Vulnerability Testing of Industrial Network Devices

Matthew Franz (mfranz@cisco.com)
Critical Infrastructure Assurance Group (CIAG)
http://www.cisco.com/go/ciag

Overview

- Introduction
  - Background
  - Related research & testing
  - Objectives & Methodology
- Summary of testing and analysis
- Conclusions and recommendations

Industrial Network Security: the Big Picture

- **Increasing adoption of Open/COTS/Non-Proprietary technology for critical applications**
- **Increasing adoption of Ethernet, Wireless and TCP/IP coupled with *a disappearing boundary between industrial and enterprise networks***
- **Large numbers of production devices and protocols with weak to non-existent security features**
- **Significant cultural/technical gaps between IT and control engineers regarding Cyber Security**
- **A lot of FUD, but few specific details about vulnerabilities (although that is starting to change)**

Industrial Network Security: Open Issues

- Will industrial devices be subject to the same design, implementation, and configuration vulnerabilities that plague IT products?
- How well do existing security products meet the needs of industrial devices, networks, and protocols?
- What new security technologies are needed to protect industrial networks?
- Do industrial vendors have the infrastructure to handle vulnerability identification and disclosure?

Known vulnerabilities in control system networks

Design	Implementation	Configuration
Insecure comm links Insecure devices & protocols *Less than weak* authentication in devices and protocols Insecure remote access (i.e. modems) Undocumented commands/backdoors	TCP/IP stack issues? Protocol flaws? OS/App flaws? Windows HMI Flaws WEP Flaws Network infrastructure device DoS	802.11 Defaults (no WEP) Weak/default passwords Inadequate filtering on router/firewall OS defaults and failure to apply patches & upgrades
Ill-defined or unrealistic security requirements	***Insecure coding practices and inadequate testing***	***Default insecure features and difficult/non-scalable features***

Related Cisco CIAG Projects

- SCADA Protocol Vulnerability analysis and testing
- Modbus/TCP support for Linux Netfilter

  http://modbusfw.sourceforge.net

- Factory Automation Security

  Plant floor architecture using Cisco hardware and software security solutions (PIX, VPN, ACS, 2955)

- Virtual SCADA HoneyNets

  Simulate multiple devices, protocols, and networks to gather attack data and raise awareness of intrusion points

- Open Source (Java) AGA 12-1 Reference Implementation

  http://scadasafe.sourceforge.net/

A Pen-Test Post (August 2003)

"What concerns me is the number of process control devices that are now offering embedded HTTP servers, connectivity over IP, etc. Given the reported vulnerabilities on Bugtraq, etc, w.r.t embedded IP stacks in devices like JetDirect cards and the like, I would to know how reliable the IP stacks are in those devices."

http://www.securityfocus.org/archive/101/329129/2003-07-10/2003-07-16/2

Is there really a threat?

- I’m not even directly connected to the Internet

  The traditional perimeter is eroding – SOHO/VPN, wireless, dial-up, partner connectivity

  Multiple application entry points – SMB, Email, Web Browser, Web Server

- Worms and viruses only target Windows machines

  Network infrastructure devices aren’t directly targeted either, but end up as “collateral damage”—*what happens to automation devices?*

  Instead of attacking windowsupdate.com or whitehouse.gov write your worm to…

- The bottom line—*security controls must be integrated throughout the network and end-devices must be hardened*

Methodology

- Apply “lessons learned” from network infrastructure devices to devices and networks stacks used in control systems

  PLC Communication Modules

  Internet-enabled IO devices

  Ethernet-enabled Microcontrollers

  Generic purpose RTOS

- Determine the impact of known attacks and identify remedies—both in the network and the implementation

- No vendors or specific devices will be identified, so don’t ask!

Objectives

- Survey vulnerabilities and security features in Ethernet-enabled devices

  Focus on low-hanging fruit—conduct known attacks against TCP/IP protocols and applications

  Avoided automation protocols such as Modbus/TCP, Ethernet/IP, Fieldbus HSE, etc.

- Provide reasonable baselines for what is possible now and provide data to support long-term enhancement (vs. security wish-list)

- Stimulate further testing and research by vendors, users, and researchers

Testing & Analysis

- TCP/UDP Scans
- OS fingerprinting
- TCP Initial Sequence Number Entropy
- TCP Resource Exhaustion
- Applications Information
  - *Discovery* Protocols
  - Embedded Web Servers
- Miscellaneous security features

Port Scans and Stack Fingerprinting

- Overview

  A variety of techniques can be used to identify which applications, protocols are present and to determine the signature of a network device

  Determine which TCP/UDP and IP Protocol (i.e. IGMP) are active

- Results

  From 1-2 UDP to 10-15 ports open

  “Simple” port scans (of 200-300) ports did cause some devices and applications to become unresponsive

  Non-standard protocol behavior prohibited scanning in many cases—lack of UDP port unreachable and SYN-ACK from closed ports

  OS Fingerprinting not as effective as against general purpose OS’s—devices not yet in the database

TCP Resource Exhaustion

- Goal is to exhaust memory or connections via a relatively small number of packets—not link layer saturation
  - SYN Floods are most well known, but attacks can occur at anytime during the 3-way handshake
  - Number of connections and timeouts are critical
- Results (best to worst case)
  - Quick timeout and recovery – impact ended once attack stopped
  - Prevented any new connection attempts to the port under attack or all TCP-based
  - Terminated existing TCP sessions
  - Complete lockup/crash of device (automatic/manual reboot)

TCP Initial Sequence Number Randomness

- Technique

  Send SYN record sequence number in SYN-ACK segment

  Allows a remote blind attacker to hijack or terminate the TCP sessions

- Results

  Fixed/time incremental sequence numbers used on majority devices

  Able to exploit this to terminate sessions

- TCP Sequence number entropy issues were “fixed” in mid/late-1990s in general purpose devices and network devices

Discovery Protocols

- Majority of devices have a proprietary UDP protocol used to initially identify and configure the device

  Windows configuration application sends a single broadcast/multicast packet to subnet

  Device responds with name of vendor, type of device, and firmware version (worst case)

  Password authentication may/may not be required to configure device

- Trivial to craft discovery packets with a sniffer and simple packet-generators
- Some devices used vendor-specific capabilities of an open protocol (such as Modbus/TCP) instead of proprietary discovery protocol – use

Discovery Protocol Hardening

- If protocol is only used for discovery and initial configuration, allow it to be disabled once device reaches configured state
- Disable support for unicast and filter broadcasts on router/switch
- If open protocols are used, you need to filter on message type (i.e. Modbus/TCP function codes)

Embedded HTTP Servers

- The only thing worse than HTTP would be NetBIOS or DCOM ☺
- HTTP Server Identification
  - None
  - Third Party-Toolkit
  - Device Name and Version
- Authentication Options
  - None
  - HTTP Basic Authentication
  - Form Variable (possibly Java/JavaScript)

Embedded HTTP Servers (cont.)

- Information provided to an attacker (assuming no authentication)
  - Device name, type
  - Device process, memory, network information
- Configuration options
  - Only one product had the ability to disable the web server--this should be mandatory

Security Features Currently Available

- Service profiles to disable unnecessary (and potentially vulnerable) applications
- Simple application access control lists by IP address—but not true packet filtering
- Secure terminal administration via SSH

Overall Device Recommendations

- Allow services to be disabled

  For devices that have management interface (telnet, SSH, HTTP) add options to disable services based on customer requirements

  For those that don't, add this feature to the Windows configuration tool

- Add IP-based access controls to end devices
- Conduct basic robustness testing using known attack tools and techniques (i.e. ISIC and Nessus)

Overall Network Recommendations

- Aggressively filter traffic to control systems networks and to Serial-Ethernet gateways

  Block UDP, broadcast/multicast, and high risk ports on perimeter routers and switches

- Use remote access VPN to provide granular authentication and authorization to remediate lack of security features on end devices
- Consider deployment of host based IDS for Microsoft platforms that "cannot be updated"
- Development of custom signatures for network IDS for automation protocols (??)

Conclusions

- High risk that worm/scanning activity could impact communication with Ethernet-enabled industrial devices

  Management ports were relatively easy to kill

  Physical reboot often required to correct problem

  On modular IO devices the core functional (control loop) was usually unaffected

- Some vendors and implementations are "doing the right" thing, so minimum security behavior is possible—no room for excuses

Conclusions (cont.)

- In general, you get what you pay for—both for quality and features
- Most issues were in applications (especially TCP) but complete crash (reboot and loss of IP connectivity) was less common that expected
- Non-compliant devices are harder to find using known scanning techniques, but that doesn’t make them more secure
- Blind TCP sequence number attacks (to reset or hijack) should be possible given the weak ISN generation routines in most stacks

Areas for further research & testing

- Develop more formal set of minimum requirements for robustness and security features in embedded industrial devices (SP-99 WG3?)
- Vulnerability analysis and testing embedded web servers and web applications (whether commercial toolkit or homegrown)
- Comprehensive security analysis and vulnerability testing (both design & implementation) of open automation protocols including system-wide impact of protocol attacks
- Learn from the mistakes of others—while not all “IT Security” solutions apply, the vulnerability assessment

References

- Pothamsetty & Balinsky,"A Structured and Practical Methodology for Security Evaluation of an IP Stack"

  http://www.cisco.com/security_services/ciag/documents/stack-howto.pdf

- Trinux: A Linux Security Toolkit – contains precompiled versions of popular security tools

  http://trinux.sourceforge.net

- Nessus

  http://www.nessus.org

References (cont.)

- **Presentation (and white paper, eventually) available at:**

  http://www.scadasec.net/
  http://www.io.com/~mdfranz/papers/

  http://www.cisco.com/go/ciag/