SIU Southern Illinois University
CARBONDALE

Department of Computer Science
Southern Illinois University Carbondale

CS 491/531
SECURITY IN CYBER-PHYSICAL SYSTEMS

Lecture 12: Hacking ICS

DR. ABDULLAH AYDEGER

LOCATION: ENGINEERING A 409F

EMAIL: AYDEGER@CS.SIU.EDU

Outline

Common Attack Targets

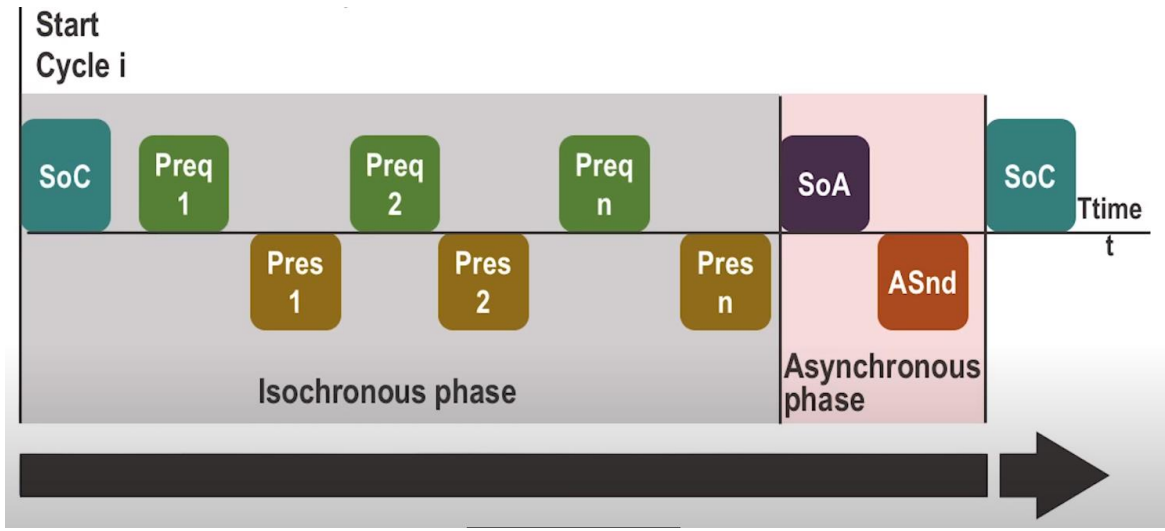
Common Attack Methods

Examples of Incidents

Recall: How Powerlink Works

POWERLINK cycle consists of three phases

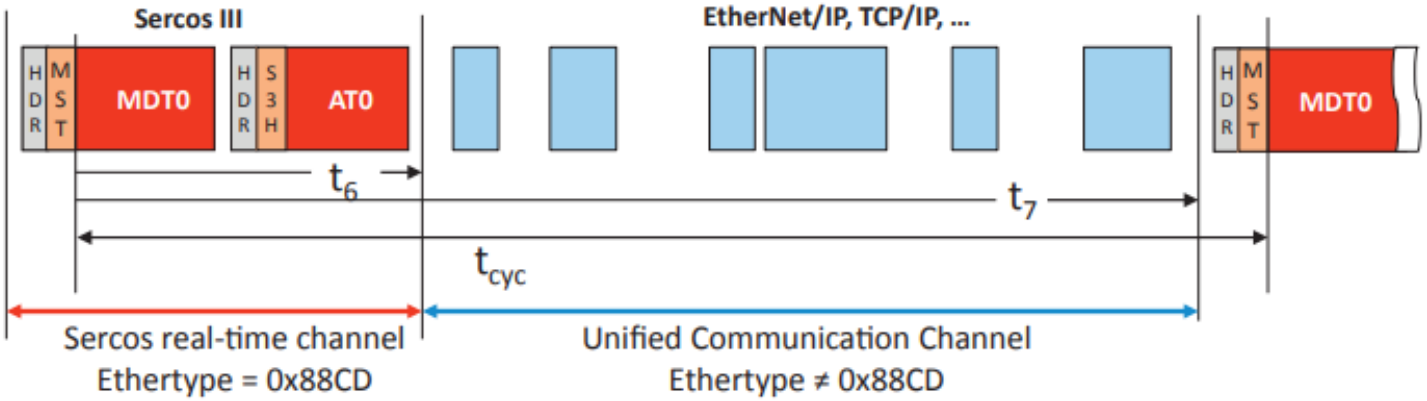
- (1) MN sends a "Start of Cycle" (SoC) frame to all CNs to synchronize the devices
- (2) Payload data is then exchanged or isochronous, phase
  - Slave responses are broadcast, eliminating source address resolution
- (3) The third phase of a cycle is the asynchronous phase, which is where non-time-critical data such as TCP/IP data or parameter configuration data is transferred



Recall: Sercos III: IP Channel

Unallocated time within a cycle to be freed up for other network protocols such as IP

- This "IP Channel" allows the use of broader network applications from the same device—for example, a web-based management interface that would be accessible to business networks



HDR: Header
S3H: Sercos III header

MDT: Master Data Telegram (MDT):
AT: Reply telegram
MST: Master Sync Telegram

t_{cyc}: Cycle time (31.25 µs ... 65 ms)
t_6: Start of the UC channel
t_7: End of the UC channel

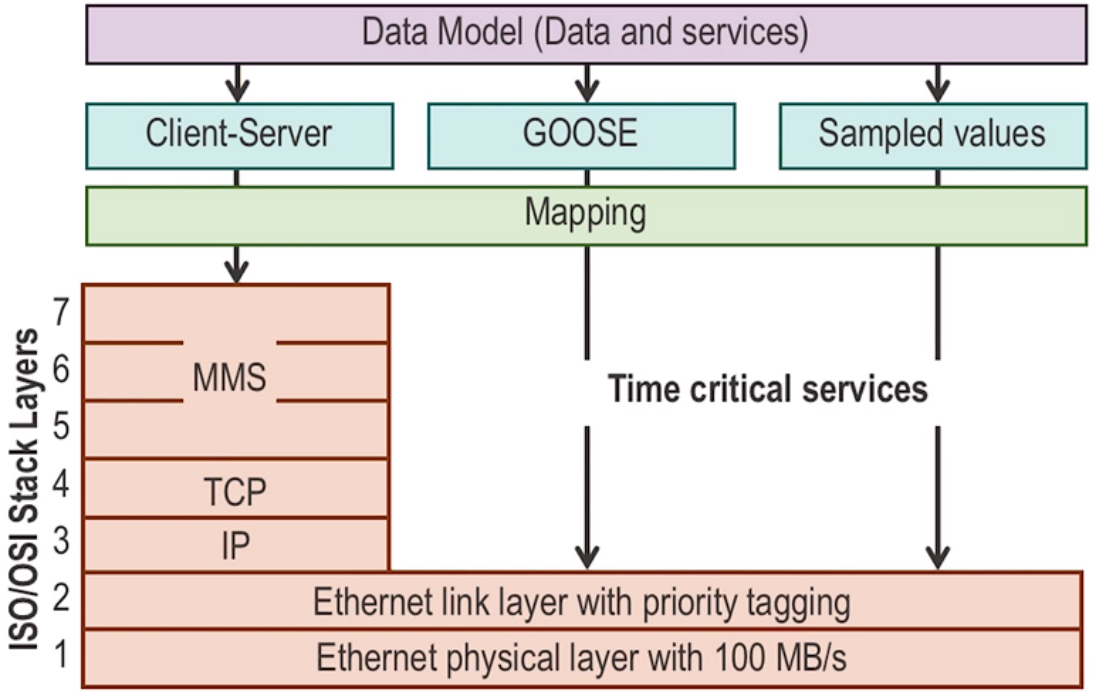
Recall: Protocols supported by IEC 61850

Machine to machine (M2M) or device to device:

- Generic Object Oriented Substation Events (GOOSE)

Client-server:

- MMS (Manufacturing Message Specification)



Consequences of Successful Cyber Incident

Local:

- Impact to quality and business reputation
- Loss of production/intellectual property

Regional:

- Catastrophic equipment failure
- Localized loss of life

Global:

- Generalized panic and widespread loss of life

Safety in ICS

Most ICS employ automated safety mechanisms to avoid catastrophic failures

- Would it solve the problem of critical consequences of cyber incidents?
- Many of these safety control mechanisms utilize same messaging and control protocols used by ICS operational processes
  - Some of the mechanisms are even integrated to protocol itself

Safety systems are very significant

- However, security will not be provided

Common Industrial Targets

Engineering Workstations

SCADA server/historian

Protocols

Examples of Attack Targets

Target	Attack Vectors	Attack Methods	Consequences
Access control system	-Identification cards	-RFID Spoofing	-Unauthorized physical access or access to ICS assets
Data Historian	-Business network client -Database integration communication channel -Remote user access	-Installation of malware via unvalidated software -Database injection -Insecure communication protocols	-Manipulation of process -Credential leakage (business or control) -Unauthorized access to ICS assets
Master or slave devices	-Unvalidated firmware -Weak communication problems -No authentication (or weak) for "write" operations	-Distribution of malicious firmware -Exploitation of INP -Buffer overflow	-Delay system -Mechanical damage -Suppression of critical status/alarms - safety

Examples of Attack Targets

Target	Attack Vectors	Attack Methods	Consequences
Operator workstation (HMI)	-Operational applications -USB -Control network	-Installation of malware via USB -Authorization of ICS HMI functions without sufficient access control mechanisms	-Plant shutdown -Product quality -Credential leak (control)
Telecommunication systems	-Public Key infrastructure -Internet visibility	-Disclosure of private key via external compromise -Exploitation of device connected to public networks -Network access through unmonitored access points	-Credential leak (control) -Information leak -Unauthorized remote access -Command and control
ICS Technician	-Social engineering -Email attachments -File shares	-Transmission of malware on control network via unauthorized connection -Exploitation of applications with administrative rights	-Plant shutdown/delay -Mechanical sabotage -Modification of status messages

Common Attack Methods

MiTM:

- Intercept traffic between two target systems
  - Inject new traffic
- Works only if the connection lacks encryption and authentication
  - Even if auth or encryption is used -> listen for key exchanges and interrupt with your own key
    - Not that simple due to long period of time to re-establish communication
- Most INP authenticate in cleartext
  - Some don't even have authentication

Common Attack Methods

DoS:

- In IT system response is slowed down until DoS is resolved, in Industrial network system shutdown is possible
  - A few examples: loss of communication with device, crashing particular services within device
- Loss of communication may lead to “Loss of Control” or “Loss of View”
  - This will result the system to move “safe state”
    - For instance; oil spill, plant fire and explosion

Common Attack Methods

Replay Attack:

- Requires in-depth knowledge of ICS operations
- Can be considered and can behave as MiTM
- Can alter behavior of entire system (replay on PLC)

https://www.langner.com/wp-content/uploads/2017/03/to-kill-a-centrifuge.pdf

https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf

Common Attack Methods

Compromising HMI (Engineering Work Station):

- Obtain command and control of ICS
- Exploit device vulnerability and install remote access to the console
  - Finding vulnerabilities by penetration testing
- No knowledge of industrial protocols needed (or no ladder logic, etc.)
  - Only interpret GUI to change values within a console

Examples of Weaponized ICS Threats

Aurora Generator Test

- Idaho National Laboratory ran in 2007 to demonstrate how a cyberattack could destroy physical components of the electric grid
- The experiment used a computer program to rapidly open and close a diesel generator's circuit breakers out of phase from the rest of the grid and cause it to explode
- This vulnerability is referred to as the Aurora Vulnerability
  - https://www.youtube.com/watch?v=fJyWngDco3g

Examples of ICS Incidents

STUXNET

- It was the first virus to include code to attack Supervisory Control and Data Acquisition (SCADA) systems (infection started 2007)
  - Poster child of industrial malware
- It is (was at the time of its discovery) the most complicated virus / worm ever discovered
- Average viruses are about 10k bytes in size
  - Stuxnet was 500 KB (and no graphics)
- It is unusual for a virus to contain one zero-day vulnerability. Stuxnet had 4
- Stuxnet also acted like a rootkit – hiding its actions and its presence

What is zero-day vulnerability?

The term "zero-day" originally referred to the number of days since a new piece of software was released to the public, so "zero-day" software was software that had been obtained by hacking into a developer's computer before release

Once a patch is written and used, the exploit is no longer called a zero-day exploit

- These attacks are rarely discovered right away

For zero-day exploits, t1b – t1a ≤ 0 so that the exploit became active before a patch was made available

- t_0: The vulnerability is discovered (by anyone)
- t_{1a}: A security patch is published (e.g., by the software vendor)
- t_{1b}: An exploit becomes active
- t_2: Most vulnerable systems have applied the patch

STUXNET

"As the story goes, the Stuxnet worm was designed and released by a government (the U.S. and Israel are the most common suspects) specifically to attack the nuclear power plant in Iran. How could anyone not report that? It combines computer attacks, nuclear power, spy agencies and a country that's a pariah to much of the world. The only problem with the story is that it's almost entirely speculation." - Bruce Schneier

What we "know" it does:

- Infects windows
- Looks for Siemens SIMATIC WinCC/Step 7 controller software
- Reads and changes bits in the PLC
- Spreads through network/USB
- Various updating mechanisms

How does Stuxnet penetrate a network?

The Stuxnet version discovered in June, 2010 initially spread through flash drives. *.lnk file on flash drive identifies a reference to a file (expected to be an icon). However, no test to verify.

No memory corruption, 100% reliable

Once virus is uploaded and running, it hides the .lnk and source files

Zero Days: https://www.imdb.com/title/tt5446858/

Lessons learned from Stuxnet

Previous Belief	Lesson Learned
Control systems can be isolated from other networks, eliminate risk of cyber incident	They are still subject to human who can use USB
PLC and RTUs don't run modern OS, don't have necessary attack surface	PLCs can be affected and have been affected by malware
Firewall/IDS are sufficient	Blacklisting based defense is not sufficient due to zero-day vulnerabilities, whitelist defenses should be considered against unknown exploits

Adobe Exploits

Example of recent shift in attack paradigm from lower-level protocol and OS to application layer

How this works?

- PDF attached to email from trusted source (spear phishing)
  - Distribution of manuals/reference materials using PDF
- PDF feature of "Launch action" to run executable embedded within PDF
- Available in Kali Linux and Social Engineering Toolkit (SET)

https://github.com/trustedsec/social-engineer-toolkit

How to proceed if infection detected

Not to clean it directly

- May have subsequent levels of infection that exist (staying idle and undetected)
- Valuable info such as infection path, other compromised hosts

First step to isolate the infected host

Collect as much as possible forensics data

- System logs, network traffic, memory analysis data

Sandbox the infected device/system