Nope, S7ill Not Secure: Stealing Private Keys From S7 PLCs

Nadav Adir, Eli Biham, Sara Bitan, Alon Dankner, Ron Freudenthal, Or Keret

Technion


TLS 1.3 – Iron clad armor


Stuxnet

Cyber-Physical Systems

- PLCs are the core of cyber-physical systems
- Ensure seamless operation of essential services, including
  - Electricity grids
  - Transportation control systems and more…
- Industry 4.0 transforms CPS
  - Transition from isolated air-gapped systems to cloud-connected environment

Who are we?

Alon Dankner

Security Researcher
Technion

Security Researcher
Nokod Security

Nadav Adir

B.Sc. Graduate
Technion

Technion Cyber Lab has a plentiful history of exposing vulnerabilities in Siemens' PLCs

The PLC's Structure and Interfaces

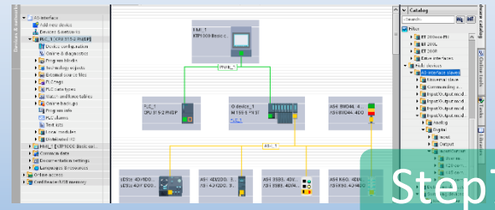
The Operator

The Engineer

TIA Portal

HMI – SCADA

Engineering Station



S7 Protocol

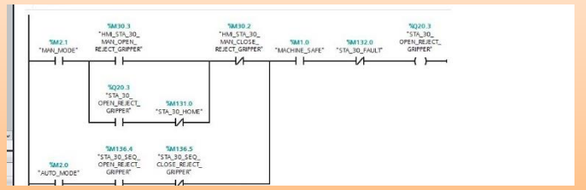
S7 Protocol

The Attacker

PLC

S7-1500

Control Program Execution



The Asset Owner

Edition

S7CommPlus Protocol

- S7 is a proprietary protocol
- Designed to control and monitor the PLCs
  - Examples: program download, PLC configuration, and read/write to PLC variables
- Uses TLS 1.3 for secure communication

Client

```
> Frame 28: 308 bytes on wire (2464 bits), 308 bytes captured (2464 bits) on interface \Device\NP
> Ethernet II, Src: PCSSystemtec_e7:19:ef (08:00:27:e7:19:ef), Dst: SiemensIndus_61:e1:23 (e0:dc:
> Internet Protocol Version 4, Src: 192.168.0.81, Dst: 192.168.0.18
> Transmission Control Protocol, Src Port: 51068, Dst Port: 102, Seq: 420, Ack: 1522, Len: 254
> TPKT, Version: 3, Length: 254
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
v Transport Layer Security
  v TLSv1.3 Record Layer: Application Data Protocol: S7 Communication Plus
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 242
      Encrypted Application Data [truncated]: da4d4a182405e4cbaaf3ab6a817d7d982ea8fec860c56e9ca
      [Application Data Protocol: S7 Communication Plus]
```

PLC

The Evolution of the S7 Protocol

Unencrypted Protocol

Self-Developed Cryptographic Protocol

Standard Protocol but Improperly Implemented

We are here

Stuxnet (Anonymous) 2010

Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs (Biham et al.) 2019

The Race to Native Code Execution in PLCs (Keren) 2021

Research Objective

- Compare the version of S7 protected by TLS to the version protected by the self-developed protocol
  - Is it resilient to attack that the previous version was susceptible to?
  - Is it susceptible to attacks that the previous version was immune to?

- Threat model:
  - The attacker already has network access
  - Any vulnerable device in the network can serve as an attack machine

Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

Summary and Mitigations

Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

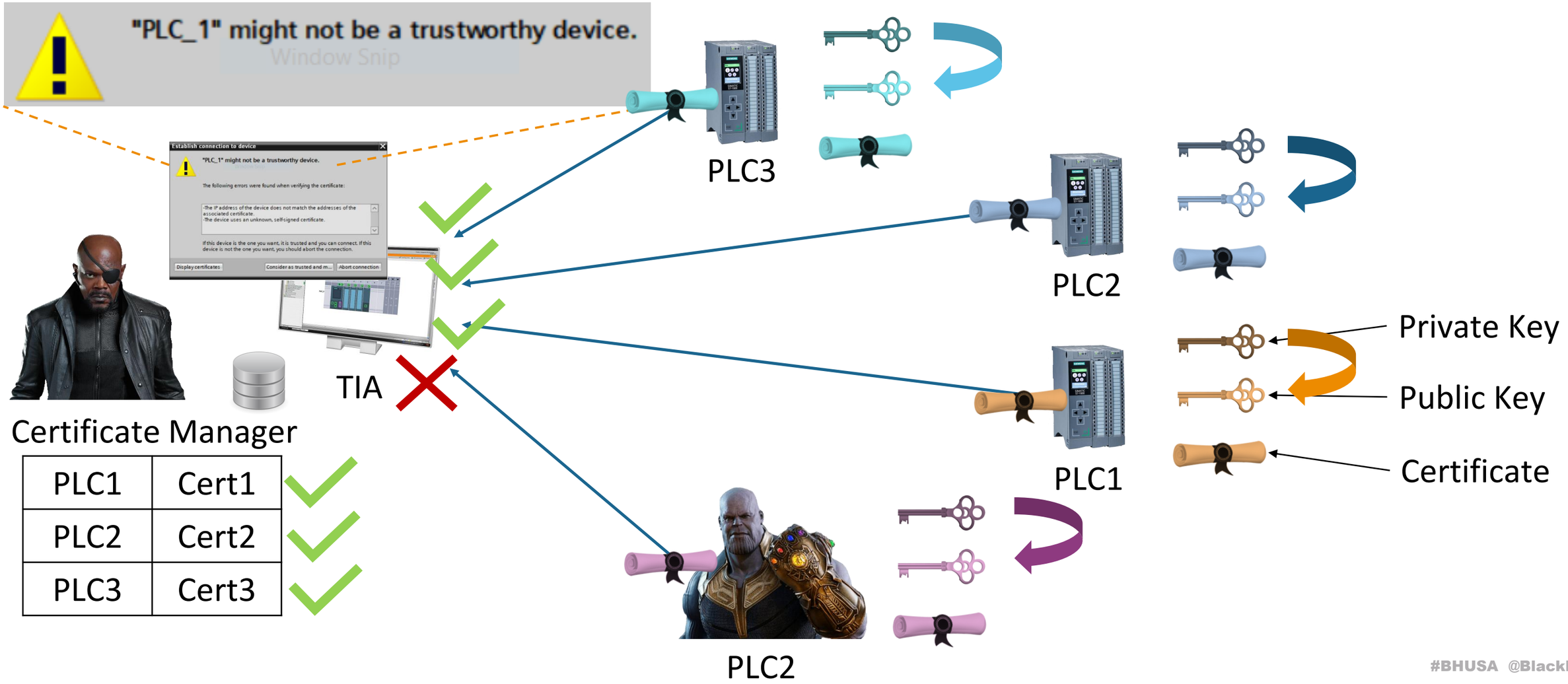
Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

Summary and Mitigations

S7 PLC Authentication

Option 1: Use Initial Self-Signed Certificate



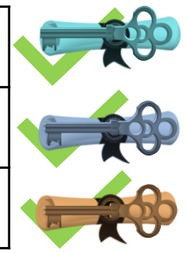
S7 PLC Authentication

Option 2: Use Key and Certificate Provisioned by TIA

TIA

Certificate Manager

PLC1	Cert1
PLC2	Cert2
PLC3	Cert3



PLC3

PLC2

PLC1

PLC2

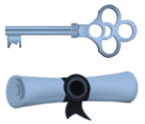
PLC Hardware Configuration Download

- The keys and certificates are part of the PLC hardware configuration
- The PLC configuration includes additionally
  - IP address
  - I/O modules
  - Other settings

Certificate Manager

Configuration Download Request

Response



Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

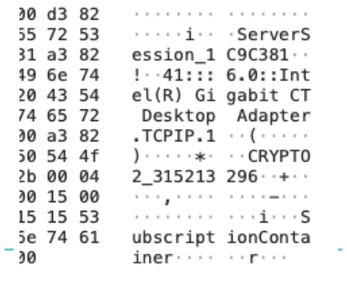
Summary and Mitigations

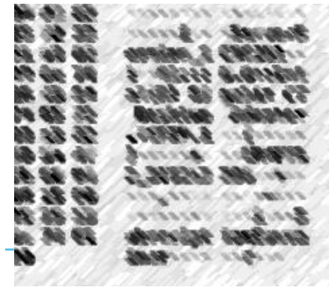




Traffic Interception and Decryption

- A proxy on S7 communication
- Intercepts and manipulates messages between the client and the PLC





Certificate Manager

Request

Attacker's Response

Attacker's Request

Response

Technion's Cyber Lab

S7 Packet Sniffer

- We implemented a packet sniffer
  - Standard TLS proxy tools do not support the S7 protocol stack
- Open-source

```
    57 2.226848    192.168.0.50    192.168.0.18    S7COMM…   122 ←14139 Ver:[V2] Seq=9 [Req GetVarSu
    61 2.257086    192.168.0.18    192.168.0.50    S7COMM…    92 →14139 Ver:[V2] Seq=9 [Res GetVarSu
    65 2.339088    192.168.0.50    192.168.0.18    S7COMM…   122 ←15217 Ver:[V2] Seq=10 [Req GetVarS
    69 2.371321    192.168.0.18    192.168.0.50    S7COMM…   150 →15217 Ver:[V2] Seq=10 [Res GetVarS
    73 2.486904    192.168.0.50    192.168.0.18    S7COMM…   122 ←18425 Ver:[V2] Seq=11 [Req GetVarS
    77 2.528822    192.168.0.18    192.168.0.50    S7COMM…    89 →18425 Ver:[V2] Seq=11 [Res GetVarS
```

```
▼ S7 Communication Plus
  ▼ Header: Protocol version=V2
      Protocol Id: 0x72
      Protocol version: V2 (0x02)
      Data length: 53
  ▼ Data: Request SetVariable
      Opcode: Request (0x31)
      Reserved: 0x0000
      Function: SetVariable (0x04f2)
      Reserved: 0x0000
      Sequence number: 12
      Session Id: 0x70000cac
```

```
0000  e0 dc a0 61 e1 23 00 1b  21 39 b8 e6 08 00 45 00   ···a·#·· !9····E·
0010  00 6c 00 01 00 00 40 06  f8 f6 c0 a8 00 32 c0 a8   ·l····@· ·····2··
0020  00 12 3e b9 00 66 00 00  00 01 00 00 00 01 50 18   ··>··f·· ······P·
0030  20 00 b9 3f 00 00 03 00  00 44 02 f0 80 72 02 00    ··?···· ·D···r··
0040  35 31 00 00 04 f2 00 00  00 0c 70 00 0c ac 34 00   51····· ··p···4·
0050  00 00 34 01 90 77 00 08  01 00 00 04 e8 89 69 00   ··4··w·· ·····i·
0060  12 00 00 00 00 89 6a 00  13 00 89 6b 00 04 00 00   ·····j· ···k····
0070  00 06 00 00 00 00 72 02  00 00                     ·····r· ··
```

Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

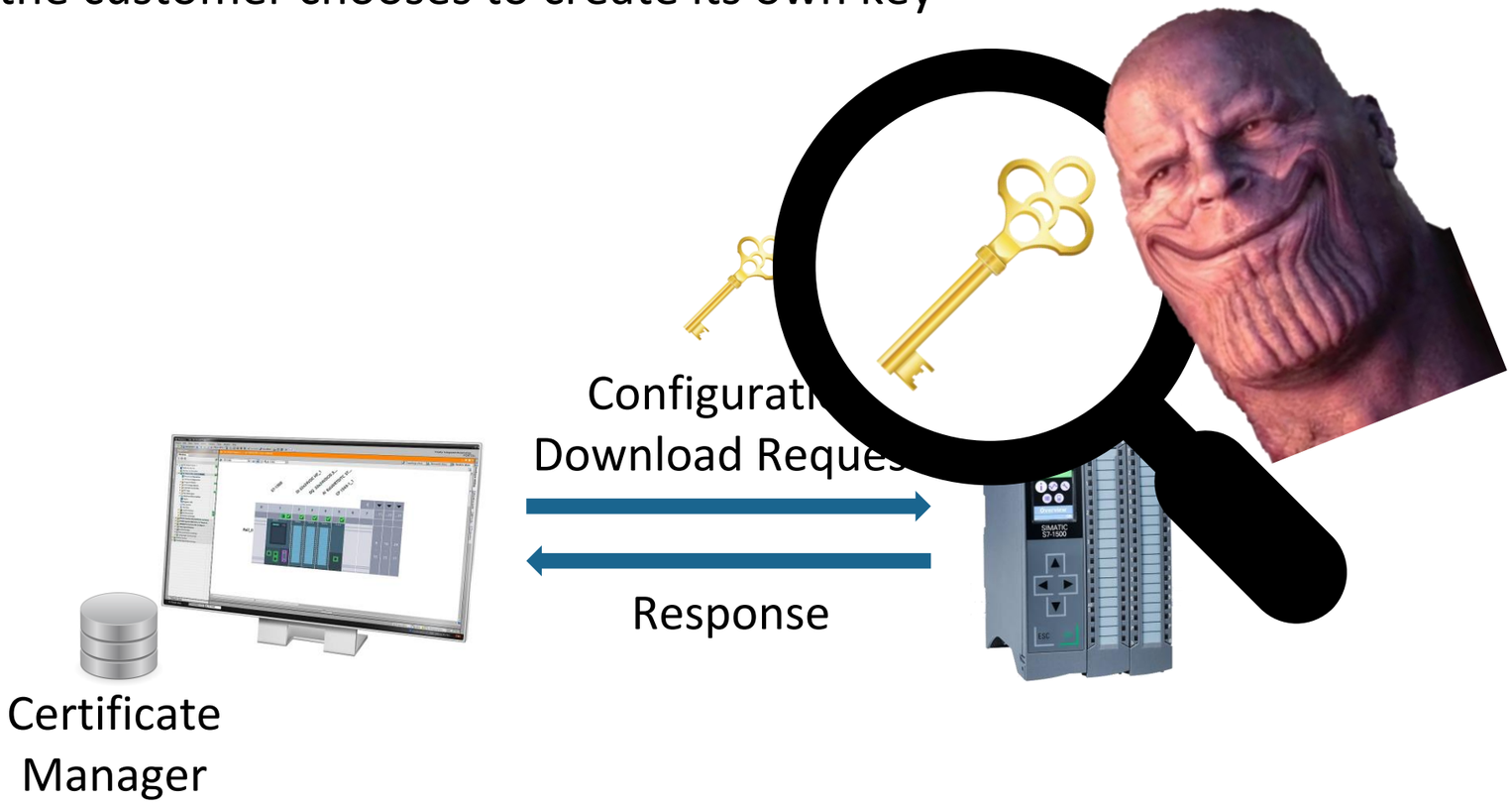
Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

Summary and Mitigations

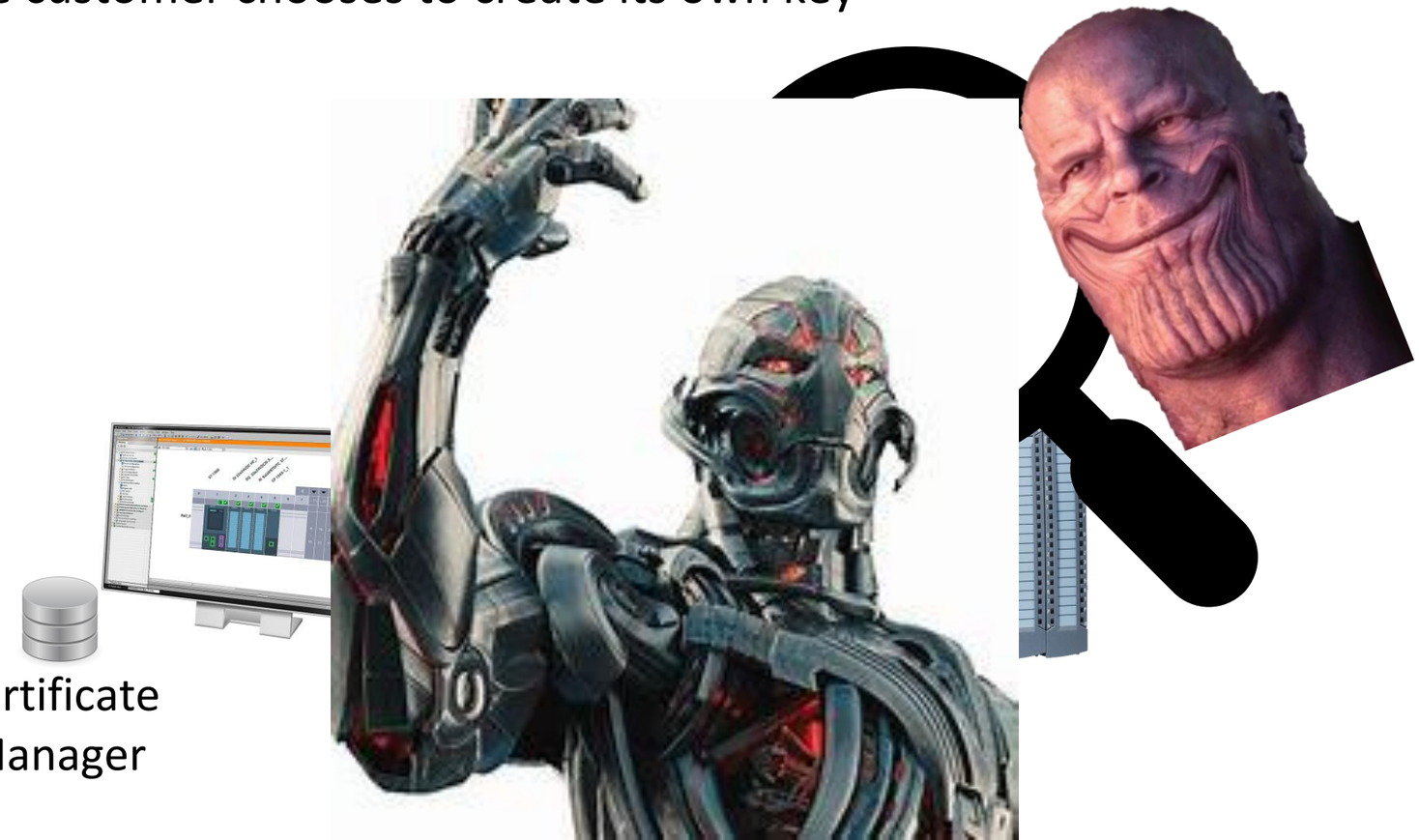
PLC's Key Provisioning

- Certificate and private key are issued to the PLC by the TIA
  - Occurs if the customer chooses to create its own key



PLC’s Key Provisioning

- Certificate and private key are issued to the PLC by the TIA
  - Occurs if the customer chooses to create its own key



The S7 Configuration Download Protocol

Client

PLC

pk_0
sk_0

$cert_0$

TLS 1.3 Handshake

ClientHello

ServerHello, **$cert_0$**, CertVerify, Finished

Finished

pk_i
sk_i

$cert_i$

$EncryptedSecretKey_i$

$hash = SHA1(2, pk_i, cert_i.fingerprint)$

Hardware Configuration Download Request

$EncryptedSecretKey_i$, $cert_i$, hash, salt

$SHA1(2, pk_i, cert_i.fingerprint) \stackrel{?}{=} hash$

$sk_i = Dec\_func(EncryptedSecretKey_i)$

New S7 Session

ClientHello

ServerHello, **$cert_i$**, CertVerify, Finished

pk_i
sk_i

$cert_i$

The PLC Private Key Protection

- We reverse-engineered the private key encryption process
  - It uses standard encryption algorithms
  - The private key is encrypted by a sequence of ephemeral keys

```
124 8.728769    192.168.0.50    192.168.0.18    S7COMM…    99 ←17449 Ver:[V2] Seq=14 [Req Explore] Area=NativeObjects.theCommCont_Rid
128 8.795574    192.168.0.18    192.168.0.50    S7COMM…   306 →17449 Ver:[V2] Seq=14 [Res Explore] Retval=OK NativeObjects.theASRoot_Rid
132 10.661288   192.168.0.50    192.168.0.18    S7COMM…    99 ←10650 Ver:[V2] Seq=15 [Req Explore] Area=NativeObjects.thePkiContainer_Rid
136 10.733636   192.168.0.18    192.168.0.50    S7COMM…  4106 →10650 Ver:[V2] Seq=15 [Res Explore] Retval=OK NativeObjects.theASRoot_Rid
```

```
Item Value: ID=PkiItem.PublicData_Rid (Blob) = 0x010003722d2d2d2d2d42454749
    ID Number: PkiItem.PublicData_Rid
  > Datatype flags: 0x00
    Datatype: Blob (0x14)
    Blob root ID: None
    Blob size: 886
    Value [truncated]: 010003722d2d2d2d2d424547494e204345525449464943415445
Attribute
  Element Tag-Id: Attribute (0xa3)
  Item Value: ID=PkiItem.PrivateData_Rid (Blob) = 0x010001008bc8dc94df766718d
```

```
0e70  2d 2d 2d 2d 2d 42 45 47  49 4e 20 45 4e 43 52 59   -----BEG IN ENCRY
0e80  50 54 45 44 20 50 52 49  56 41 54 45 20 4b 45 59   PTED PRI VATE KEY
0e90  2d 2d 2d 2d 2d 0a 4d 49  48 73 4d 46 63 47 43 53   -----·MI HsMFcGCS
0ea0  71 47 53 49 62 33 44 51  45 46 44 54 42 4b 4d 43   qGSIb3DQ EFDTBKMC
0eb0  6b 47 43 53 71 47 53 49  62 33 44 51 45 46 44 44   kGCSqGSI b3DQEFDD
0ec0  41 63 42 41 6a 2b 33 44  66 50 4c 74 42 6c 33 41   AcBAj+3D fPLtBl3A
0ed0  49 43 43 41 41 77 0a 44  41 59 49 4b 6f 5a 49 68   ICCAAw·D AYIKoZIh
0ee0  76 63 4e 41 67 6b 46 41  44 41 64 42 67 6c 67 68   vcNAgkFA DAdBglgh
0ef0  6b 67 42 5a 51 4d 45 41  53 6f 45 45 45 52 6c 37   kgBZQMEA SoEEERl7
0f00  5a 2f 47 39 6a 70 4d 4a  52 73 61 4d 6c 57 49 54   Z/G9jpMJ RsaMlWIT
0f10  70 67 45 67 5a 43 2b 0a  6f 44 49 4c 37 76 67 4e   pgEgZC+· oDIL7vgN
0f20  35 30 31 54 75 62 6c 77  68 71 38 31 2f 76 49 71   501Tublw hq81/vIq
```

Private Key Encryption

"Hey Jarvis, encrypt the private key and make it as complex as possible"

"Right away sir, I'll just use the TIA Portal"

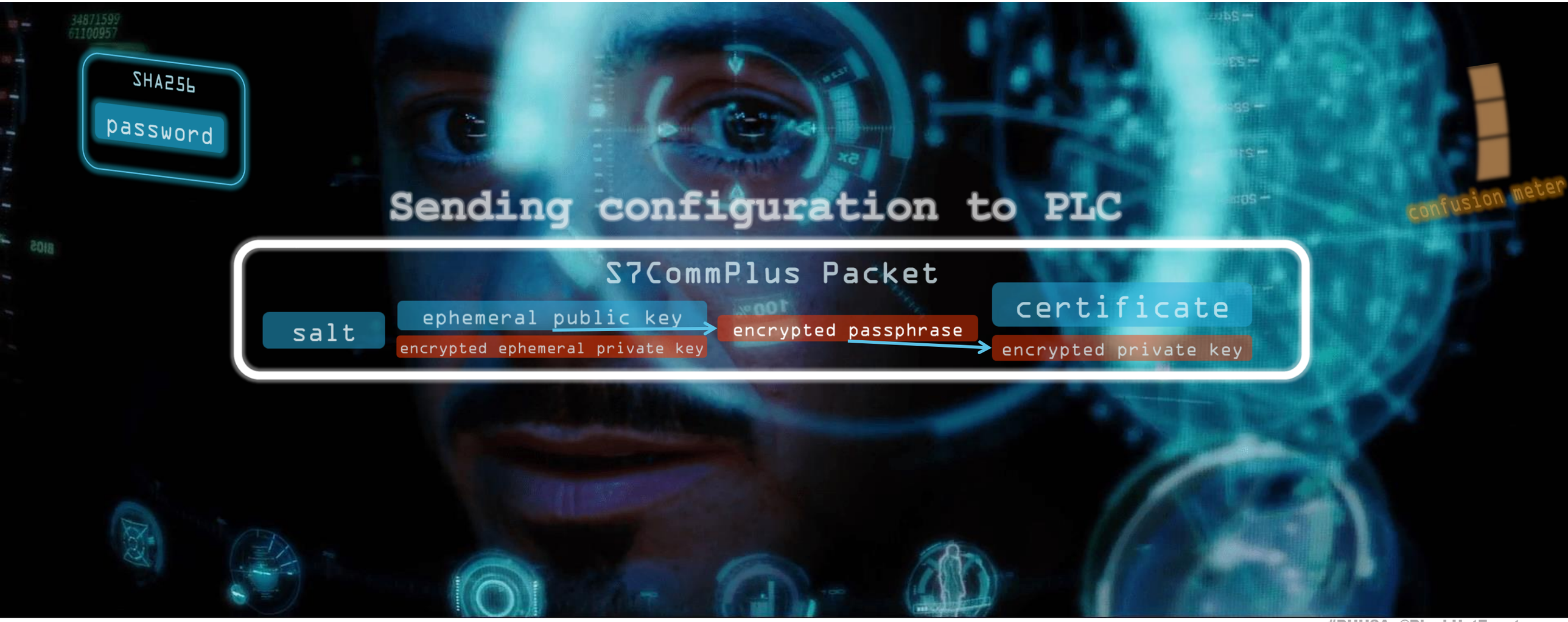
Private Key Encryption

Private Key Encryption

Private Key Encryption

Private Key Encryption

Private Key Encryption



Private Key Encryption

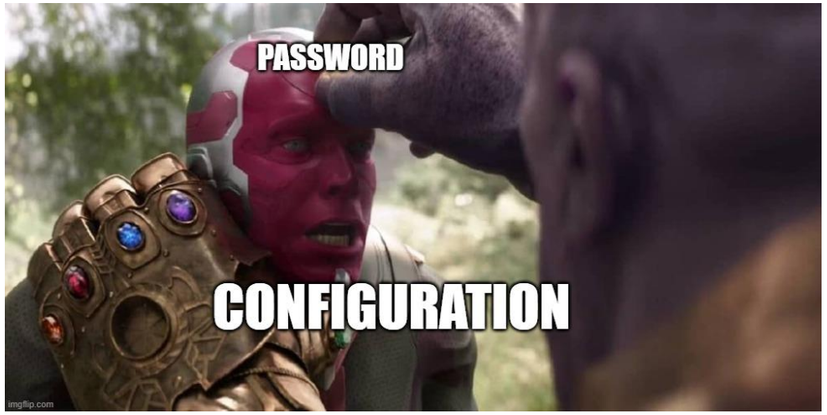
Private Key Encryption

The PLC Configuration Password

- When the user creates a new TIA project, he can setup the PLC configuration password
- Complexity policy:
  - >8 characters, lowercase letter, uppercase letter, and a number
- If the user does not setup the password, a null string is used

Provisioning the Configuration Password

- The password is sent to the PLC on the first configuration download of the project
- In plaintext! (over TLS)
- Attacker can steal the password



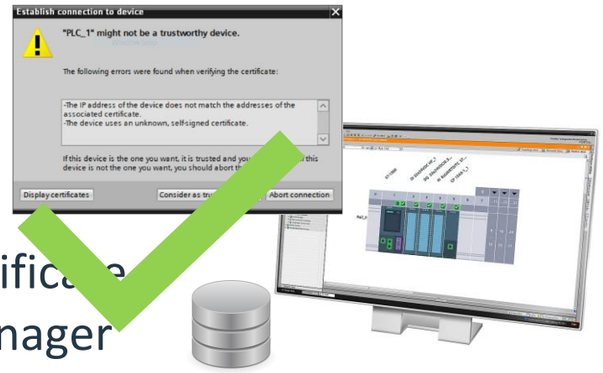

```
396 42.466587   192.168.0.50   192.168.0.18   S7COMM…   134 ←14465 Ver:[V2] Seq=50 [Req SetVarSubStreamed]
400 43.204392   192.168.0.18   192.168.0.50   S7COMM…    85 →14465 Ver:[V2] Seq=50 [Res SetVarSubStreamed] Retval=OK
```

```
S7 Communication Plus
  Header: Protocol version=V2
  Data: Request SetVarSubStreamed
      Opcode: Request (0x31)
      Reserved: 0x0000
      Function: SetVarSubStreamed (0x057c)
      Reserved: 0x0000
      Sequence number: 50
```

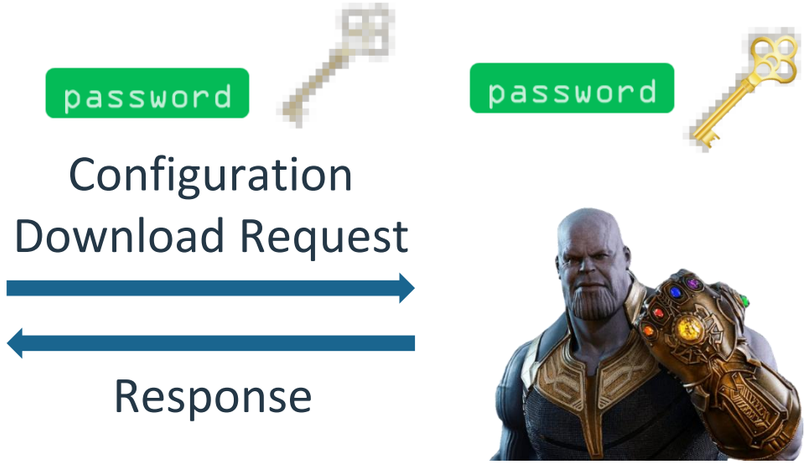
```
0000  e0 dc a0 61 e1 23 00 1b  21 39 b8 e6 08 00 45 00   ···a·#·· !9····E·
0010  00 78 00 01 00 00 40 06  f8 ea c0 a8 00 32 c0 a8   ·x····@· ·····2··
0020  00 12 38 81 00 66 00 00  00 01 00 00 00 01 50 18   ··8··f·· ······P·
0030  20 00 f0 2b 00 00 03 00  00 50 02 f0 80 72 02 00    ··+···· ·P···r··
0040  41 31 00 00 05 7c 00 00  00 32 70 00 0c ac 34 00   A1···|·· ·2p···4·
0050  00 00 c9 20 04 01 83 1f  00 00 04 e8 89 69 00 12   ··· ···· ·····i··
0060  00 00 00 00 89 6a 00 13  00 89 6b 00 04 00 00 00   ·····j·· ··k·····
0070  01 00 14 00 08 41 61 31  32 33 34 35 36 0b 00 00   ·····Aa1 23456···
0080  00 00 72 02 00 00                                  ··r···
```

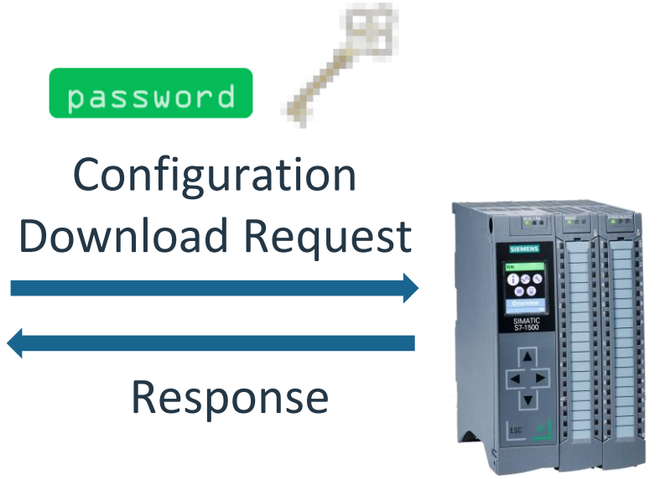
Attack #1: Extracting the PLC's Secrets During the Initial Key Provisioning

- **Impact: The attacker can decrypt and modify all the PLC-TIA communication**
- The MITM intercepts the configuration password
  - and the encrypted TLS private key
- Since this is the first connection, TIA didn't pin the certificate yet…
  - The user must trust the certificate which might be authentic or forged…



Certificate Manager





Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

Summary and Mitigations

Attack #2: Rogue Client

- **Impact: The attacker manipulates the PLC operation**
- We implemented a python script that impersonates a legitimate client
  - Exploits the lack of client authentication
  - Uses open-source libraries (in previous research we had to RE and implement the protocol)
- The attacker sends arbitrary control commands to the PLC
  - such as stop CPU, modify PLC configuration, or write to PLC variables

Attacker's Request

Response

Attack #3: PLC Private Key Retrieval

New Attack

- **Impact: Decrypts and modifies all the network traffic, at any time**
- Use the rogue client to send upload configuration request
- **The returned configuration contains the private key**
- The decryption process is identical to the one used in initial key provisioning attack
- $Knowing\ the\ password \rightarrow Knowing\ the\ private\ key$

Attack #1  Attack #3

Initial Key Provisioning	PLC Key Retrieval
Must be performed during initial provisioning	Can be performed any time
Attacker needs network access to both PLC and TIA	Attacker only needs access to the PLC
Can be performed if user hasn't setup a password	Can be performed if user hasn't setup a password
Can be performed if user has setup a password	If user has setup a password, the attacker must know it

Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

Retrieving the Private Key of a Production PLC

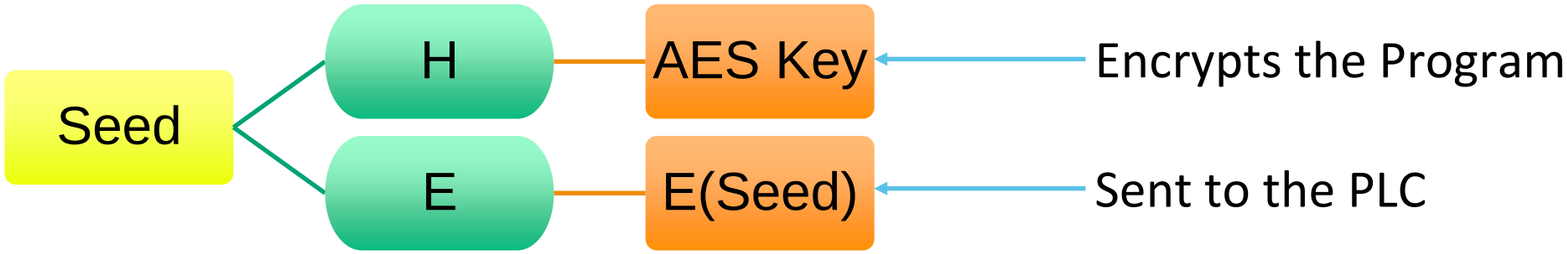
A Stealth MITM Attack

Summary and Mitigations

Control Program Protection

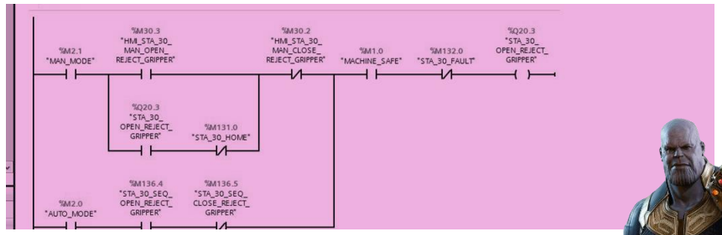
- TIA uses AES to encrypt the control program
  - The AES key is derived the from a random seed
  - The seed is encrypted under the hardcoded common PLC key
- The encrypted program and the encrypted seed are sent to the PLC over TLS

Shared by all PLCs with the same firmware version



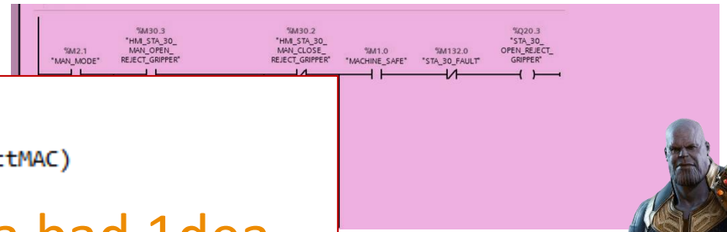
Attack #4: Malicious Control Program Injection

- **Impact: The PLC runs a malicious control program**
- The program transmission is susceptible to replay attacks
- The attacker creates a malicious control program in his own lab
  - and uses rogue client to download it to any PLC



Attack #4: Malicious Control Program Injection

- **Impact: The PLC runs a malicious control program**
- The program transmission is susceptible to replay attacks
- The attacker creates a malicious control program in his own lab
  - and uses rogue client to download it to any PLC



```
Attribute
   Element Tag-Id: Attribute (0xa3)
   Item Value: ID=Block.AdditionalMAC (Struct) = 1820 (StructMAC)
      ID Number: Block.AdditionalMAC
      Datatype flags: 0x00
      Datatype: Struct (0x17)
      Value: StructMAC
      Item Value: ID=MACAlgorithm (UDInt) = 0
      Item Value: ID=MACEncryptedKey (Struct) = 1800 (StructSecurityKey)
      Item Value: ID=MACData (Blob) = 0xea1dadab8c00000001000000000000f5bb2d96580a8
      Terminating Item/List
Attribute
   Element Tag-Id: Attribute (0xa3)
   Item Value: ID=FunctionalObject.Code (Blob) = 0xefbeadde7c00000001000000020000003
      ID Number: FunctionalObject.Code
```

0xea1dadab = a bad 1dea

Attack #5: Rogue PLC

New Attack

- **Impact: Impersonates a legitimate PLC and send forged status messages**
  - using the retrieved private key and certificate
- TIA does not show a warning message
  - Since the pinned certificate, issued to the extracted key, is used

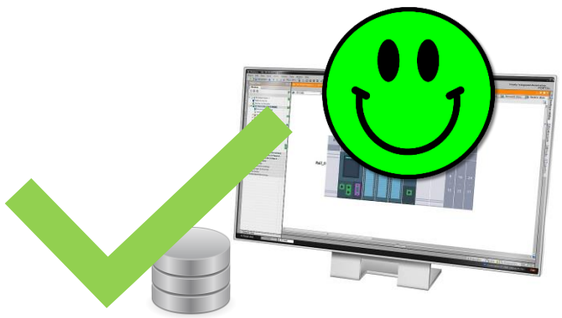
Certificate Manager

Request

Attacker's Response

Attack #6: A Stealth MITM Attack

- **Impact: Stuxnet effect:**
  - When the operator queries the PLC state
    - as the malicious program is running
  - The MITM substitutes the real state with a forged healthy-looking state
    - displayed to the operators
  - No warning message is presented to the user (as in Rogue PLC)



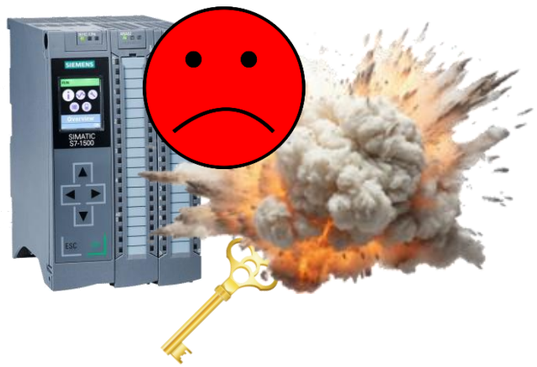
Certificate Manager

State Request

Attacker's Response ☺

State Request

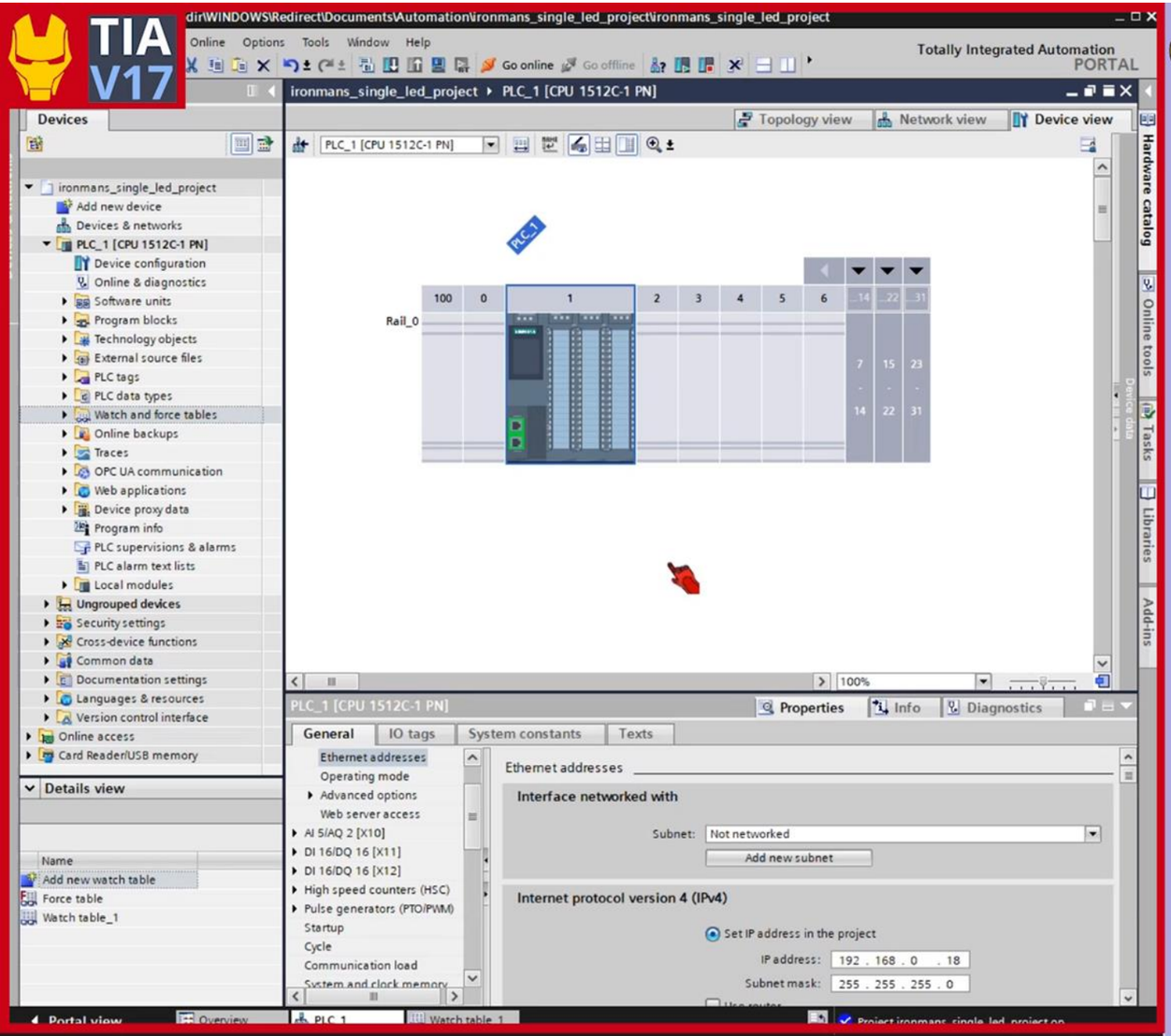
Real Response ☹

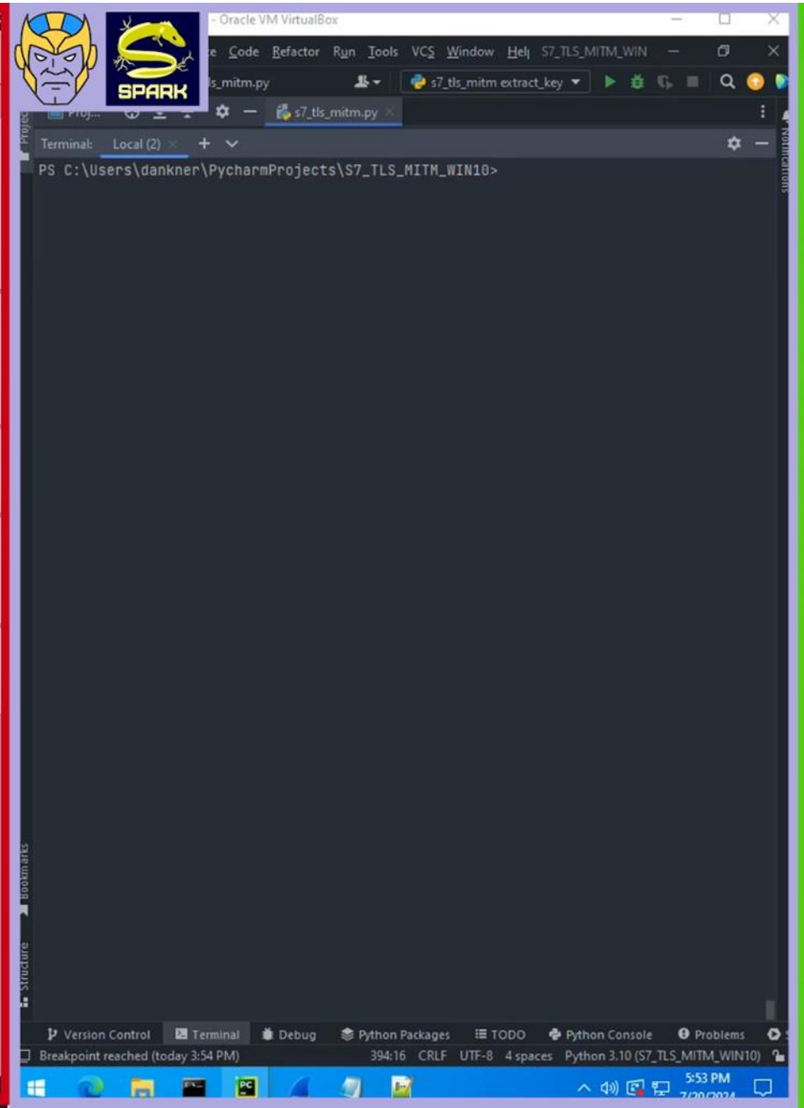


S7 Version Comparison

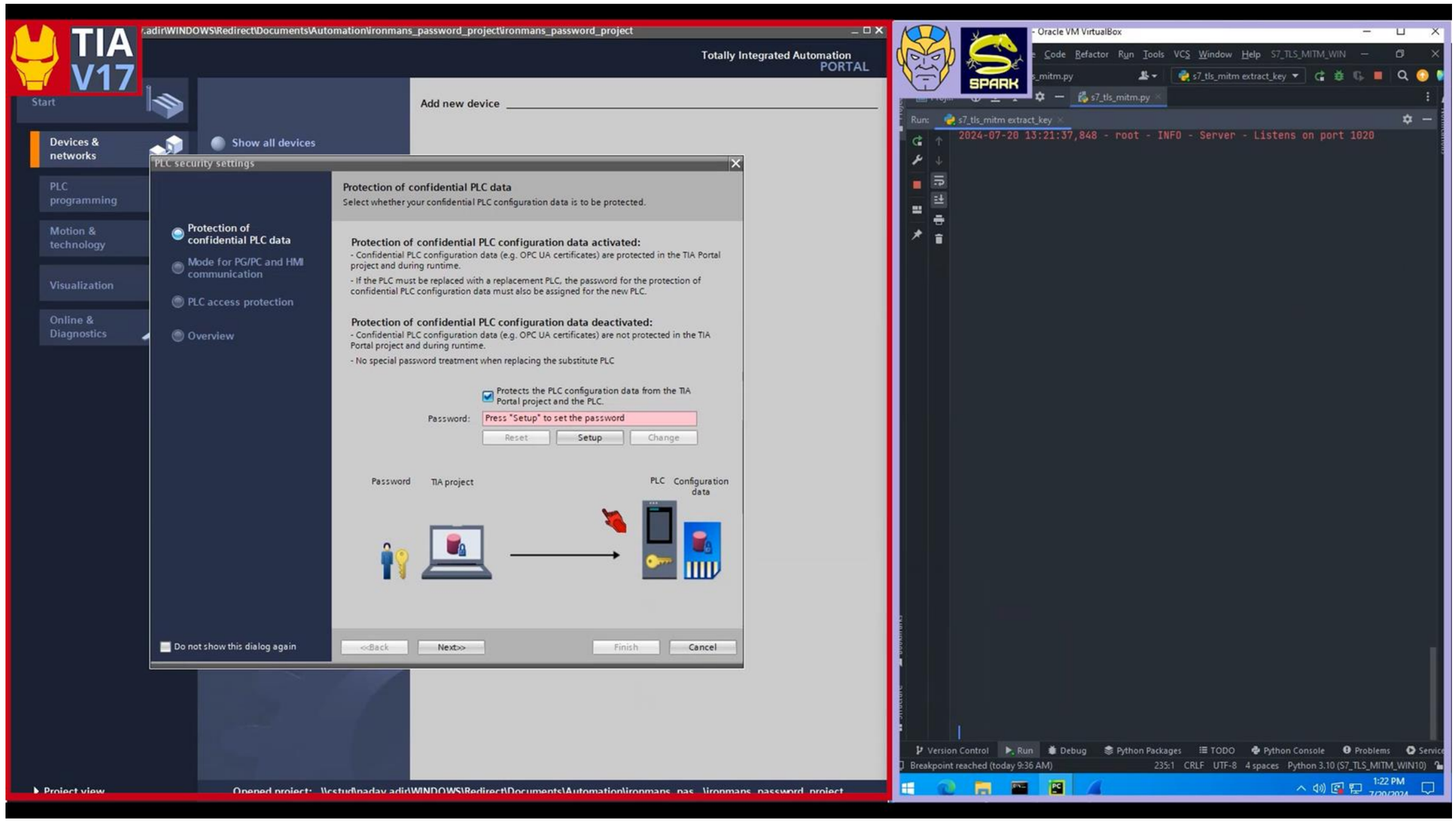
#	Attack	S7 w/o TLS	S7 over TLS
1	Extracting PLC secrets during initial download	X	√
2	Rogue client	√	√
3	PLC private key retrieval	X	√
4	Malicious control program injection	X	√
5	Rogue PLC	√	√
6	A stealth MITM attack	X	√

Stealth Man-in-the-Middle Attack Demo





Password Stealing Demo



Talk Roadmap

Background on the S7 PKI

Traffic Interception and Decryption

Extracting the Private Key During Initial Provisioning

Retrieving the Private Key of a Production PLC

A Stealth MITM Attack

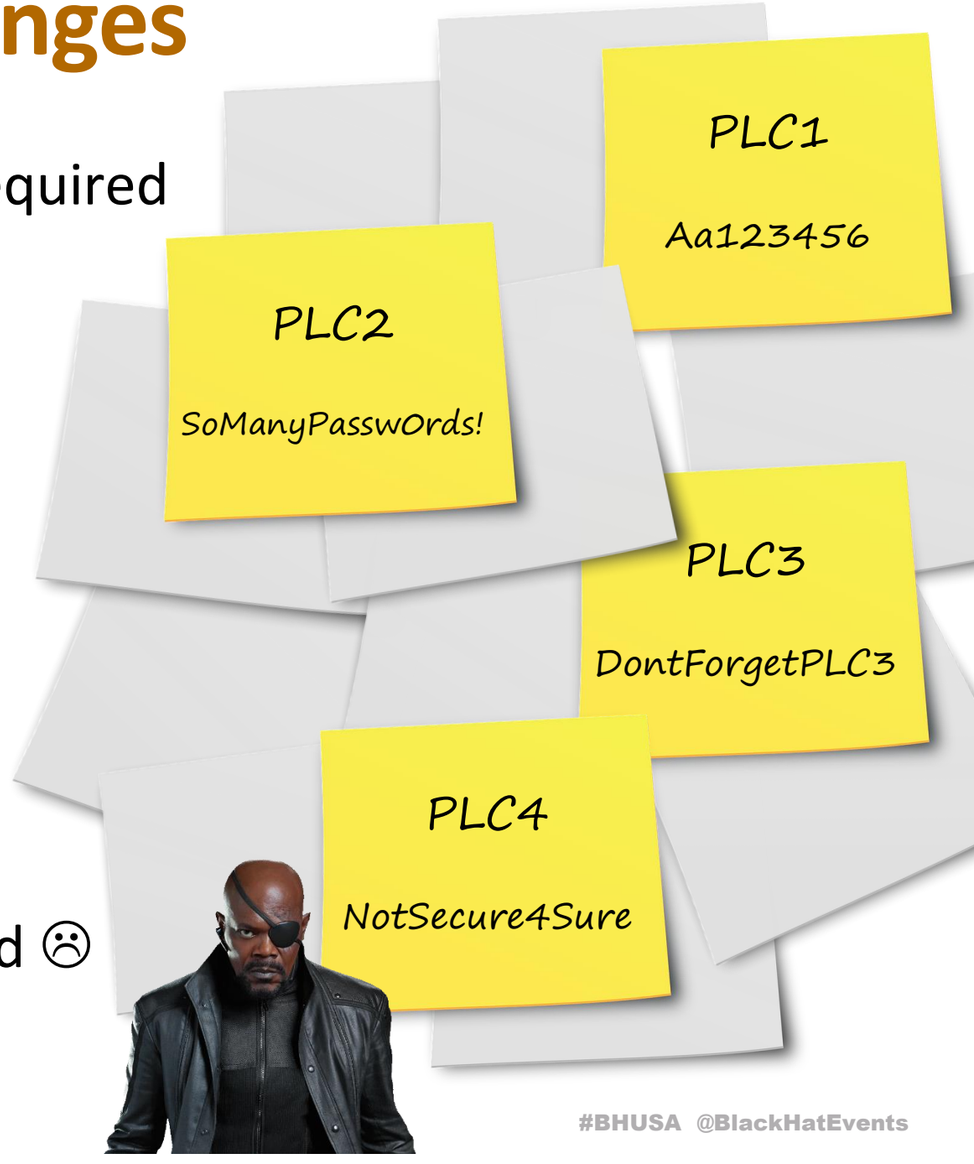
Summary and Mitigations

Mitigations – ICS Customers

- Set all three kinds of passwords supported by S7
  1. PLC configuration password
  2. CPU access protection password
     - that restricts access to privileged op
- To use CA certificates
  3. TIA user password
- To protect the control prog
  4. Know-how protection passv
- Passwords must be strong and unique per PLC

Mitigations – Challenges

- A huge number of strong and unique passwords is required
  - For all users and PLCs
- Clearly – unmanageable
  - Therefore, users do not use them
  - Or set the same passwords on all PLCs
    - Thus, eliminating the security effect of these passwords
- Protecting all those passwords is almost impossible
- Trusted CA does not help if the private key is exposed ☹



Siemens' Response

- We discloded our findings to Siemens about a year ago

- In order to mitigate the attacks, Siemens recommends customers to:
  - Perform initial provisioning in a secure environment
  - Use passwords!

Mitigations – PLC Vendors

Implement full mutual authentication

Each party authenticates the other party

The private key must not leave the PLC

It should be impossible to retrieve the private key from the PLC

Avoid self-signed certificates for initialization

Use vendor certificates instead

Use key exchange protocols

Don’t send the key/passwords over the network

The K7 Protocol

- Countermeasure: a new security architecture
- Provides authentication and authorization
- Based on capability tickets

Biham, Eli, Bitan, Sara, and Dankner, Alon. **"K7: A Protected Protocol for Industrial Control Systems that Fits Large Organizations."** Sixth Annual Industrial Control System Security (ICSS) Workshop. 2020. https://dl.acm.org/doi/abs/10.1145/3442144.3442149.

Patent US 2022/0182229 A1

A Message to the Customers

Demand Secure Products!!

Thank you

Nadav Adir, Eli Biham, Sara Bitan, Alon Dankner, Ron Freudenthal, Or Keret

Technion

{nadav.adir,biham,sarab,dankner,ronf,or.keret}@cs.technion.ac.il