Vulnerability Assessment in Smart Grids

Jinyuan Stella Sun
UTK
Fall 2016

THE UNIVERSITY of TENNESSEE KNOXVILLE

Contents

- Background
- Roadmap
- Vulnerability Assessment of Phasor Networks
- Defense and countermeasures

Background

- The advent of Smart Grid
  - A class of technologies used to modernize electricity delivery systems, using computer-based remote control and automations
  - Two-way communication and computer processing that has been used for decades in other industries

Background

- The advent of Smart Grid
  - Benefits by incorporating richer data
    - Better interoperability
    - Big improvements in efficiency
      - Electricity delivery system
      - Energy users
    - A more resilient power grid

Background

- Data security is critical
  - Security: control, operation, applications in the smart grid rely on accurate and timely data

Background

- Data security under threat
  - External: hackers, state sponsored cyberwarfare targeting the critical infrastructure.
  - Internal: Disgruntled employee, industrial espionage

Background

News on attacks on decoy SCADA system

We Set Up a Decoy. Hackers Came. From Beijing. And Chattanooga

By Jordan Robertson | Sep 30, 2014 8:29 PM ET | 6 Comments | Email | Print

Three months after online decoys were set up pretending to be industrial-control systems, **we wrote** about how computers from the U.S., China and Russia were found to be the biggest sources for launching scouting attacks against these fake critical infrastructures.

This week, **ThreatStream**, a cyber-security company that set up the target computers at Bloomberg's request, went deeper with the data. Hidden in the larger dataset, which catalogued thousands of reconnaissance probes against our honeypots, was a subset of attacks that

Photographer: Sipa via AP Photo

Bloomberg News 9/30/2014

Background

Attack on U.S. Electrical Grid Could Cost $1 Trillion

ARTICLE | COMMENTS (1)

Email | Print

By BEN DIPIETRO CONNECT

— zhangyang13576997233/Shutterstock.com

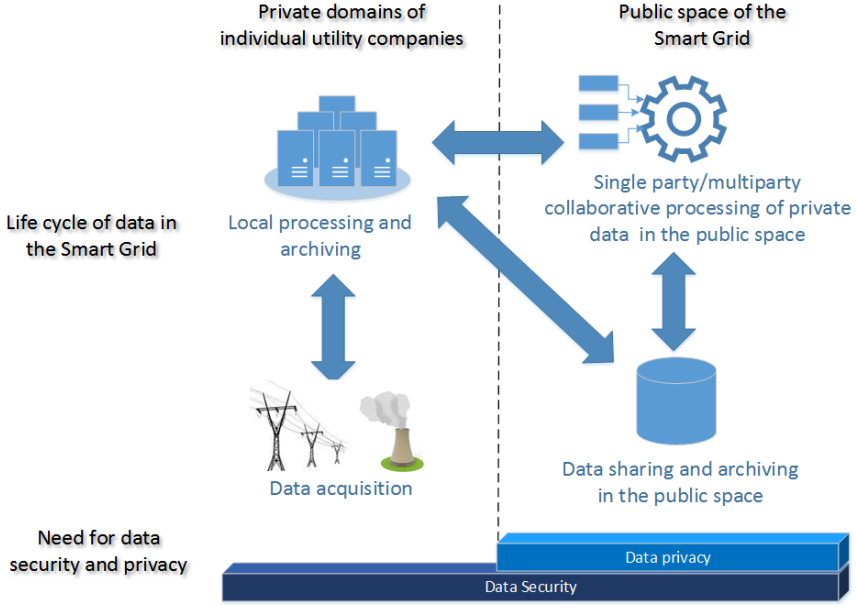
The economic damage to the world economy from a cyberattack on the U.S. power grid would total between $243 billion to more than $1 trillion, depending on the nature and severity of the attack, according to a report released Wednesday.

Background

- Challenges
  - New technologies
  - Larger volume, wider variety
  - More entities involved
    - Multiple data creators (ownership)
    - Multiple data consumers
    - Private data cross multiple trust boundaries

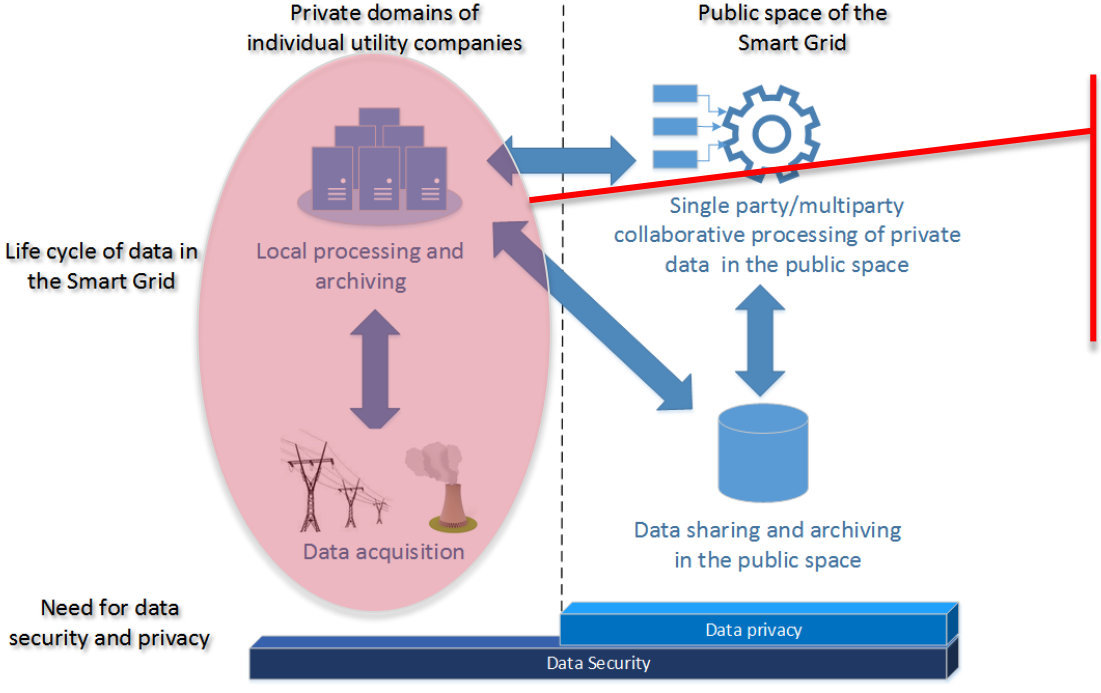
Background

- Data-centric perspective



Roadmap

- Research topics



Vulnerability Assessment of Phasor Network

Phasor Network Applications

- Introduction
  - Phasor network enables many useful phasor data applications
  - Phasor data applications rely on accurate and timely phasor data collected and transferred by the phasor network
  - Vulnerabilities may exist in the standards, protocols, implementations, and configurations of the phasor network technologies.

What is Vulnerability Assessment?

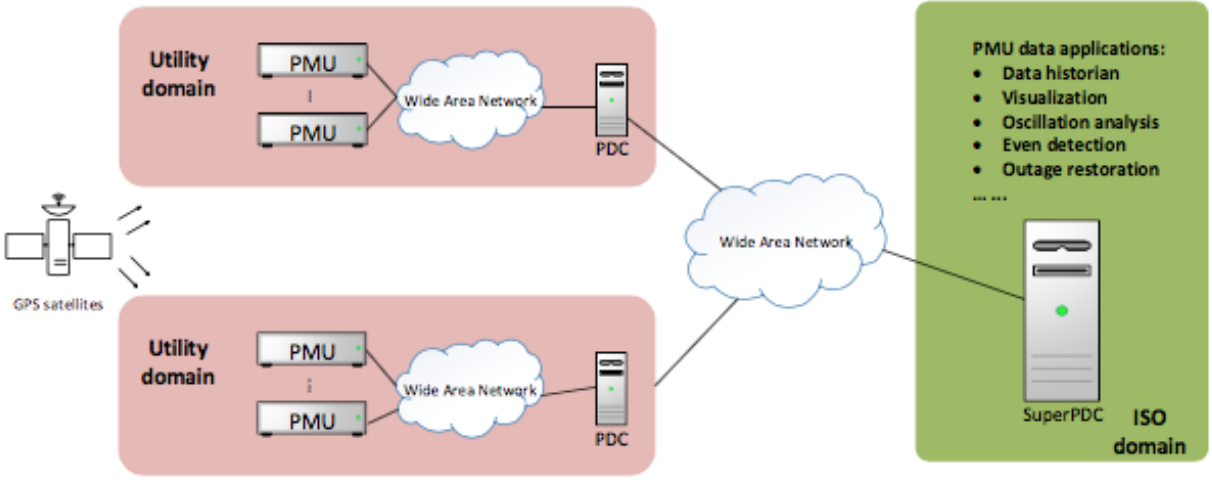
- Vulnerability assessment
  - The process of identifying, quantifying, and prioritizing the vulnerabilities of a system, network, or application.

Vulnerability Assessment in Literature

- State-of-the-art
  - Zhu, Bonnie, Anthony Joseph, and Shankar Sastry. "A taxonomy of cyber attacks on SCADA systems." *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*. IEEE, 2011.
  - Stewart, John, et al. "Synchrophasor Security Practices." (2010).
  - Sridhar, Siddharth, Adam Hahn, and Manimaran Govindarasu. "Cyber–physical system security for the electric power grid." *Proceedings of the IEEE* 100.1 (2012): 210-224.

A Typical Phasor Network

- Preliminary
  - Phasor network

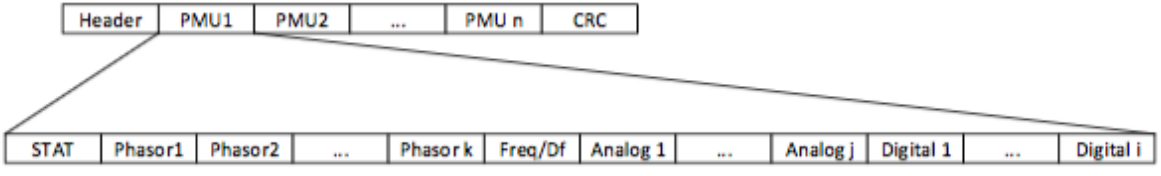


IEEE C37.118 Standard

- Preliminary
  - IEEE C37.118 standard
    - Synchronization to the UTC time
    - Time accuracy
    - Definitions of synchrophasors
    - Criterion for the evaluation of quality of synchrophasor measurements
    - Messaging system
      - Four types of frames
      - A data transfer protocol

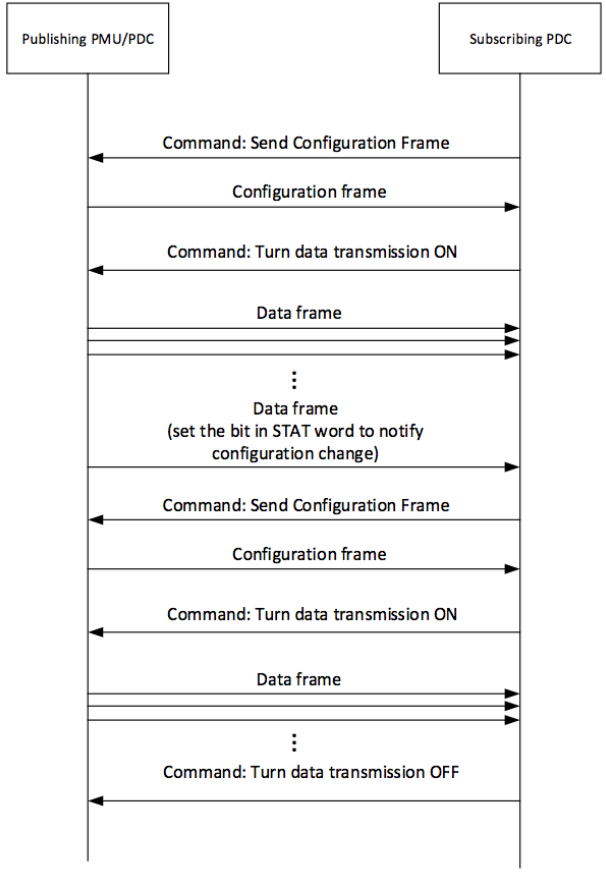
C37.118 Data Format

- Preliminary
  - IEEE C37.118 standard
    - Frames
      - Header frame
      - Configuration frame
      - Command frame
      - Data frame



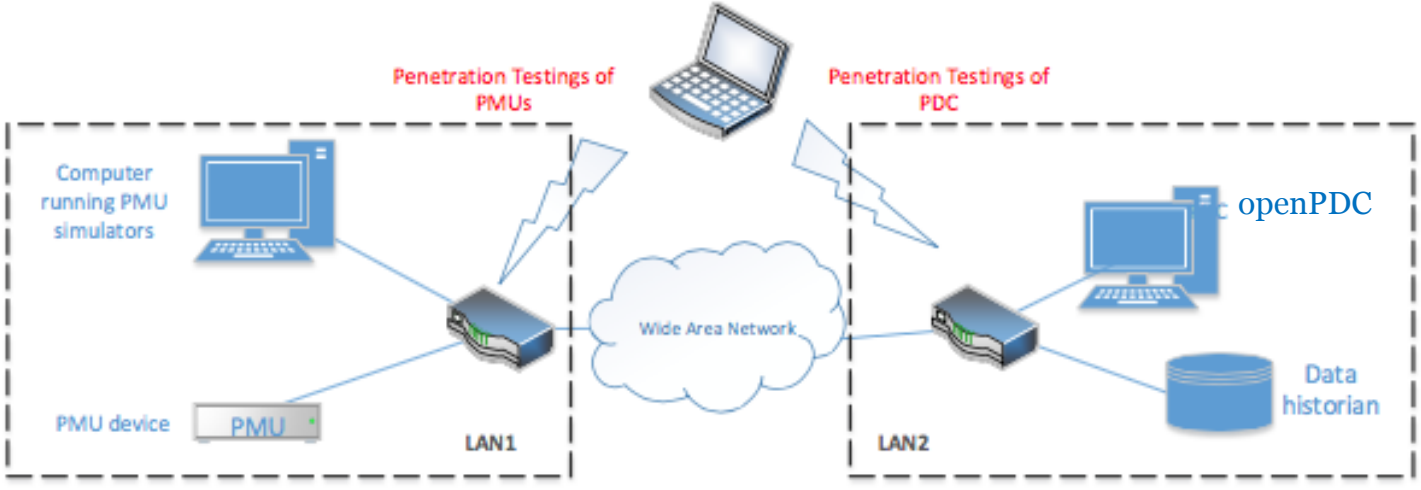
C37.118 Protocol

- Preliminary
  - IEEE C37.118 standard
    - Protocol



Prototype Phasor Network

- Preliminary
  - Small prototype of phasor network



Penetration Testing Procedure

- What is penetration testing?
  - Using the discovered vulnerabilities to exploit a system, network, or application
- We followed the procedure of penetration testing
  - Formally, it is defined in PTES (penetration testing execution standard)
    - Pre-engagement interactions
    - Intelligence gathering
    - Threat modeling
    - Vulnerability analysis
    - Exploitation
    - Post-exploitation
    - Reporting

Key Steps

- We focus on the key steps
  - Reconnaissance
  - Exploitation
  - Exploit development

    -exploit: an exploit is the means by which an attacker, or pentester, takes advantage of a flaw within a system, an application, or a service. An attacker uses an exploit to attack a system in a way that results in a particular desired outcome that the developer never intended.

Reconnaissance

- Reconnaissance
  - Collect information about the system under test
  - Host discovering, operating system fingerprinting, packet sniffing
  - Social engineering

Exploitation

- Vulnerability Exploitation
  - Validate the possible vulnerabilities
    - Automated
    - Manual

Exploit Development

- Exploit development
  - Develop practical attacks that exploits the vulnerabilities
  - Serve as a proof to convince the asset owner their system is vulnerable
  - Provide mitigation recommendations

Pentesting Techniques/Attacks Used

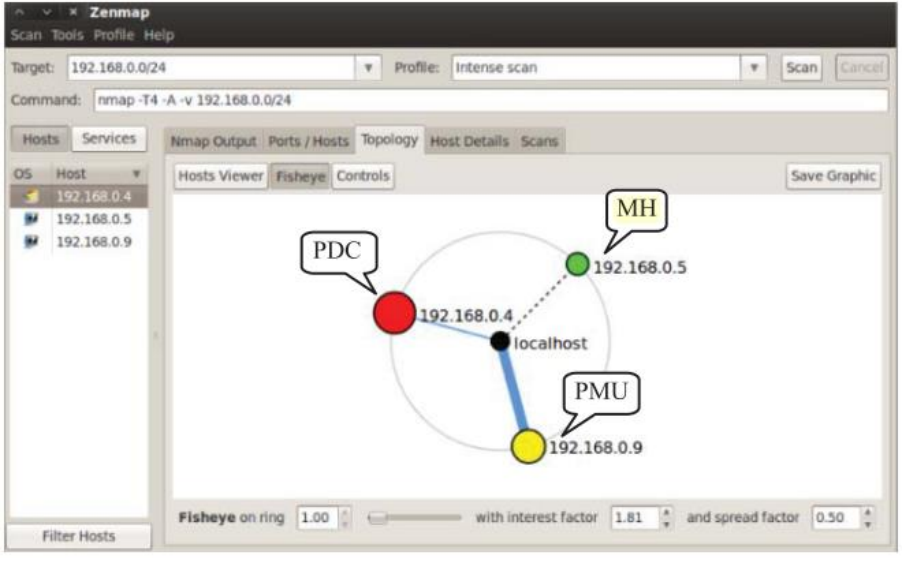
- Packet Sniffing
  - Shared media network: listening to network traffic using NIC under promiscuous mode
  - Switched network: MAC flooding or ARP poisoning to force the network traffic to be forwarded to the sniffer
  - Wireshark
- Packet Injection
  - Send packets to target network service.
  - Packets appear to be legitimate but will interfere with normal execution of the network services or applications.
  - Scapy
- Fuzz testing (Fuzzing)
  - Enumerate all possible inputs (emulate inputs that cross trust boundaries)
  - Test the devices with frames carrying the enumerated inputs
  - Identify inputs that cause the network service to behave abnormally or even crash
  - Scapy

Common Pentesting Tools

- Metasploit
  - Consists of modules: auxiliaries, exploits, payloads
- Kali Linux
  - Contains more than 300 pentesting tools for various use cases (password cracking, wireless attack, …)
- Nmap
  - Network mapper
  - Contains a set of tools: Nmap, Nping, Zenmap

Reconnaissance Result (1)

- Reconnaissance Result – Host discovering



Reconnaissance Result (2)

- Reconnaissance Result – Packet sniffing

```
▶User Datagram Protocol, Src Port: 36835 (36835), Dst Port: 41123 (41123)
▼IEEE C37.118 Synchrophasor Protocol, Configuration Frame 2
 ▶Synchronization word: 0xaa31
  Framesize: 374
  PMU/DC ID number: 1
  SOC time stamp (UTC): 2015-04-09 18:33:18
 ▶Time quality flags
  Fraction of second (raw): 855
 ▼Configuration data, 1 PMU(s) included
    Resolution of fractional second time stamp: 16777215
    Number of PMU blocks included in the frame: 1
  ▼Station #1: "MK208           "
     PMU/DC ID number: 1
   ▶Data format in data frame
     Number of phasors: 3
     Number of analog values: 0
     Number of digital status words: 1
   ▶Phasor names (3)
   ▶Digital status labels (16)
   ▶Phasor conversation factors (3)
   ▶Masks for digital status words (1)
     .... .... .... ...1 = Nominal line freqency: 50Hz
     Configuration change count: 0
    Rate of transmission: 25 frame(s) per second
```

Reconnaissance Result Summary

- Summary of Reconnaissance Result
  - Packets are not encrypted or integrity protected
    - PMU/PDC ID
    - Configuration information of the data frame
    - Possible attacks: eavesdropping, packet modification
  - Lack of user or message authentication mechanisms
    - Possible attacks: packet injection, impersonation
  - Stateful Protocol
    - Possible attacks: Denial-of-Service (DoS)
  - PDC stores and processes external inputs using SQL
    - Possible attacks: SQL injection

Exploitation

- Vulnerability Exploitation
  - Criteria for choosing vulnerabilities
    - Easy to exploit
    - High impact on the data security

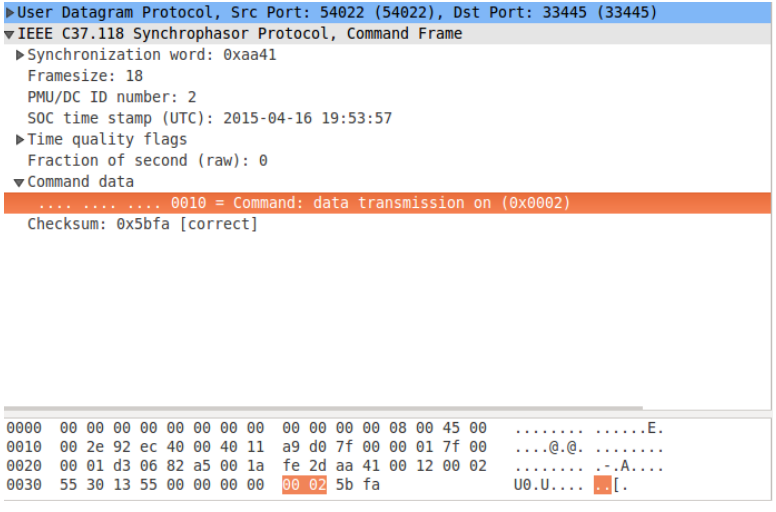
Exploitation Details

- Vulnerability Exploitation

Cause of Vulnerabilities	Possible attacks	Testing technique
Lack of encryption	eavesdropping, replay	Packet sniffing
Lack of user authentication	impersonation man-in-the-middle attack	Packet sniffing Packet injection
Lack of message authentication	frame modification	Packet sniffing Packet injection
Unexpected frames	Denial-of-Service	Packet injection Fuzzing
Lack of input validation	SQL injection code injection	Packet injection

Exploitation Result (1)

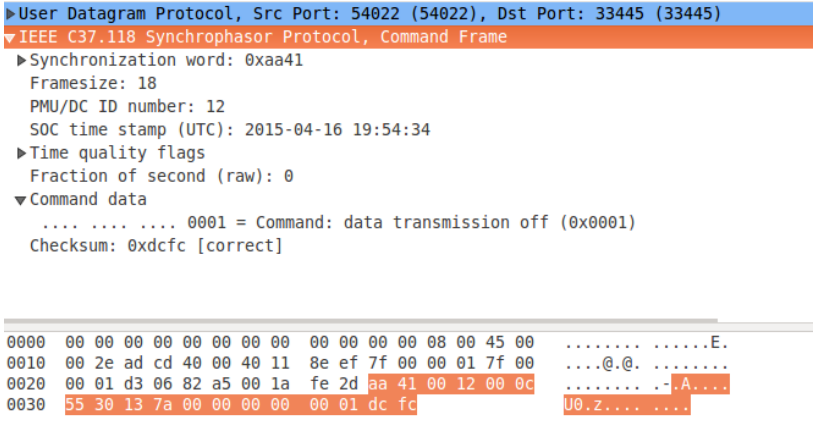
- Vulnerability Exploitation
  - Lack of encryption → Eavesdropping



Captured C37.118 Command Frame: start data transmission

Exploitation Result (2)

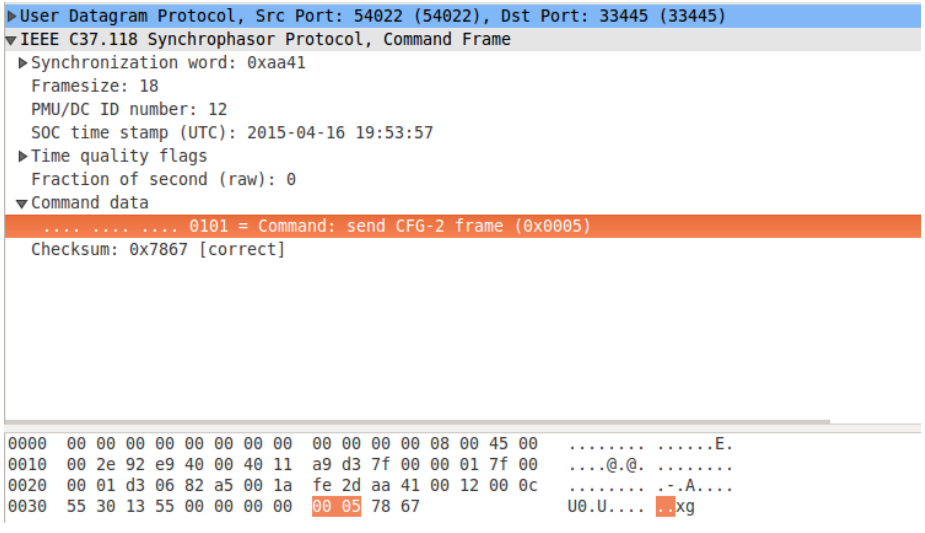
- Vulnerability Exploitation
  - Lack of encryption → Eavesdropping



Captured C37.118 Command Frame: stop data transmission

Exploitation Result (3)

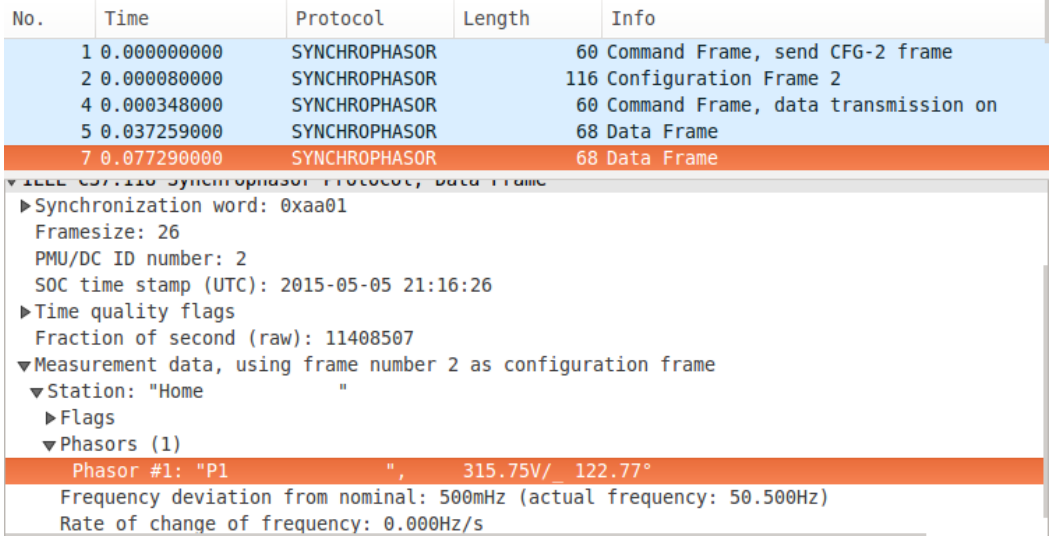
- Vulnerability Exploitation
  - Lack of encryption → Eavesdropping



Captured C37.118 Command Frame: request CONFIG-2

Exploitation Result (4)

- Vulnerability Exploitation
  - Lack of encryption → Eavesdropping



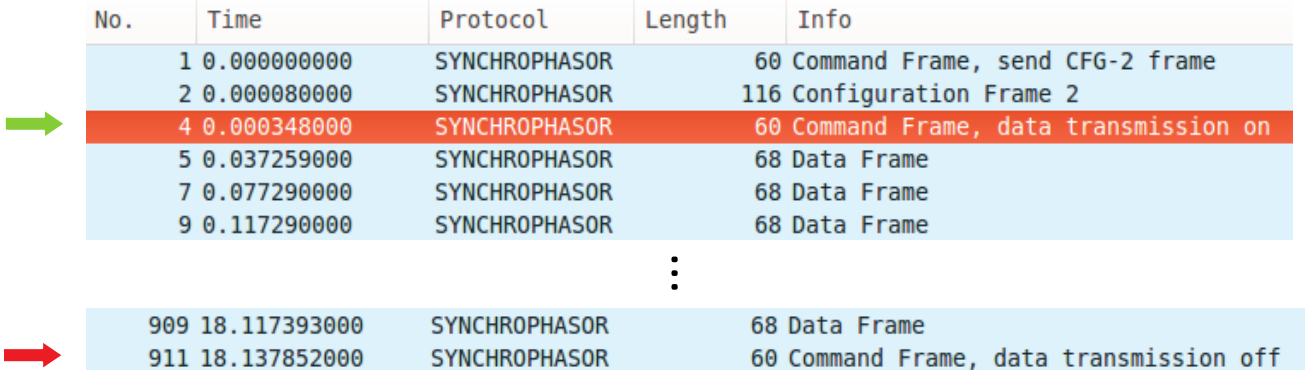
Captured C37.118 Data Frame

Exploitation Result (5)

- Vulnerability Exploitation
  - Lack of user and message authentication → frame spoofing
  - Procedures
    - Capture an authentic frame
    - Duplicate the captured frame but change the bytes that indicate the actual commands, measurements, or configurations to the spoofed values.
    - Change the time stamp of the frame
    - Recalculate the checksum
    - Inject the forged frames

Exploitation Result (6)

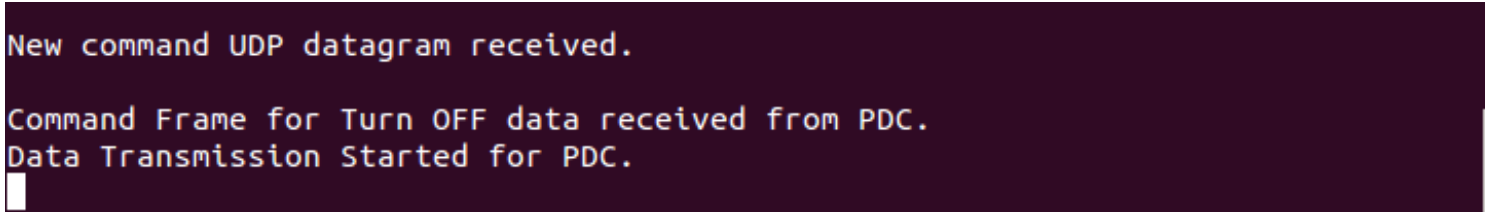
- Vulnerability Exploitation
  - Lack of user and message authentication
    - Command frame spoofing



No.	Time	Protocol	Length	Info
1	0.000000000	SYNCHROPHASOR	60	Command Frame, send CFG-2 frame
2	0.000080000	SYNCHROPHASOR	116	Configuration Frame 2
4	0.000348000	SYNCHROPHASOR	60	Command Frame, data transmission on
5	0.037259000	SYNCHROPHASOR	68	Data Frame
7	0.077290000	SYNCHROPHASOR	68	Data Frame
9	0.117290000	SYNCHROPHASOR	68	Data Frame
⋮				
909	18.117393000	SYNCHROPHASOR	68	Data Frame
911	18.137852000	SYNCHROPHASOR	60	Command Frame, data transmission off

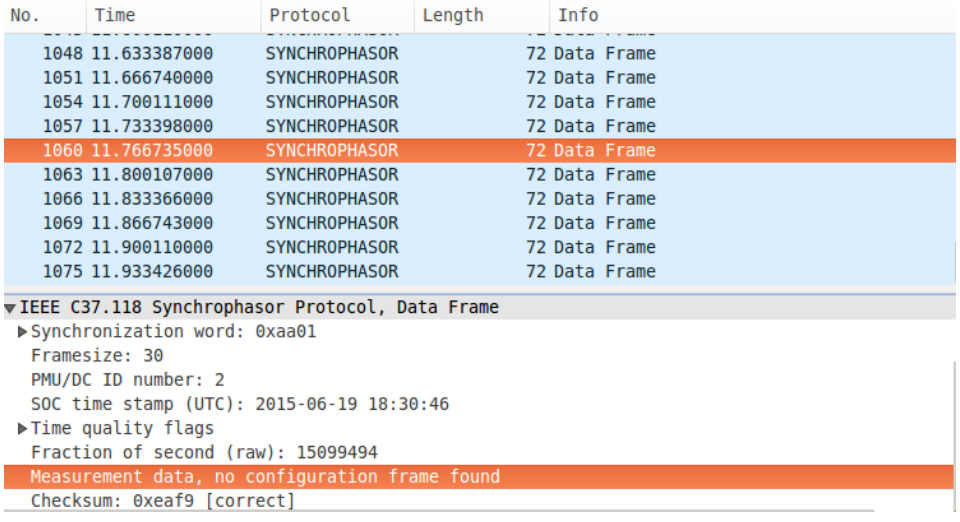
Exploitation Result (7)

- Vulnerability Exploitation
  - Lack of user and message authentication
    - Command frame spoofing



Exploitation Result (8)

- Vulnerability Exploitation
  - Lack of user and message authentication
    - Configuration frame spoofing

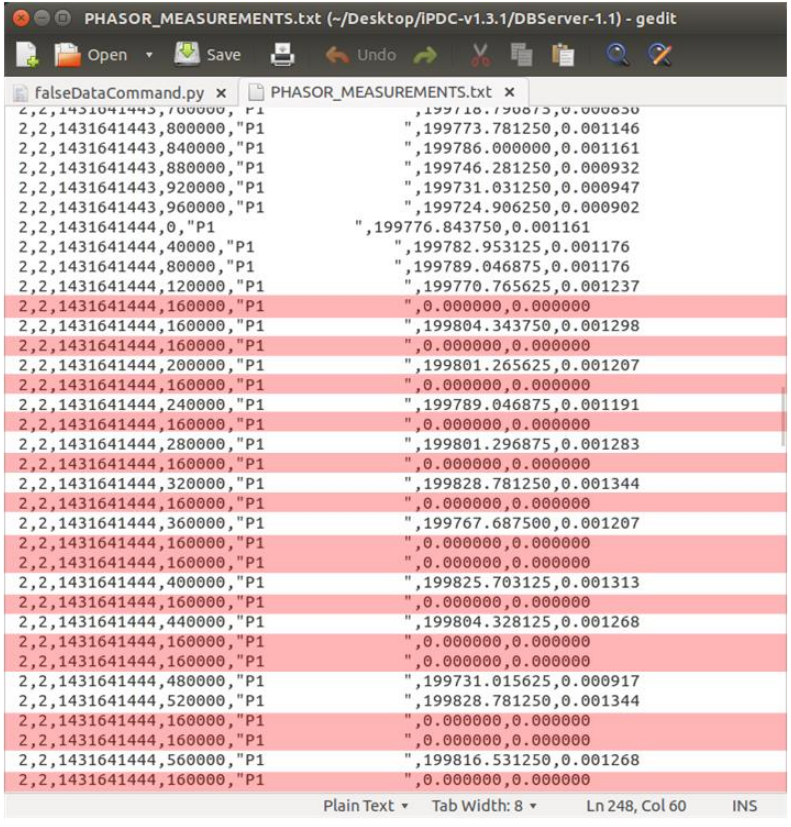


No.	Time	Protocol	Length	Info
1048	11.633387000	SYNCHROPHASOR	72	Data Frame
1051	11.666740000	SYNCHROPHASOR	72	Data Frame
1054	11.700111000	SYNCHROPHASOR	72	Data Frame
1057	11.733398000	SYNCHROPHASOR	72	Data Frame
1060	11.766735000	SYNCHROPHASOR	72	Data Frame
1063	11.800107000	SYNCHROPHASOR	72	Data Frame
1066	11.833366000	SYNCHROPHASOR	72	Data Frame
1069	11.866743000	SYNCHROPHASOR	72	Data Frame
1072	11.900110000	SYNCHROPHASOR	72	Data Frame
1075	11.933426000	SYNCHROPHASOR	72	Data Frame

```
▼IEEE C37.118 Synchrophasor Protocol, Data Frame
 ▶Synchronization word: 0xaa01
  Framesize: 30
  PMU/DC ID number: 2
  SOC time stamp (UTC): 2015-06-19 18:30:46
 ▶Time quality flags
  Fraction of second (raw): 15099494
  Measurement data, no configuration frame found
  Checksum: 0xeaf9 [correct]
```

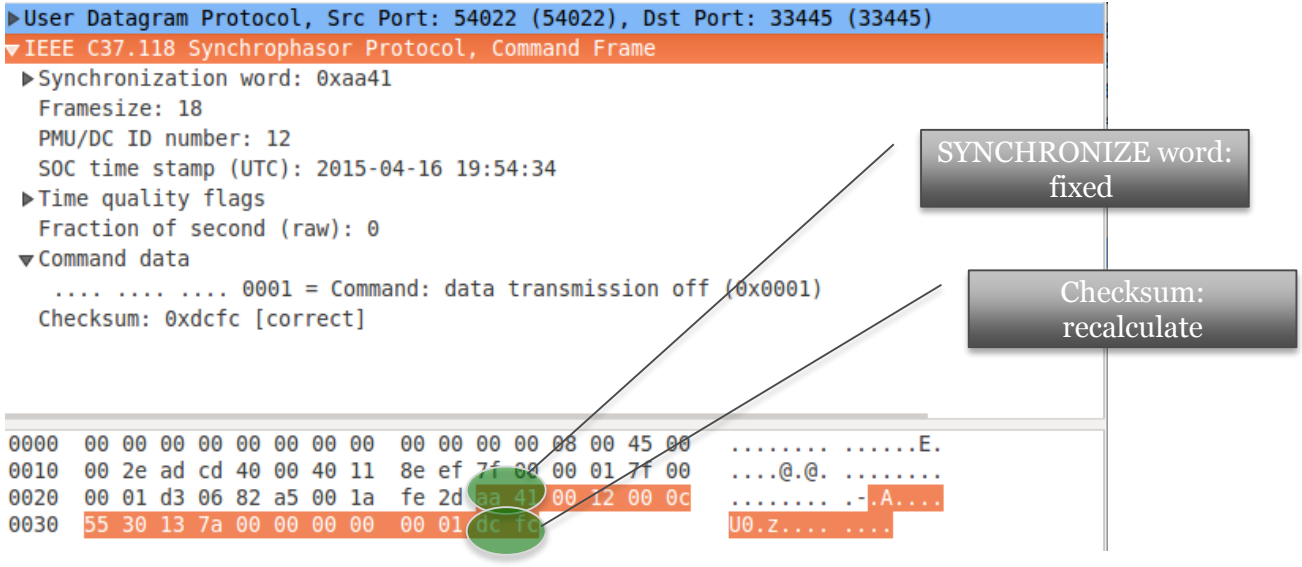
Exploitation Result (9)

- Vulnerability Exploitation
  - Lack of user and message authentication
    - Data frame spoofing



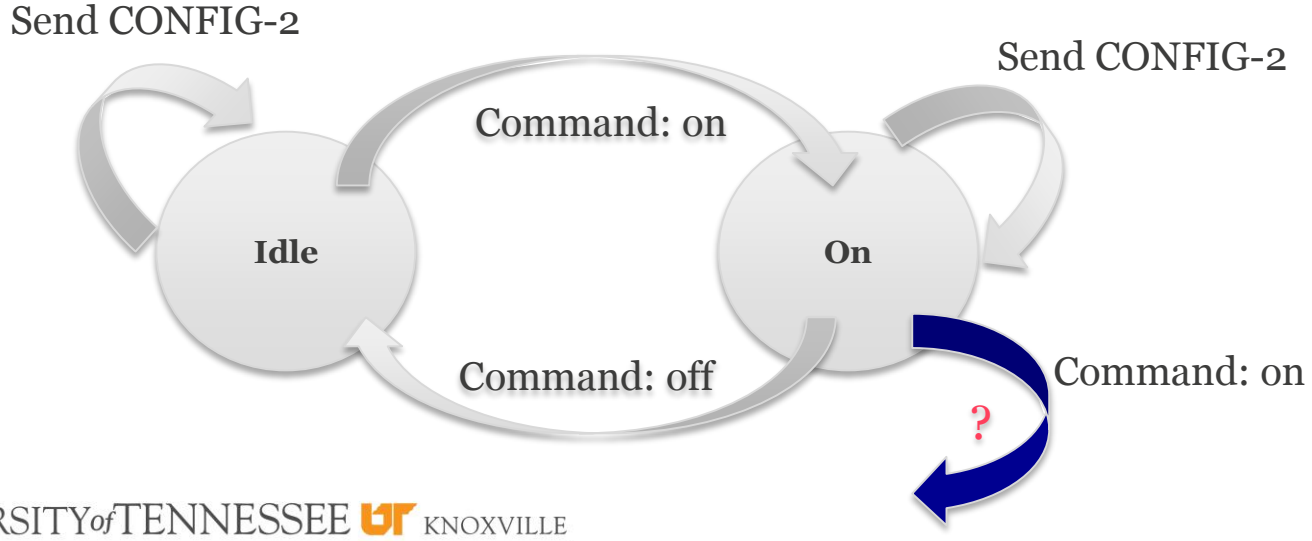
Exploitation Result (10)

- Vulnerability Exploitation
  - Mishandling of unexpected frames
    - To improve the efficiency of fuzzing…



Exploitation Result (11)

- Vulnerability Exploitation
  - Fuzz Testing
    - Command frame fuzzing: PMU simulator became unresponsive after receiving fuzzed command frames that indicate the command "Send CONFIG-2" and duplicate "Turn data transmission on" command frames



Exploitation Result (12)

- Vulnerability Exploitation
  - Fuzz Testing
    - Command frame fuzzing: PMU simulator became unresponsive after receiving fuzzed frames that indicate the command ‘’Send CONFIG-2’’ and duplicate ‘’Turn data transmission on” command frames.
    - Configuration frame fuzzing: passed
    - Data frame fuzzing: passed

Exploitation Result (13)

- Vulnerability Exploitation
  - Lack of input validation (SQL injection)
    - SQL was used to manage the configurations of different registered PMU devices. SELECT * FROM MAIN_CONFIG_TABLE WHERE DEVICE ID = PMU_ID_Number
    - PMU_ID_Number is provided by external input and extracted from the received configuration frame
    - If the PMU_ID_Number is specified as “2; DROP TABLE_MAIN_CONFIG_TABLE”
      - The SQL query becomes:

```
SELECT * FROM MAIN_CONFIG_TABLE WHERE DEVICE ID = 2;
DROP TABLE MAIN_CONFIG_TABLE
```

Exploitation Result (14)

- Vulnerability Exploitation
  - Lack of input validation (SQL injection)
    - Passed SQL injection test
      - Sanitize the input
      - Use parameterized queries with strongly typed parameters

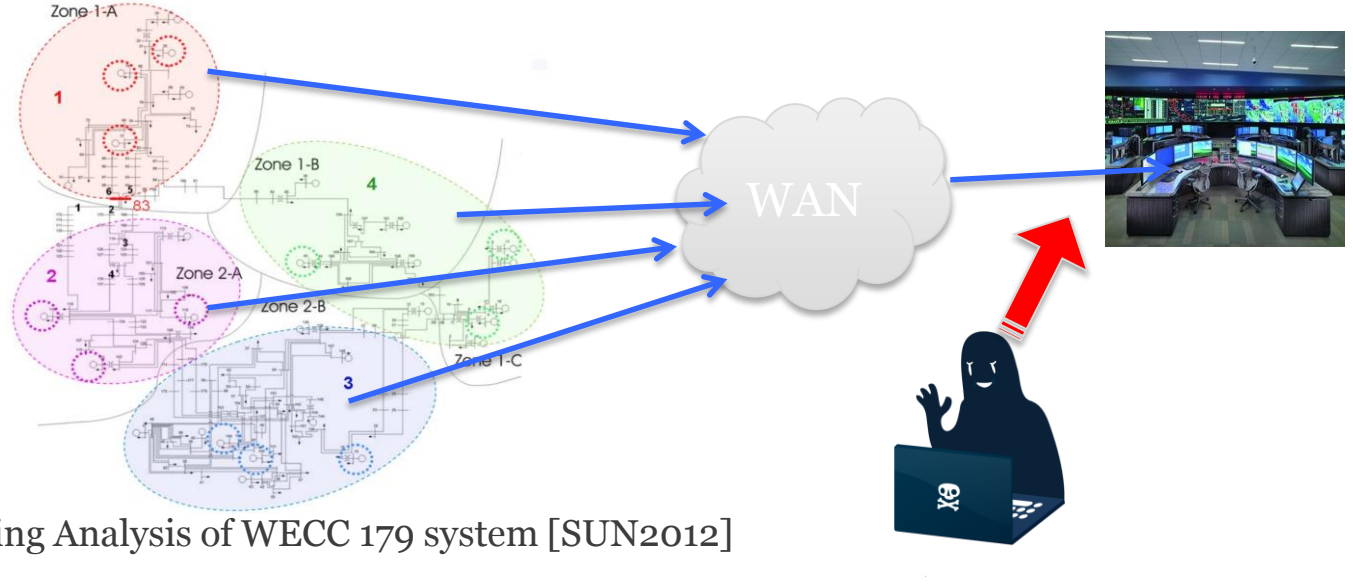
Input validation: ensured to be a 16-bit positive integer

Exploit Development

- Exploit development: Data stream hijacking
  - Exploit vulnerabilities – command frame spoofing and data frame spoofing
  - A practical attack that hijacks the data transmission stream
  - Can be performed with a Scapy script
  - Attackers taking over ongoing phasor phasor data transmission and sending falsified measurement data to the upstream PDC to mislead the user of the data.
  - Demonstrated with WECC 179-bus system model

Exploit Development Scenario

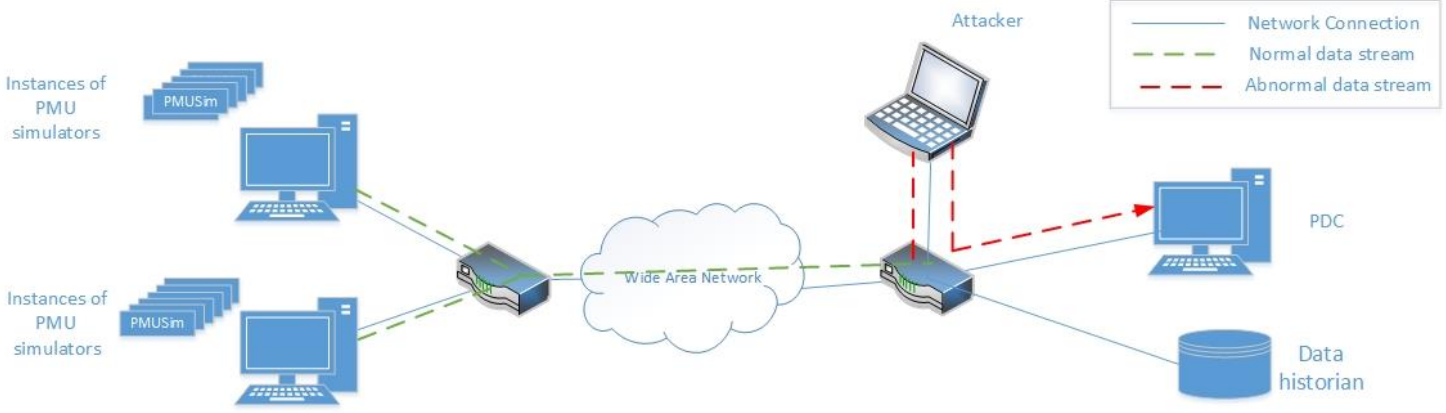
- Exploit development: Data stream hijacking
  - Scenario



Clustering Analysis of WECC 179 system [SUN2012]

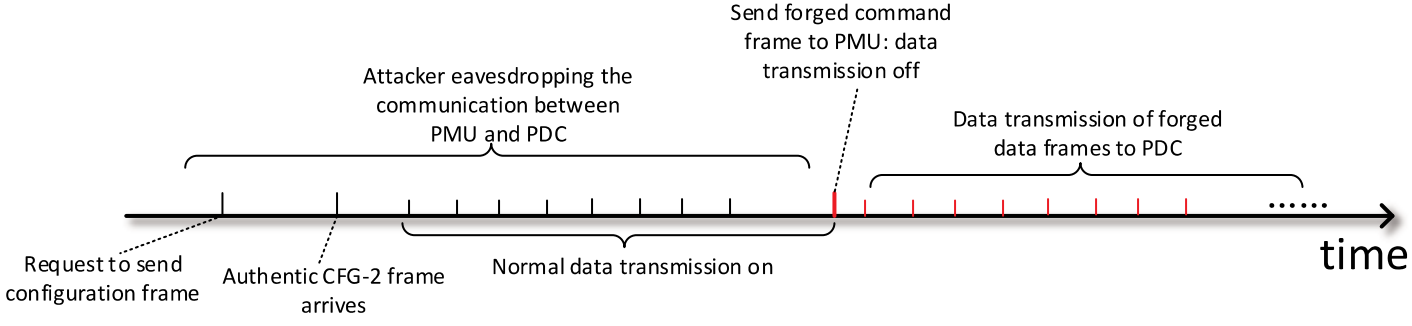
Exploit Development Setup

- Exploit development: Data stream hijacking
  - Testbed set up for demonstration



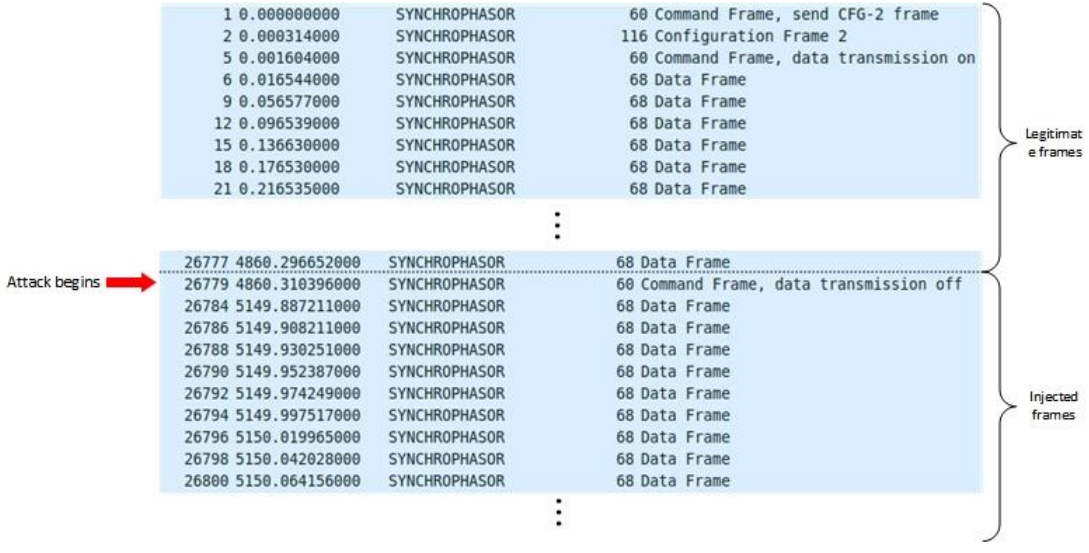
Exploit Development Steps

- Exploit development: Data stream hijacking
  - Attack timeline



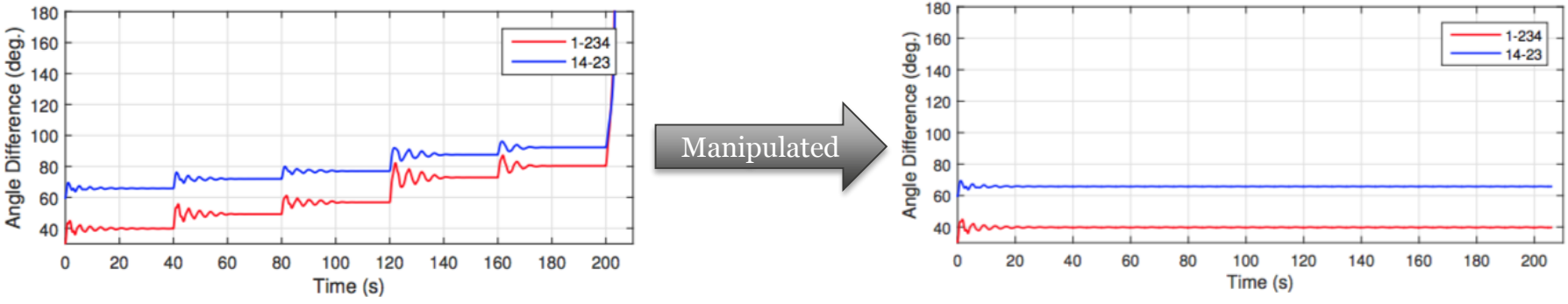
Exploit Development Steps (2)

- Exploit development: Data stream hijacking
  - Wireshark capture during the attack



Exploit Development Result

- Exploit development: Data stream hijacking
  - Impact on situational awareness



Defense and Countermeasures

- Security recommendations and best practices
  - Use encryption (SSL/TLS, IPsec)
  - Enable mutual authentication (X.509 certificates)
  - Use message authentication code (SSL/TLS, IPsec)

  - End-to-end encryption compatible devices should be preferred
  - Thorough fuzz testing of all network interfaces
  - Follow the guideline to avoid SQL injection attack
  - Deploy an intrusion detection system
  - Use redundant devices and communication infrastructure

Let's Try Password Cracking with Kali

- Password cracking
  - some crackers claim 30% success rate
- Try with Kali
  - John the Ripper
  - Hashcat
  - and many more…

Assignment

- Basic hacking exercise: password cracking
- Following the instructions on the remaining slides, show me screenshots of your results and answer questions on the slides.
- In addition, answer this question:

  What is password salting? Is it more secure than unsalted password? Why?

Install Kali

- Download VirtualBox or VMware images (I used this)

https://www.offensive-security.com/kali-linux-vmware-arm-image-download/

- Username: root
- Password: toor

- Or download from Kali.org

https://www.kali.org/downloads/

JtR-unsalted password hash

- Go to Applications (upper left corner) -> 05 password attacks -> john
- Check manual
- Create pwd.txt (or any name) on Desktop (or any directory)

JtR-unsalted password hash

- Insert an entry into pwd.txt
  root:7b24afc8bc80e548d66c4e7ff72171c5
- This is an md5 hash of “toor”
- In the terminal opened, enter Desktop and type:
- john --format=raw-md5 --pot=./list.pot pwd.txt
- Check your list.pot on Desktop
- If stead, you just type this, what do you see?

  john pwd.txt

JtR-salted password hash

- In the termnal, type

unshadow /etc/passwd /etc/shadow > mypass.txt

john mypass.txt

john --show mypass.txt

- Look for your cracked password
- You can also create your own salted password hash list using a generator like this one:

http://online-code-generator.com/md5-hash-with-optional-salt.php

Hashcat-cracking with wordlist

- Download a wordlist (password dictionary) rockyou.txt (or of your choice) on Desktop

http://scrapmaker.com/download/data/wordlists/dictionaries/rockyou.txt

- Check manual
- Create ntlm.txt on Desktop and insert an entry

C27975D3A5B9E95ACD37EC1B1B7598B8

- This is an ntlm hash of “ashley”
- You can insert more entries (maybe sth. not in rockyou.txt) into ntlm.txt

Hashcat-cracking with wordlist

- Go to Applications -> 05 password attacks -> hashcat
- Check manual
- Enter Desktop and type

hashcat -m 1000 ntlm.txt rockyou.txt

hashcat -m 100 ntlm.txt rockyou.txt

- What do you see after each command?

  - 1000 is md5

  - 100 is sha1