DA spec. overly reliant on the vendor to tack on additional encryption to secure protocol (e.g., SSL/TLS)
- Fail…

OPC-XML-DA Specification – "security"



2.8 Security

The assumption that OPC XML-DA makes is that the transport will handle security, e.g., HTTPS

The OPC specifications define interfaces that provide open access to various forms of process control information. Such information can be of great importance to the operations of an enterprise and should therefore be protected. Vendors and end-users must work together to ensure that sensitive information is guarded against unauthorized access. Unauthorized access can include both data espionage and sabotage of critical control parameters.

In the past, many companies have simply chosen to adopt a "wide-open" security policy for DCOM OPC servers and have relied on firewalls to protect from intruders. With the advent of web service technology, process control information is no longer restricted to the confines of a LAN. Web services are frequently deployed outside the firewall, potentially exposing important information to any person connected to the Internet.

End-users (network and site administrators) are responsible for enabling and properly configuring the security features of their selected web server components (for example, enabling the SSL capabilities of Microsoft IIS). This may include restricting access to web services to authorized users.

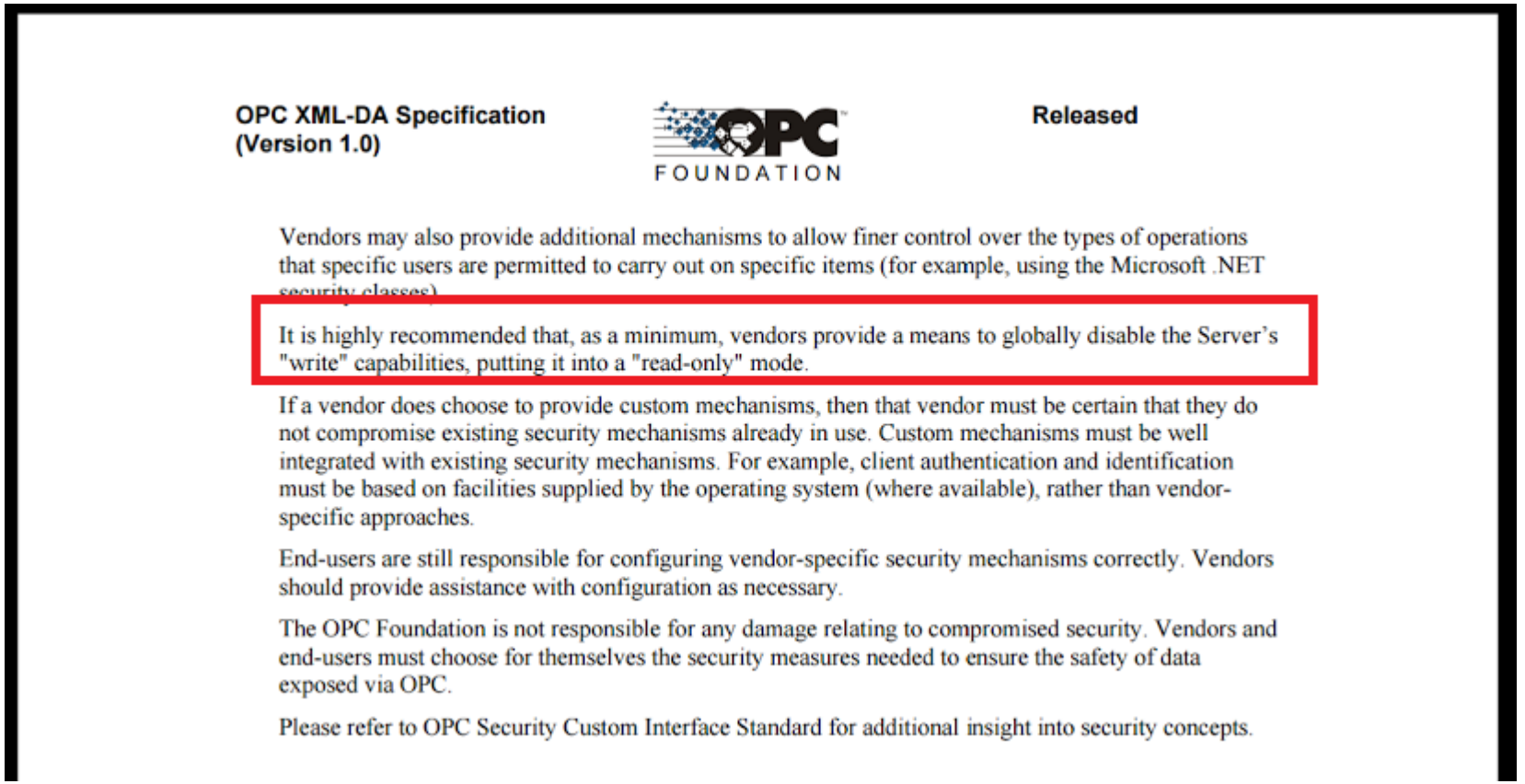
25

OPC XML-DA Specification (Version 1.0)

OPC FOUNDATION

Released

OPC-XML-DA Specification – “disable write”



OPC XML-DA Specification (Version 1.0) OPC FOUNDATION **Released**

Vendors may also provide additional mechanisms to allow finer control over the types of operations that specific users are permitted to carry out on specific items (for example, using the Microsoft .NET security classes).

It is highly recommended that, as a minimum, vendors provide a means to globally disable the Server's "write" capabilities, putting it into a "read-only" mode.

If a vendor does choose to provide custom mechanisms, then that vendor must be certain that they do not compromise existing security mechanisms already in use. Custom mechanisms must be well integrated with existing security mechanisms. For example, client authentication and identification must be based on facilities supplied by the operating system (where available), rather than vendor-specific approaches.

End-users are still responsible for configuring vendor-specific security mechanisms correctly. Vendors should provide assistance with configuration as necessary.

The OPC Foundation is not responsible for any damage relating to compromised security. Vendors and end-users must choose for themselves the security measures needed to ensure the safety of data exposed via OPC.

Please refer to OPC Security Custom Interface Standard for additional insight into security concepts.

Example OPC-XML-DA Read Request Object Items

- Wind speed
- Break status
- Rotor pitch angle
- Power production
- Rotor RPM
- Nacelle direction
- Ambient temperature inside nacelle
- Operating status of CANopen controllers
- Misc. temperatures
  - Oil, rotor, generator

Example OPC-XML-DA Write Request Commands

- Specifics will vary from vendor to vendor…
- Wind turbine operating state (on/off)
- Turbine emergency shutdown (non-graceful)
- Change maximum power generation output (0…x)
- Change nacelle pitch or yaw

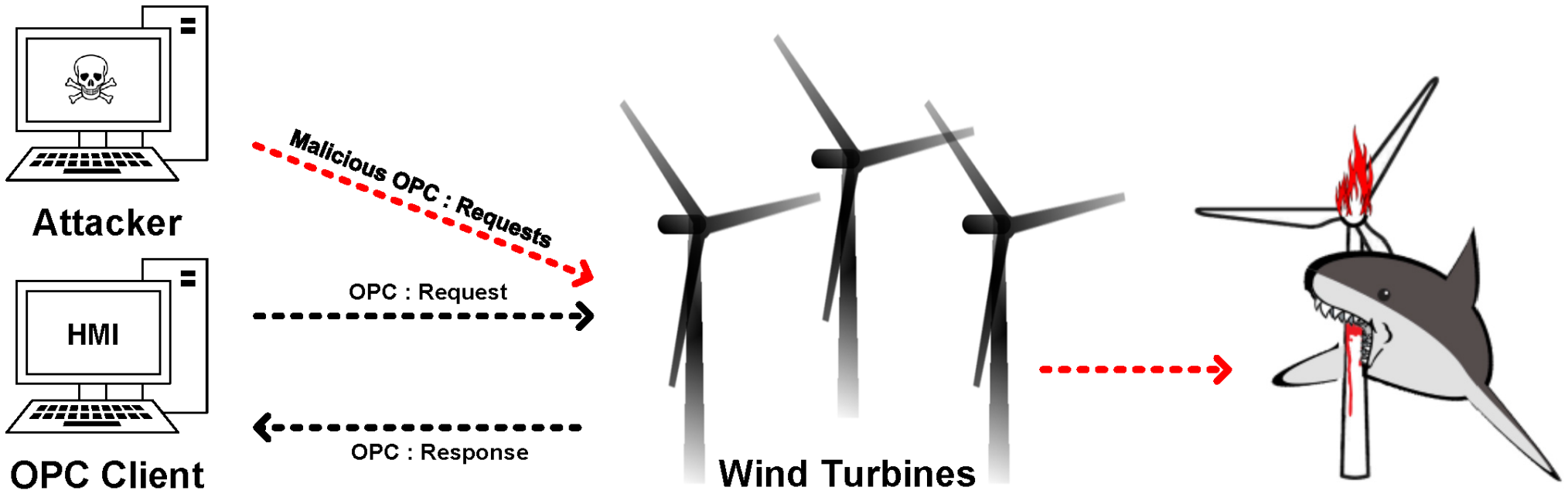
Wind Farm Control Network Access Vectors

- Access can be achieved in a number of ways
  - Physical access to remote turbine/substation in the middle of a field
    - Physical security mechanisms can be easily defeated with lock picks or bolt cutters
  - Compromised vendor network
  - Compromised supply chain
  - Etc.
- Attach rogue device to ICS switch inside the turbine
  - Fiber/Ethernet
  - Raspberry Pi with cellular or Wi-Fi module for remote out-of-band access
  - Boom. You're in.

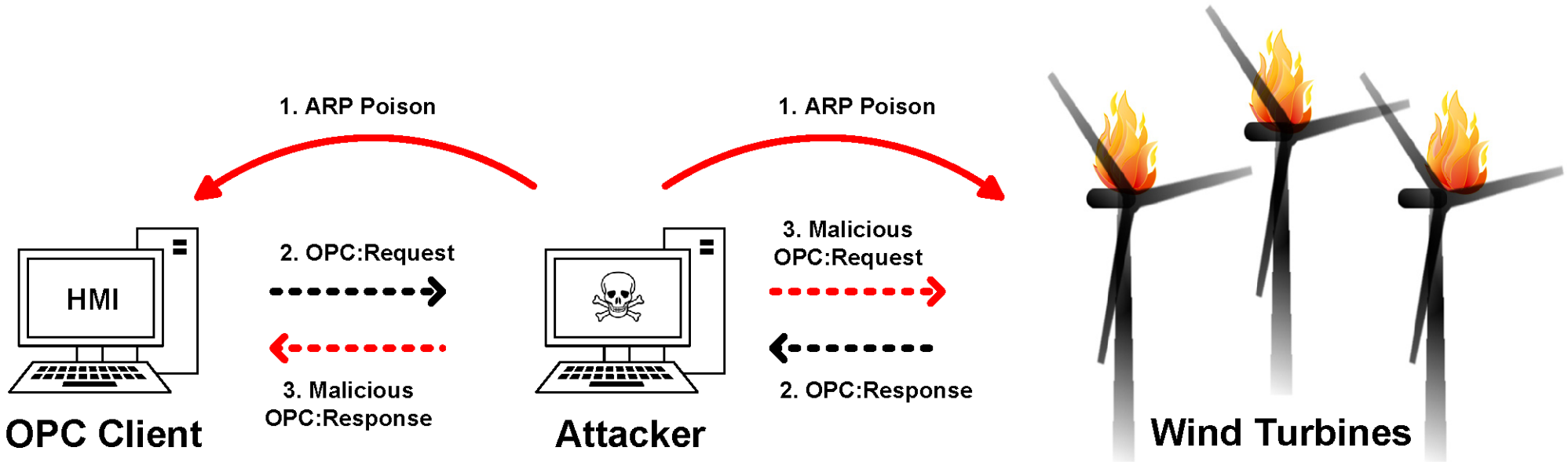
Building Blocks for Wind Farm Security Assessment Tools

- Developed wind farm network security assessment tools
  - Specifically, to attack IEC-61400-25 protocols and network services
- Command-and-control protocol reverse engineering
  - Tcpdump/Wireshark – Static analysis
  - Scapy – Dynamic analysis
- Raspberry Pi 3 (Linux)
  - Python 2.7
  - Bash
  - Scapy – for packet manipulation/fabrication
  - Nmap – for identifying remote OPC servers (running inside of wind turbines)
  - Iptables – for dropping/forwarding packets
- Wind* suite of tools

Windshark


Attacker
HMI
OPC Client
Malicious OPC : Requests
OPC : Request
OPC : Response
Wind Turbines

Windpoison


1. ARP Poison
1. ARP Poison
HMI
2. OPC:Request
3. Malicious
OPC:Response
3. Malicious
OPC:Request
2. OPC:Response
OPC Client
Attacker
Wind Turbines

Wind Farm Attacks!

- Hijack control of wind turbine(s)
  - Scan for OPC servers
  - Fabricate/replay OPC XML-DA write messages using Windshark
- Damage wind turbine(s)
  - Scan for OPC servers
  - Fabricate/replay OPC XML-DA write message using Windshark
    - In a systematic manner (e.g., wear out the braking system, motors, rotors, etc.)
  - Goal: Increase failure rate of key mechanical components (e.g., gearbox)
- Disrupt and/or damage wind farm(s)
  - ARP cache poison all Ethernet nodes on the broadcast domain
  - Intercept, block, modify and fabricate OPC messages
  - Fabricate OPC responses to OPC clients (HMIs) in order to fool SCADA operators

Windworm

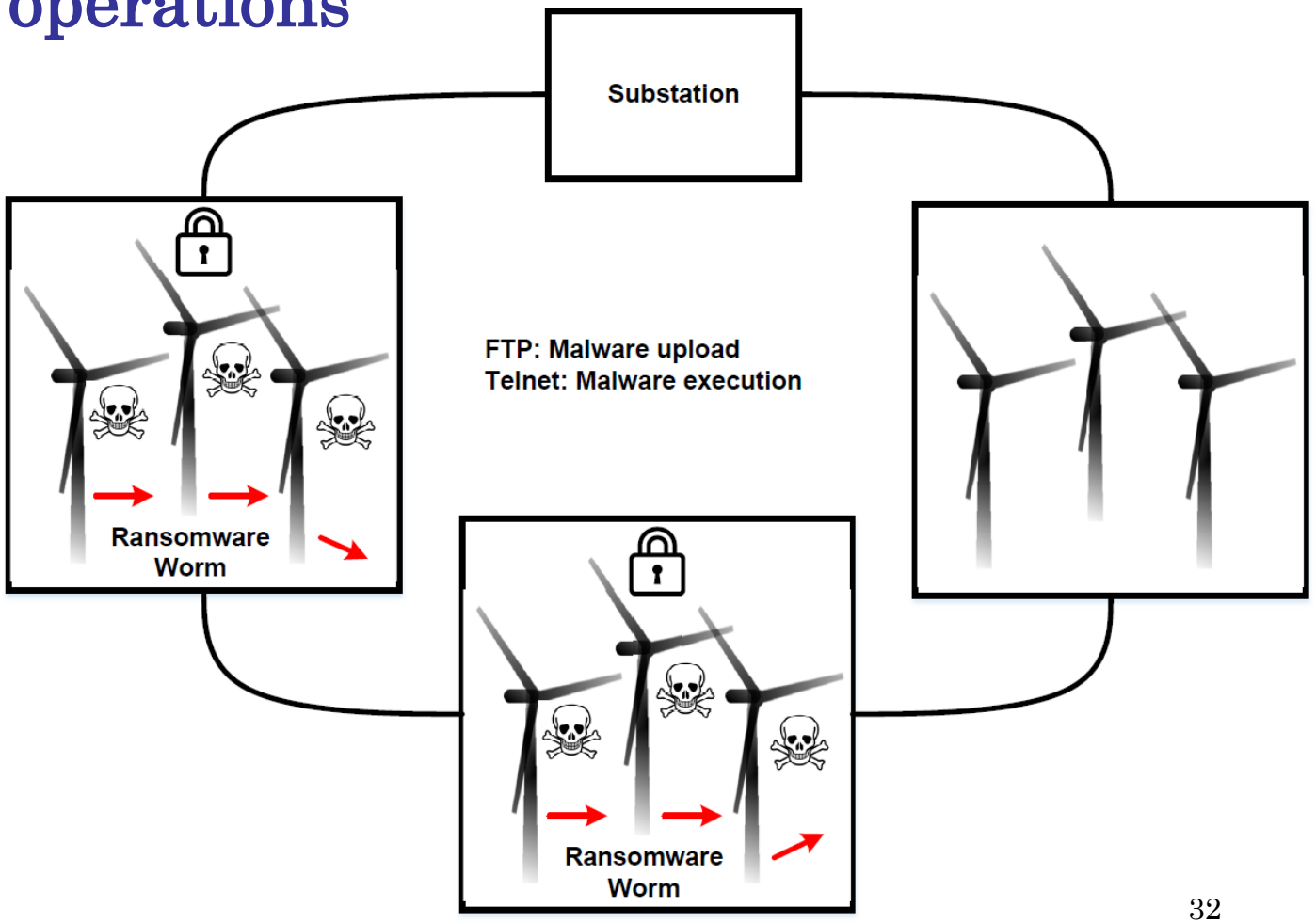
- Targeting programmable automation controllers (PACs)
  - Cross-compile malware for Embedded platforms (e.g., Windows, Linux, RTOS)
- Malware propagation technique
  - Malware upload -> FTP
  - Malware execution -> Telnet
- Leverage root user accounts with default/weak passwords
- Modify critical process control variables
  - Modify/overwrite process values, set points, etc.
  - Initial values usually stored and loaded from a vendor defined configuration file (CANopen Electronic Data Sheet) in the automation controller
- Repeat… pwn wind farm

CANopen

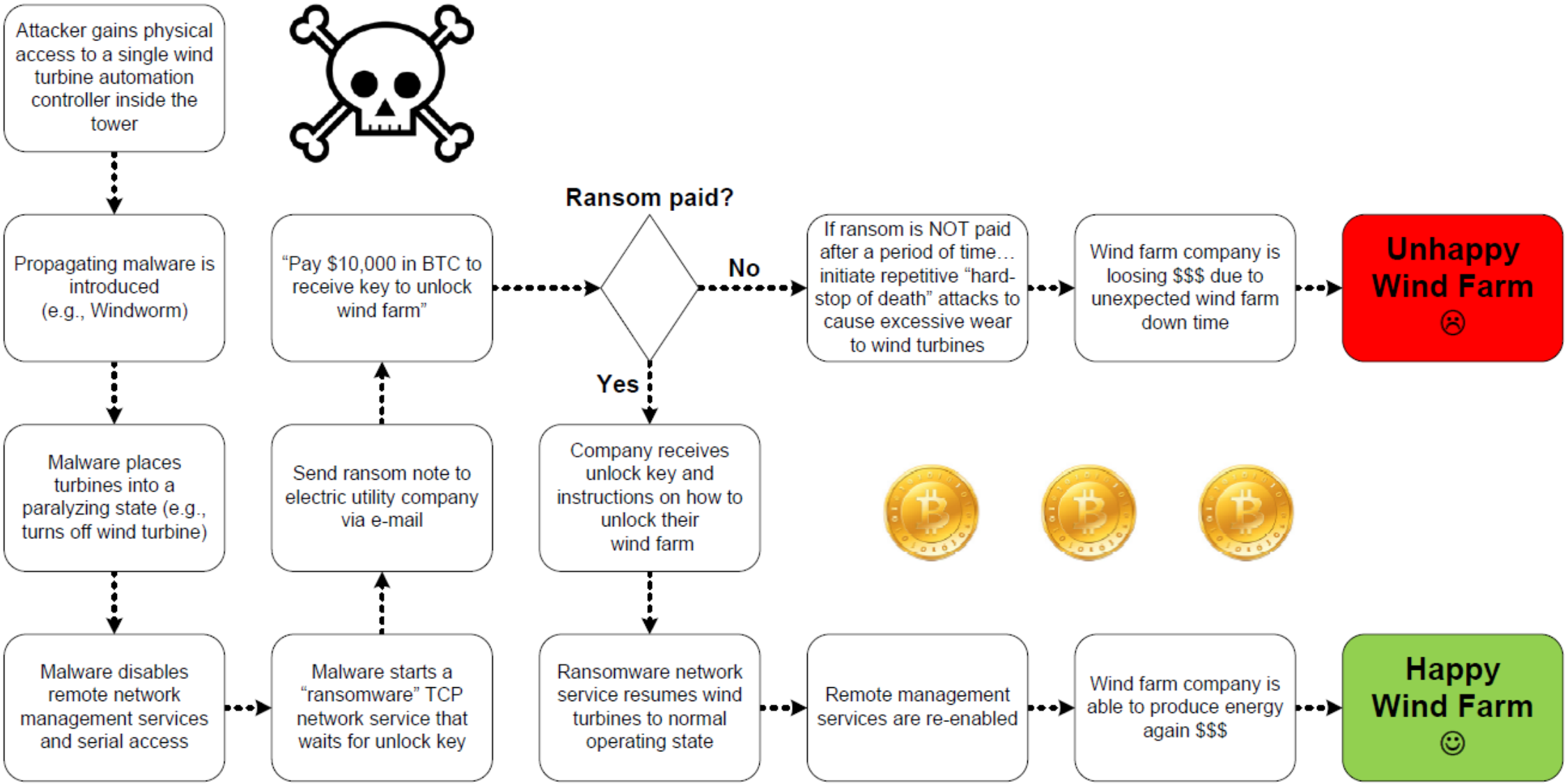
- Commonly used in industrial automation control systems
- Every CANopen node contains a remotely accessible data structure that is used for configuration and communication (object dictionary)
  - 16-bit index, 8-bit subindex
  - Contains wind turbine operational data and operating parameters
- Electronic Data Sheet maps out the entirety of a CANopen node's object dictionary
  - Index, Object name, Data type, R/W permissions, PDO mapping
- CANopen protocols
  - Process Data Object (PDO): Used for sending real-time data between nodes
  - Service Data Object (SDO): Used for setting and reading values from the object dictionary of a remote device

Windransom Scenario

- Goal: Paralyze wind farm operations
- Unless ransom is paid
  - $$$BTC$$$
- How would this work??



How to Ransomware a Wind Farm for Bitcoin?



What is the Potential Impact (lost revenue) Due to Wind Farm Downtime?

- What is the financial cost to the wind farm energy company?
  - Assume 100% dependence on wind energy (no other renewable sources)
  - Assume a 35% capacity factor (worst case)
    - 250 MW × 365 days × 24 hours × 35% = 766.5 GWh = 766,500 MWh = 766,500,000 kWh
- Ransomware infected wind farm example
  - 250 MW (max capacity)
  - 167 x 1.5 MW wind turbines
  - @ $0.12 cents/kWh
  - @ $120/MWh

Downtime (hours)	Cumulative cost of wind farm downtime
1	~ $10,500 (35% capacity) - $30,000 (max capacity)
8	~ $84,000 - $240,000
24 (one day)	~ $252,000 - $720,000
48 (two days)	~ $504,000 - $1,440,000
72 (three days)	~ $756,000 - $2,160,000
168 (one week)	~ $1,764,000 - $5,040,000
336 (two weeks)	~ $3,528,000 - $10,080,000
672 (one month)	~ $7,056,000 - $20,160,000
2016 (three months)	~ $21,168,000 - $60,480,000

Wind Farm Malware Outbreak Recovery?

- How to recover from a large-scale attack?
  - Different perspectives on this depending on who you are (e.g., operator and vendor)
  - Reimage systems (timely)?
  - Replace hardware (costly and timely)?
- How do you know the infection has been fully remediated?
  - How confident are you that it won't reappear?
- In the mean time, the operator is losing out on the ability to produce energy
  - Which means they're loosing $$$

Key Takeaways and Conclusions

- Wind farm control networks are extremely susceptible to attack
  - This is just the tip of the iceberg
- Be proactive
  - Don't wait on vendors to provide "security"
  - Verify vendor claims on "security"
  - Retrofit security as needed
- Wind turbine network isolation
  - Inline firewalls at each tower
  - Bump-in-the-wire with encrypted VPN tunnels
- A call-to-arms for securing wind farm control networks!

Questions?

- jason-staggs@utulsa.edu