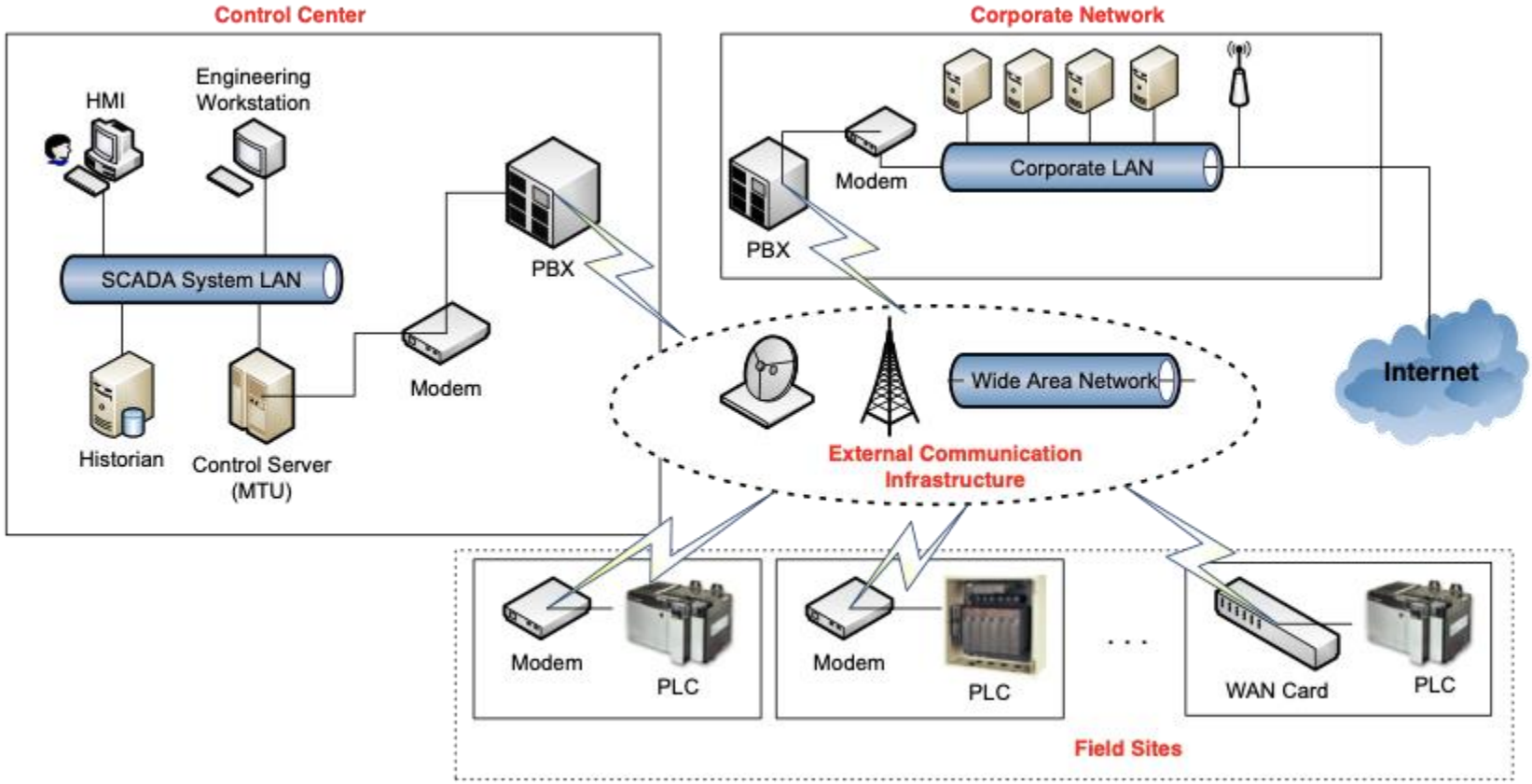
Empirical Study of PLC Authentication Protocols in Industrial Control Systems

Adeen Ayub[1], Hyunguk Yoo[2], and Irfan Ahmed[1]

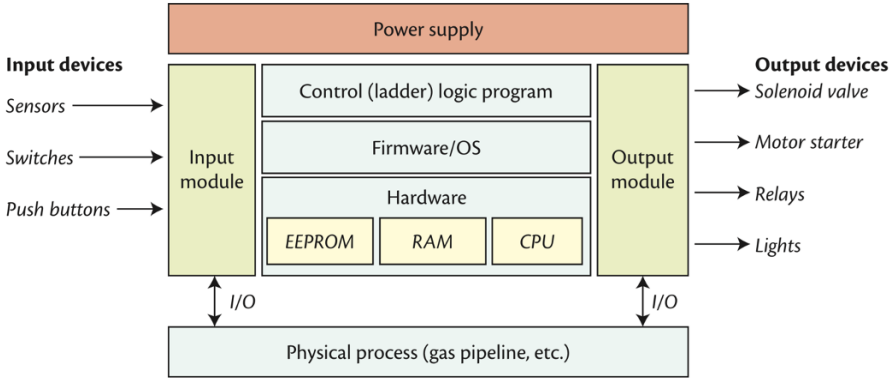
[1] Virginia Commonwealth University
[2] University of New Orleans

Industrial Control System (ICS)

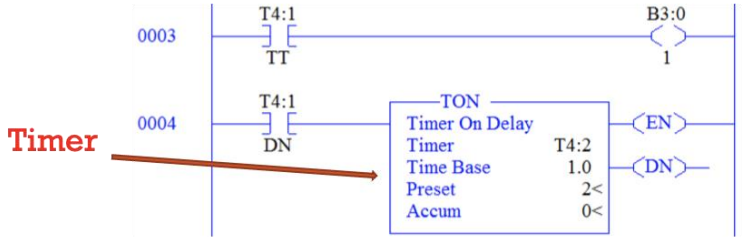


Programmable Logic Controllers (PLCs)

- Monitor and Control physical processes e.g., nuclear plant, and gas pipeline
- Run a control logic program
- Vendor-supplied engineering software
- Proprietary ICS protocol



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 PLCs
  - Sole reliance on network traffic

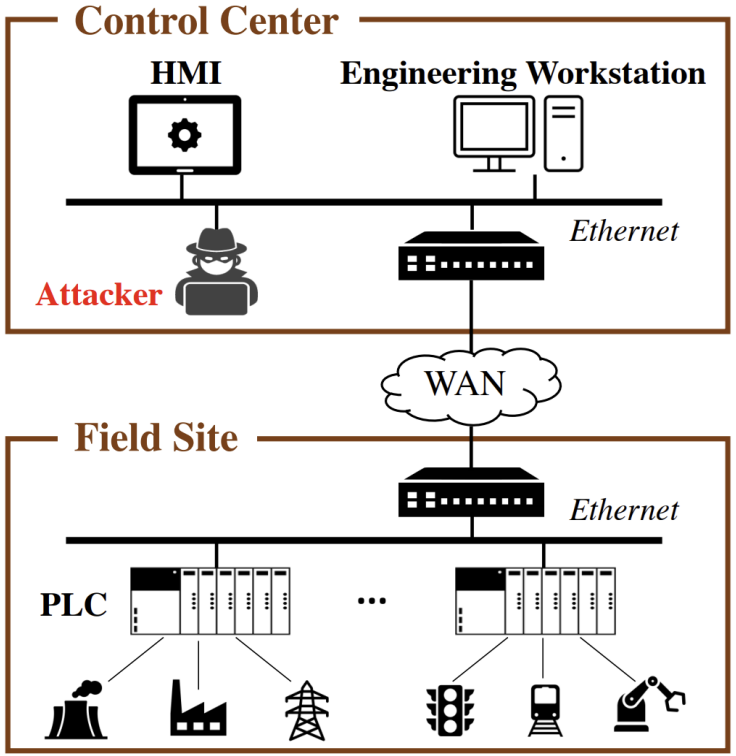
Vendors	PLCs	Engr. Software
Schneider Electric	Modicon M221	SoMachine Basic
Allen-Bradley	MicroLogix 1100 & 1400	RSLogix 500
AutomationDirect	CLICK	CLICK Software
Siemens	S7-300	SIMATIC STEP 7

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

Method

1- Understanding authentication protocol internals

2- Identifying protocol vulnerabilities

3- Mapping an identified vulnerability to the MITRE ATT&CK framework

Findings

- Eight exploitable vulnerabilities discovered
- CVEs include:
  - CVE-2021-32926
  - CVE-2020-15791
  - CVE-2018-7791
  - 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

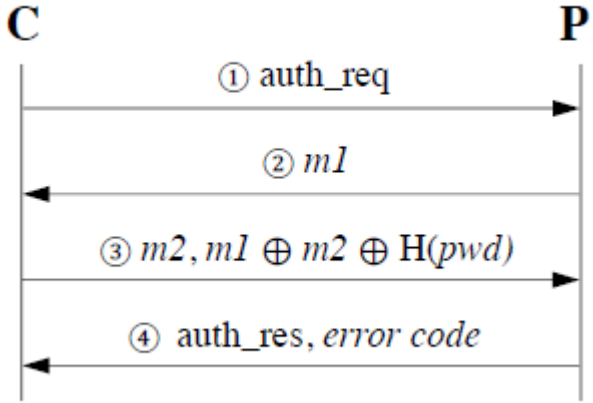
MITRE ATT&CKs launched

MITRE ATT&CK ID	Attack Name	Modicon M221	MicroLogix 1100	MicroLogix 1400	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 1: Modicon M221

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

Authentication Protocol



Protocol Vulnerabilities

1) Weak encryption scheme (V3)

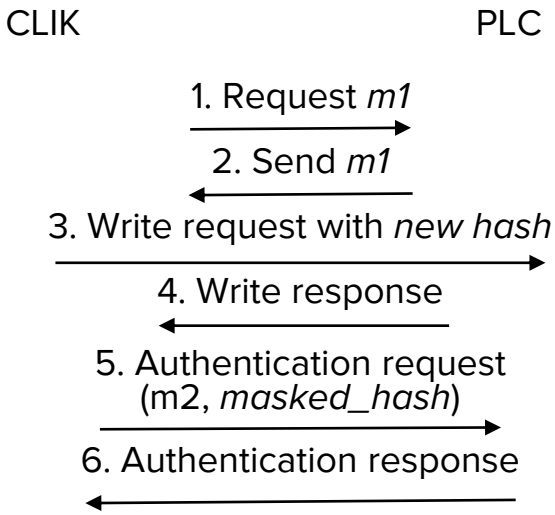
2) Small key size (V4)

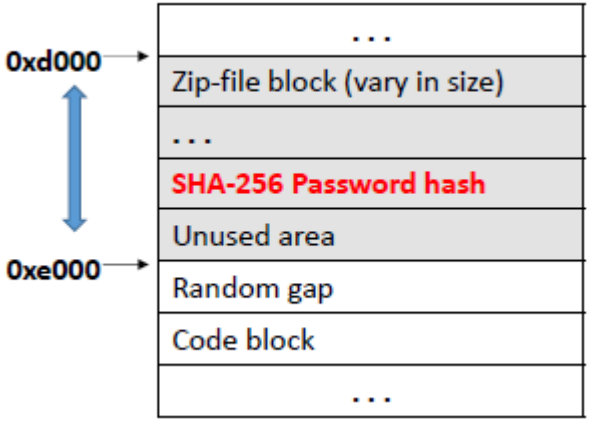
3) No write protection (V8)

MITRE ATT&CK

Unauthorised password reset (T1098)

- Kalle et al.'s password reset attack





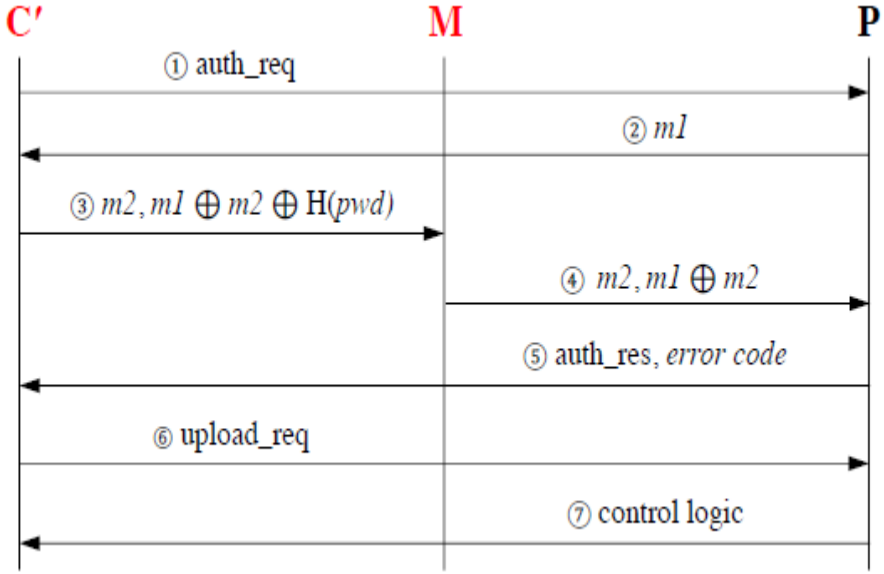
Modicon M221 Memory Layout

MITRE ATT&CK

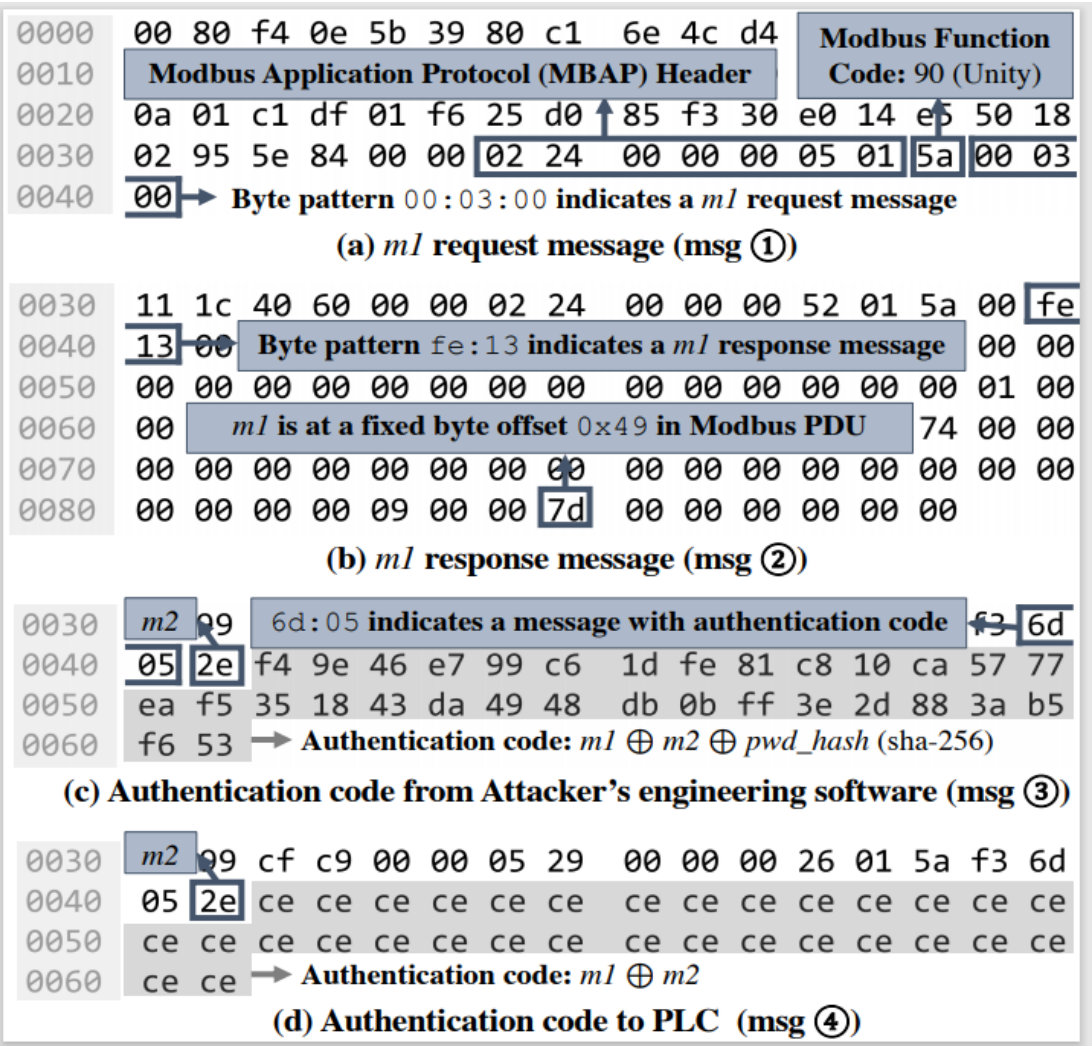
- 0x00ed (efficient) password reset

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
```



Upload a control logic into attacker's ES


0000 00 80 f4 0e 5b 39 80 c1 6e 4c d4
0010 Modbus Application Protocol (MBAP) Header
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
Modbus Function Code: 90 (Unity)
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 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 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

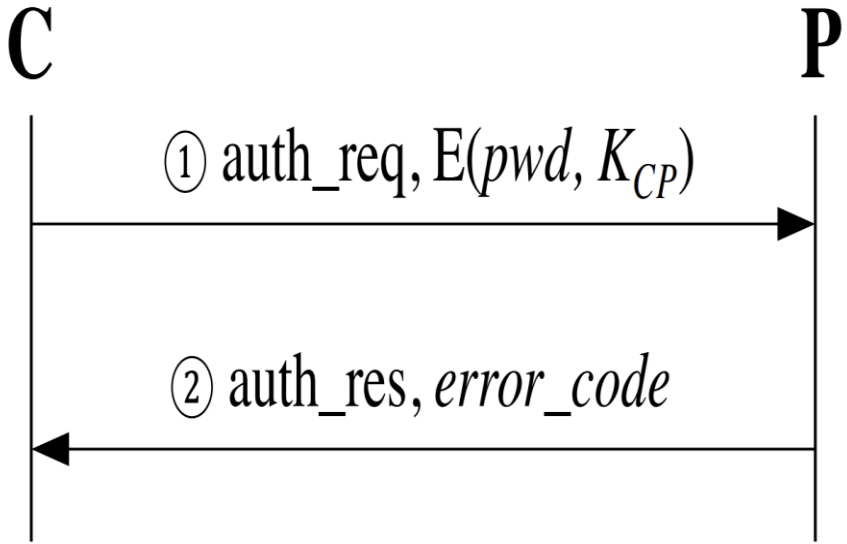
Attack type	Run time /sec	Write requests	Payload size	Failed auth. attempts	Attack success rate
0x00ed (efficient) attack	0.06571	32	128	0	100%
Kalle *et al.*	9.93	2458	32	2457	100%

Case Study 2: Siemens S7-300

- Engineering Software - SIMATIC STEP 7(TIA Portal)
- Users can opt for:
  1- Write protection
  2- Read and write protection

SIEMENS

Authentication Protocol



Encryption Algorithm

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 = Substitute(P_i)
3: **if** i ¡ 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

Character	Encoded (Hex)	Character	Encoded (Hex)	Character	Encoded (Hex)	Character	Encoded (Hex)	Character	Encoded (Hex)	Character	Encoded (Hex)
space	10	@	70	`	50	0	0	P	60	p	40
!	11	A	71	a	51	1	1	Q	61	q	41
“	12	B	72	b	52	2	2	R	62	r	42
#	13	C	73	c	53	3	3	S	63	s	43
$	14	D	74	d	54	4	4	T	64	t	44
%	15	E	75	e	55	5	5	U	65	u	45
&	16	F	76	f	56	6	6	V	66	v	46
‘	17	G	77	g	57	7	7	W	67	w	47
(	18	H	78	h	58	8	8	X	68	x	48
)	19	I	79	i	59	9	9	Y	69	y	49
*	1a	J	7a	j	5a	:	a	Z	6a	z	4a
+	1b	K	7b	k	5b	;	b	[	6b	{	4b
,	1c	L	7c	l	5c	<	c		6c	|	4c
-	1d	M	7d	m	5d	=	d	]	6d	}	4d
.	1e	N	7e	n	5e	>	e	^	6e	~	4e
/	1f	O	7f	o	5f	?	f	_	6f		

Protocol Vulnerabilities

1) Lack of nonce (V5)
2) Weak encryption algorithm (V3)
3) Small key space (V4), i.e., just 8 bits which makes it susceptible to an exhaustive key search attack
4) Same key (V6) for the communication

MITRE ATT&CK

Password Cracking (T1110.002)

Two scenarios

1- Subverting write protection

2- Subverting read/write protection

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

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