

The secrets of Schneider Electric's UMAS protocol

Pavel Nesterov
Nikita Komarov
Andrey Muravitsky

Object of research..... 3

UMAS protocol..... 3

 Network packet structure 3

 Network communication 4

 Reservation procedure 6

 Session key 6

UMAS protocol functions 8

 Functions used in the device reservation process..... 8

 Functions that require device reservation 8

 Functions that do not require device reservation 9

 Functions that control the state of the PLC 9

CVE-2020-28212: authentication bypass without Application Password..... 9

Application Password..... 10

Authentication bypass with Application Password..... 11

Updated reservation procedure with Application Password16

Conclusion18

UMAS (Unified Messaging Application Services) is a proprietary Schneider Electric (SE) protocol that is used to configure and monitor Schneider Electric PLCs.

Schneider Electric controllers that use UMAS include Modicon M580 CPU (part numbers BMEP* and BMEH*) and Modicon M340 CPU (part numbers BMXP34*). Controllers are configured and programmed using engineering software – EcoStruxure™ Control Expert (Unity Pro), EcoStruxure™ Process Expert, etc.

In 2020, a vulnerability, [CVE-2020-28212](#), was reported, which could be exploited by a remote unauthorized attacker to gain control of a PLC with the privileges of an operator already authenticated on the controller. To address the vulnerability, Schneider Electric developed a new mechanism, Application Password, which should provide protection against unauthorized access to PLCs and unwanted modifications.

An analysis conducted by Kaspersky ICS CERT experts has shown that the implementation of the new security mechanism also has flaws. The [CVE-2021-22779](#) vulnerability, which was identified in the course of the research, could allow a remote attacker to make changes to the PLC, bypassing authentication.

It was established that the UMAS protocol, in its implementation prior to the version in which the CVE-2021-22779 vulnerability was fixed, had significant shortcomings that had a critical effect on the security of control systems based on SE controllers.

As of the middle of August 2022, Schneider Electric has released an update for the EcoStruxure™ Control Expert software, as well as for Modicon M340 which fixes the vulnerability. In March 2023, the vendor released an update for the Modicon M580 PLC.

This report describes:

- the implementation of the UMAS protocol that does not use the Application Password security mechanism;
- authentication bypass if Application Password is not enabled;
- the principles on which the Application Password security mechanism is based;
- mechanisms that can be used to exploit the CVE-2021-22779 vulnerability (authentication bypass where Application Password is configured).
- operating principles of the updated device reservation mechanism.

The conclusion provides remediations and recommendations from Schneider Electric on addressing the authentication bypass vulnerability, as well as Kaspersky ICS CERT recommendations.

If you would like to request additional information or to share your thoughts on issues discussed in this article, please write to ics-cert@kaspersky.com.

Snort rules are available on the [Kaspersky Threat Intelligence](#) (ICS Reporting) portal.

Object of research

UMAS (Unified Messaging Application Services) is Schneider Electric's proprietary protocol used to configure, monitor, collect data and control Schneider Electric industrial controllers.

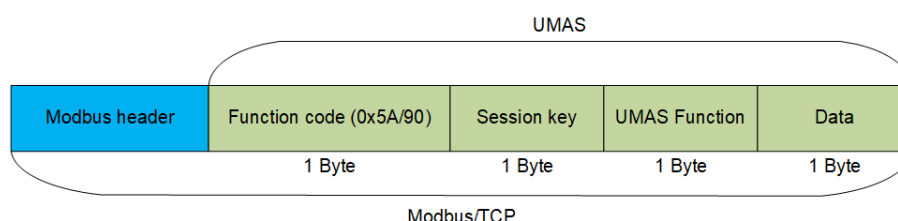
UMAS is based on a client-server architecture. In the process of our research, we used the EcoStruxure™ Control Expert PLC configuration software as the client part and a Modicon M340 CPU controller as the server part.

UMAS protocol

Network packet structure

UMAS is based on the Modbus/TCP protocol.

Structure
of the UMAS
protocol



Specifications of the Modbus/TCP protocol include reserved Function Code values that developers can use according to their needs. A complete list of reserved values can be found in the [official documentation](#).

Schneider Electric uses Function Code 90 (0x5A) to define that the value in the Data field is UMAS compliant.

The network packet structure is shown below, using a request to read a memory block (pu_ReadMemoryBlock) on the PLC as an example:

- Red: Function Code 90 (0x5A)
- Blue: Session Key 0 (0x00) (see [Session key](#))
- Green: UMAS Function 20 (0x20) (see [UMAS protocol functions](#))
- Orange: Data

Each function includes a certain set of information in Data, such as offset from the base memory address, size of the data sent, memory block number, etc.

112	2.233067	192.168.0.6	192.168.0.150	UMAS	73 [19]	UMAS: ReadMemoryBlock
114	2.245383	192.168.0.150	192.168.0.6	UMAS	579 [525]	UMAS: Response


```

> Frame 112: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF_{F658B7F2-24EC
> Ethernet II, Src: VMware_d5:a0:80 (00:0c:29:d5:a0:80), Dst: Telemech_25:c6:36 (00:80:f4:25:c6:36)
> Internet Protocol Version 4, Src: 192.168.0.6, Dst: 192.168.0.150
> Transmission Control Protocol, Src Port: 57651, Dst Port: 502, Seq: 2423, Ack: 5209, Len: 19
> Schneider UMAS Protocol
  Transaction id: 57589
  Protocol id: 0
  Data length: 13
  Unit id: 0
  Function: 90
  Connection id: 0
  Command: 0x20
  Sys Ram block number: 276
  Sys Ram address: 0
  Size: 0
  Data: 000002

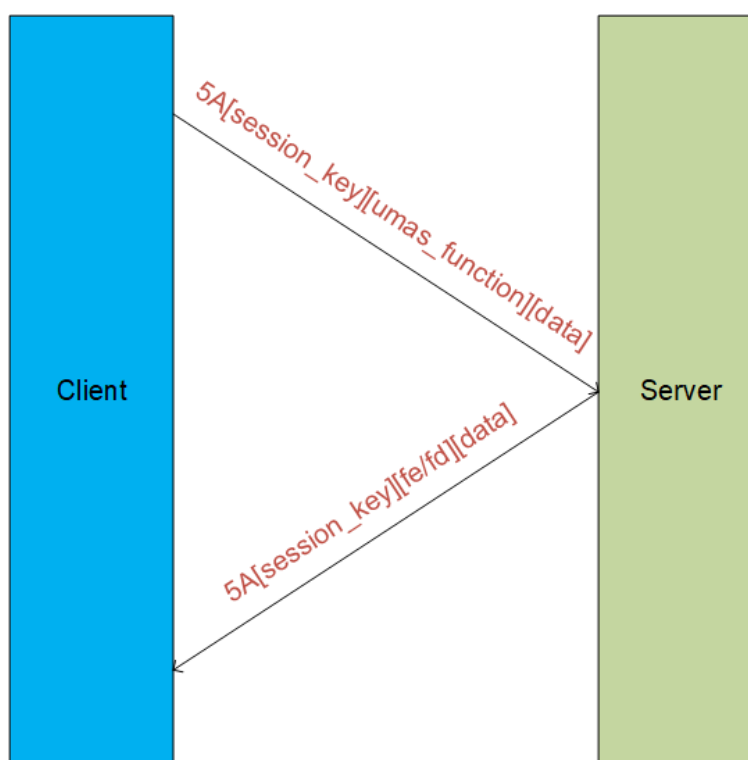
```


0000	00	80	f4	25	c6	36	00	0c	29	d5	a0	80	08	00	45	00
0010	00	3b	d2	f6	40	00	80	06	a5	d9	c0	a8	00	06	c0	a8
0020	00	96	e1	33	01	f6	71	2d	36	68	b7	64	58	0c	50	18
0030	fd	a8	b1	62	00	00	e0	f5	00	00	0d	00	5a	00	20	
0040	01	14	00	00	00	00	00	00	02							

Network communication

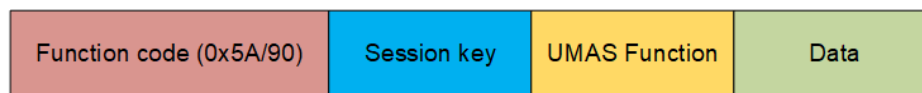
UMAS also inherits the Modbus client-server architecture. A structural diagram of the communication between the client and the server is provided below.

Communication
between
the client
(EcoStruxure™
Control Expert)
and server
(PLC)



UMAS network packet structure

In a UMAS network packet, Function Code 0x5A is immediately followed by the Session Key.



Below we examine the communication between a client and a server (a PLC, also referred to below as "device") by analyzing a sample real-world traffic fragment.

The screenshot below shows a packet containing the function **umas_QueryGetComInfo(0x01)** sent from the client (EcoStruxure™ Control Expert) to the server (the PLC).

Structure of the function:

TCP DATA - Modbus Header - 0x5A - session - 01 (UMAS function code) - 00 (data).

```
> Modbus/TCP
  Modbus
    .101 1010 = Function Code: Unity (Schneider) (90)
    Data: 000100
```

0000	00 80 f4 11 5e 23 00 0c 29 d5 a0 80 08 00 45 00	...	^#...)E
0010	00 33 08 9f 40 00 80 06 70 34 c0 a8 00 0b c0 a8	...	3-@...	p4
0020	00 96 c0 90 01 f6 7b 51 4d 19 da a3 d3 94 50 18	...	{Q MP
0030	fe 74 f5 c9 00 00 00 07 00 00 00 05 00 5a 00 01	...	tZ
0040	00

Annotations: Red arrow points to 5a (Modbus Function Code). Blue arrow points to 01 (UMAS Function code). Green arrow points to 00 (Session Key). Purple arrow points to 00 (Data).

The device should send a response to each request received. The screenshot below shows the device's response to the client's request:

```
> Modbus/TCP
  Modbus
    .101 1010 = Function Code: Unity (Schneider) (90)
    [Request Frame: 14]
    [Time from request: 0.010453000 seconds]
    Data: 00fefd030006000032000000000000
```

0000	00 0c 29 d5 a0 80 00 80 f4 25 c6 36 08 00 45 00
0010	00 3f 02 3a 40 00 40 06 b6 92 c0 a8 00 96 c0 a8
0020	00 06 01 f6 e1 33 b7 64 43 f2 71 2d 2d 07 50 18
0030	22 08 7d bd 00 00 e0 da 00 00 00 11 00 5a 00 fe
0040	fd 03 00 06 00 00 32 00 00 00 00 00 00 00 00

Annotation: Red box highlights 'fe' (Status code).

The status code is the status of execution by the device of the function sent to it by the client in the previous request. The value "fe" corresponds to successful execution of the function, "fd" – to an error. These values are present in each response sent by the device to the client's request containing a function. The status code is always located immediately after the session key.

Reservation procedure

A "reservation" procedure is required to make changes to a PLC. The procedure acts as authentication.

Only one client (e.g., an engineering workstation) can reserve a device at any specific time for configuration or status monitoring. This is required to prevent changes from being made to a device in parallel without coordination.

The screenshot below shows a request from the engineering software to the PLC to perform the device reservation procedure in its basic variant that does not use the Application Password security mechanism.

The **umas_QueryTakePLCReservation(0x10)** function is used to reserve a device.

To reserve a device, the client sends a request containing the 0x10 function to the device. The request includes the name of the client reserving the device and the value equal to the length of that name.

▼ Modbus

.101 1010 = Function Code: Unity (Schneider) (90)
Data: 0010712a00000f4445534b544f502d344e4f39565542

0000	00 80 f4 25 c6 36 00 0c 29 d5 a0 80 08 00 45 00	...%-6...).....E
0010	00 46 e0 1d 40 00 80 06 98 a7 c0 a8 00 06 c0 38	-F-@-...
0020	00 96 cf 71 01 f6 46 e4 b3 21 f3 4e d5 02 50 18	...q--F- -I-N--P-
0030	fe 63 e0 58 00 00 50 41 00 00 00 18 00 5a 00 10	-c-X--PAZ-
0040	71 2a 00 00 0f 44 45 53 4b 54 4f 50 2d 34 4e 4f	q*...-DES KTOP-4NO
0050	39 56 55 42	9VUB

Annotations:

- Session** (green arrow) points to the function code 10 (0x10) in the data field.
- Client name length** (red arrow) points to the value 0f (0x0f) in the data field.
- Client name** (blue arrow) points to the data field starting from 44 (0x44) to 4f (0x4f).
- Reservation function** (pink arrow) points to the value 10 (0x10) in the data field.

Session key

Upon completing the reservation, the device sends the value of the new one-byte session key to the client. The key is subsequently used to authorize device modification requests.

With the release of new firmware versions, the session creation mechanism has undergone some changes, namely:

- In firmware versions prior to 2.7, the session key for a Modicon M340 device after its reservation had the fixed value 0x01;
- In firmware versions 2.7 or later, the session key for a Modicon M340 device had a random value, i.e., from 0 to 0xFF, as the session key is 1 byte in size.

Until the reservation procedure is completed, a service session with the value "0x00" is used. Functions that do not require reservation can be executed in that session.

The device's response, which includes the status code (0xfe) and the **new session key**, looks as follows:

```

Modbus
.101 1010 = Function Code: Unity (Schneider) (90)
[Request Frame: 72]
[Time from request: 0.002688000 seconds]
Data: 00fe01

```

Offset	Hex	ASCII
0000	00 0c 29 d5 a0 80 00 80 f4 25 c6 36 08 00 45 00	..). -%.6..E.
0010	00 33 00 2c 40 00 40 06 b8 ac c0 a8 00 96 c0 a8	-3.,@.@.
0020	00 06 01 f6 cf 71 f3 4e d5 02 46 e4 b3 3f 50 18q-N ..F..?P.
0030	22 08 25 51 00 00 50 41 00 00 00 05 00 5a 00 fe	"-%Q..PAZ..
0040	01 04	

The status code "fe" means that the reservation procedure was successful.

In this case, the device sends the new session key value. The new session key is used in all subsequent requests during the current "reserved" session.

The following screenshot shows a request from a client to the device using the new session key immediately after the device's successful reservation. In this example, the request uses the **ex_GetPlcStatus(0x04)** function.

```

> Modbus/TCP
Modbus
.101 1010 = Function Code: Unity (Schneider) (90)
Data: 0104

```

Offset	Hex	ASCII
0000	00 80 f4 25 c6 36 00 0c 29 d5 a0 80 08 00 45 00	
0010	00 32 d2 e6 40 00 80 06 a5 f2 c0 a8 00 06 c0 a8	
0020	00 96 e1 33 01 f6 71 2d 35 b8 b7 64 55 2f 50 18	
0030	fe 58 b6 8e 00 00 e0 e8 00 00 00 04 00 5a 01 04	

The reservation procedure plays the role of authentication when making changes to a device. This means that the mechanism is critically important from the security viewpoint.

Issues related to reserving a device in the default configuration and using security features are covered in the following sections.

UMAS protocol functions

The UMAS protocol has numerous functions for communicating with target devices. Functions can be divided into two groups:

1. Functions that require reserving the device. As a rule, these are functions that make changes to the PLC.
2. Functions that do not require device reservation. Such functions do not make any changes to the PLC and do not affect its operation.

An abbreviated list of functions available in the UMAS protocol is shown below. Information on the need to reserve the device for the functions listed below is relevant to firmware version 3.30 for Modicon M340 devices without the use of the [Application Password](#) security mechanism.

Functions used in the device reservation process

1. 0x10 – umas_QueryTakePLCReservation – reserves the device.
2. 0x11 – umas_QueryReleasePLCReservation – releases the device from reservation.
3. 0x12 – umas_QueryKeepPLCReservation – reservation status.

Functions that require device reservation

Initialization functions

0x01 – umas_QueryGetComInfo – UMAS message initialization.

Functions used to request information about a device

1. 0x02 – pu_GetPlcInfo – requests information about the device
2. 0x04 – pu_GetPlcStatus – queries the PLC status
3. 0x06 – pu_GetMemoryCardInfo – requests information about the device's SD card

Functions for downloading and uploading PLC strategies

A strategy is a set of instructions and data used by the PLC to perform its main function, that is, to control terminal equipment, e.g., to automate a certain industrial process.

1. 0x30 – pumem_BeginDownload – initializes an upload from the PC to the PLC.
2. 0x31 – pumem_DownloadPacket – uploads a strategy block from the PC to the PLC.
3. 0x32 – pumem_EndDownload – ends the upload from the PC to the PLC.

4. 0x33 – pumem_BeginUpload – initializes a download from the PLC to the PC.
5. 0x34 – pumem_UploadPacket – downloads a strategy block from the PLC to the PC.
6. 0x35 – pumem_EndUpload – ends the download from the PLC to the PC.

Functions that do not require device reservation

Function that reads information from device memory

0x20 – pu_ReadMemoryBlock – reads PLC memory block.

Function that writes values to device memory

0x21 – pu_WriteMemoryBlock – writes PLC memory block.

Functions that control the state of the PLC

The following functions can be used to start or suspend the operation of the PLC. These functions do not require reservation if the [Application Password](#) security mechanism is not activated, in which case the device will successfully handle a request using a service session (0x00) (see [Session key](#)).

Unless the Application Password setting is enabled, an attacker can use these functions to stop the PLC, thereby causing significant damage to the industrial process.

1. 0x40 – ex_StartTask – Start PLC operation.
2. 0x41 – ex_StopTask – Stop PLC operation.

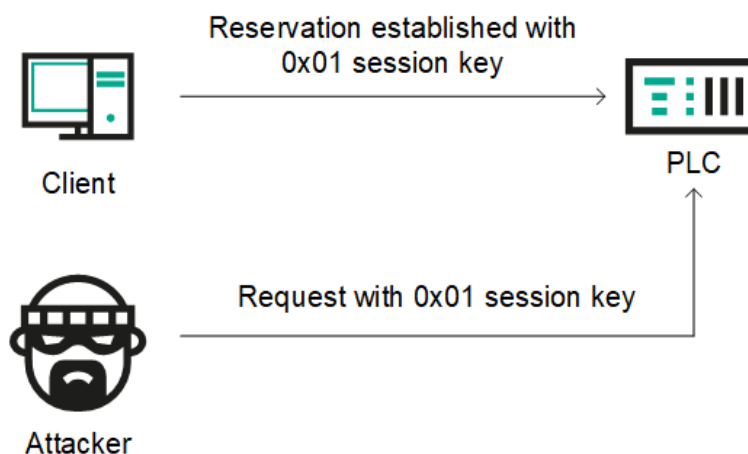
CVE-2020-28212: authentication bypass without Application Password

The main issue with the basic reservation mechanism that does not use Application Password is that an attacker can use the session key to send requests and change the device's configuration.

In firmware versions prior to 2.7 for Modicon M340 devices, the session key has the same value each time the device is reserved and is equal to "0x01". This means that attackers can make changes on the device by calling the relevant functions after reserving the device themselves or after the device has been reserved by a legitimate user.

The attack workflow is shown in the diagram below:

Remote threat actor attack workflow. Modicon M340 firmware version prior to 2.7, device reserved by an engineer



If the device has not been reserved at the time of an attack, the attacker can use the **umas_QueryTakePLCReservation(0x10)** function to reserve the device in order to make changes to it.

With Modicon M340 firmware versions 2.7 or later, the session key takes a random value after device reservation. However, the session key is one byte in length, which means there are only 256 possible session ID values. This enables a remote unauthorized attacker to brute-force an existing ID of the session between a legitimate user and the PLC.

To carry out this type of attack, a remote attacker needs to send a series of network requests on port 502/TCP of the PLC with different session ID values and look at responses returned by the PLC. If the correct session ID was sent, the attacker will get status code `0xfe`, which means that the request was fulfilled successfully. Otherwise, the attacker will get status code `0xfd`.

The operations described above can be implemented using any programming language – an attacker does not have to use EcoStruxure™ Control Expert or any other dedicated software to communicate with the device.

Application Password

To mitigate the [CVE-2020-28212](#) vulnerability, the exploitation of which could allow a remote unauthorized attacker to gain control of the PLC with the privileges of an operator already authenticated on the PLC, Schneider Electric developed a new security mechanism. Schneider Electric believed that implementing an improved security mechanism that used cryptographic algorithms to compute the session ID and increasing the session ID length would prevent brute-force attacks that could be used to crack single-byte session IDs.

Starting with firmware version 3.01 for Modicon M340 devices, Schneider Electric actively developed security mechanisms to prevent attackers from abusing UMAS functions to make changes to device operation. To implement authentication between the client and the device, Application Password should be enabled in project settings ("Project & Controller Protection"). The mechanism is designed to provide protection against unauthorized access, unwanted changes, as well as unauthorized downloading or uploading of PLC strategies.

After activating the mechanism using EcoStruxure™ Control Expert, the client needs to enter the password when connecting to a device as part of the reservation procedure.

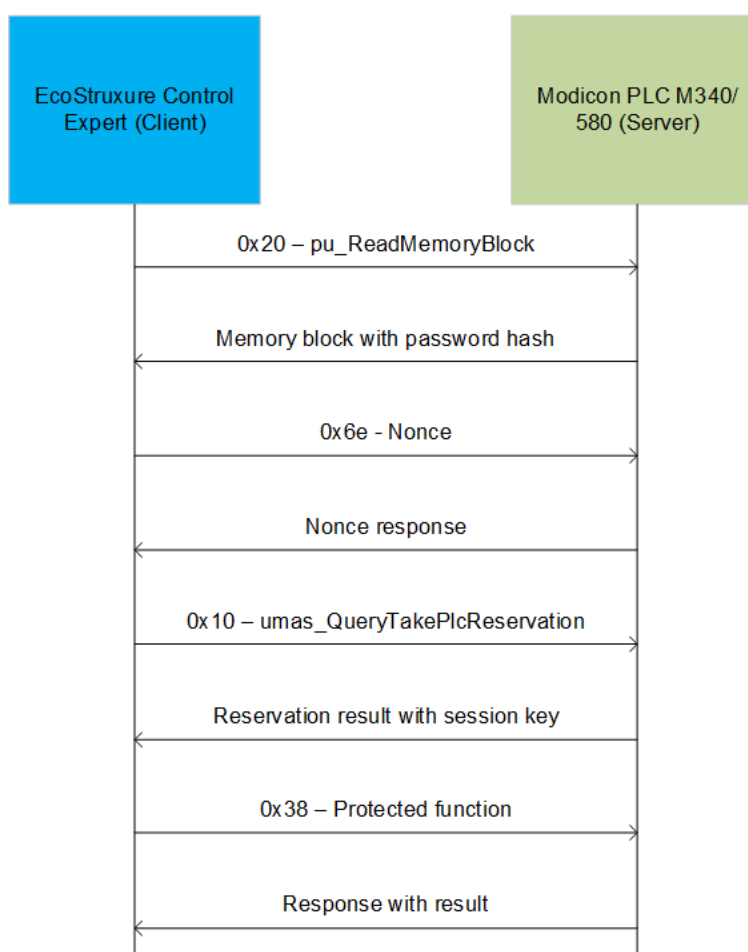
Application Password also makes changes to the reservation mechanism itself. These changes are discussed in the sections below.

Authentication bypass with Application Password

Unfortunately, an analysis conducted by Kaspersky ICS CERT experts has shown that the implementation of the new security mechanism also has flaws. The vulnerability identified during the research, [CVE-2021-22779](#), could allow a remote attacker to bypass the authentication mechanism and use functions that require reservation to make changes to the PLC.

To grasp more fully what the shortcomings of the 'improved' security mechanism are, let's look at the authentication and PLC reservation process in greater detail. The new security mechanism is based on exchanging a randomly generated byte sequence (a nonce) between the client and the server and subsequently producing a single session secret. The diagram below shows the sequence of requests sent and responses received.

Device reservation sequence with Application Password enabled



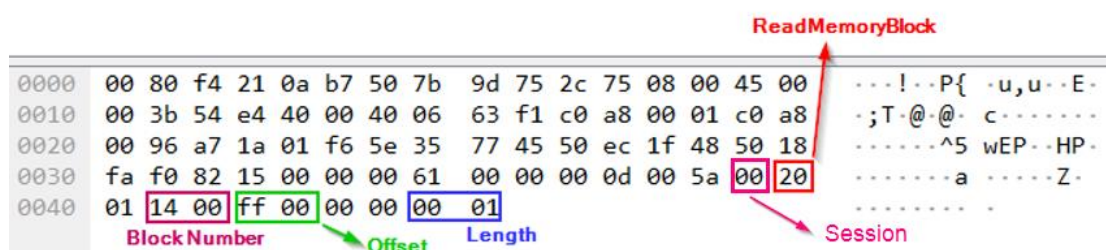
Below we look at the process in more detail.

After establishing a TCP session, the EcoStruxure™ Control Expert software sends a request to the PLC (port 502/TCP) to read a memory block using UMAS function 0x20, which does not require authentication.

Modbus

.101 1010 = Function Code: Unity (Schneider) (90)

Data: 0020011400ff0000000001



Next, the client receives a response from the PLC. The memory block is needed for further computation since it contains two base64 strings that make up a password hash.

Modbus		
.101 1010 = Function Code: Unity (Schneider) (90)		
[Request Frame: 1]		
[Time from request: 0.011671495 seconds]		
Data: 00fe0100010051510000004347493355534f3949004330696e306a38414c78553d0d0a31...		
0000	50 7b 9d 75 2c 75 00 80 f4 21 0a b7 08 00 45 00	P{·u,u· ·!·...·E·
0010	01 35 1d a6 40 00 40 06 9a 35 c0 a8 00 96 c0 a8	·5··@·@· ·5·...·
0020	00 01 01 f6 a7 1a 50 ec 1f 48 5e 35 77 58 50 18	·...·P· ·H^5wXP·
0030	22 08 b2 73 00 00 00 61 00 00 01 07 00 5a 00 fe	"·s·...a· ·...·Z·
0040	01 00 01 00 51 51 00 00 00 43 47 49 33 55 53 4f	·...·QQ· ·CGI3US0
0050	39 49 00 43 30 69 6e 30 6a 38 41 4c 78 55 3d 0d	9I·C0in0 j8ALxU=·
0060	0a 31 4f 6d 5a 42 33 31 77 57 57 6c 6c 67 47 45	·10mZB31 wWl1gGE
0070	4b 2f 75 36 45 43 7a 66 6f 39 48 55 76 59 69 4e	K/u6ECzf o9HUVYiN
0080	44 6c 6a 2b 73 59 77 77 71 74 47 38 3d 0d 0a 00	Dlj+sYww qtG8=·
0090	00 00 56 31 31 2e 30 00 00 00 57 49 4e 2d 46 51	·V11.0· ·WIN-FQ
00a0	49 52 37 51 54 38 31 4b 49 00 00 00 00 00 00 00	IR7QT81K I·...·
00b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	·...·

After this, EcoStruxure™ Control Expert generates a random byte sequence (a nonce) that is 32 bytes in length and sends it to the PLC.

Modbus		
.101 1010 = Function Code: Unity (Schneider) (90)		
Data: 006e02372a0000e762eddd784b61f764cafa84afc67c5b404f6210a8ed312b8294c48996...		
0000	00 80 f4 21 0a b7 50 7b 9d 75 2c 75 08 00 45 00	·...!·P{·u,u··E·
0010	00 59 be a5 40 00 40 06 fa 11 c0 a8 00 01 c0 a8	·Y··@·@· ·...·
0020	00 96 a7 1c 01 f6 15 28 99 e3 f5 d6 7d 67 50 18	·...·(· ·...·}gP·
0030	fa f0 82 33 00 00 00 07 00 00 00 2b 00 5a 00 6e	·...3·...·+·Z·n
0040	02 37 2a 00 00 e7 62 ed dd 78 4b 61 f7 64 ca fa	·7*·...·b· ·xKa·d·
0050	84 af c6 7c 5b 40 4f 62 10 a8 ed 31 2b 82 94 c4	·... [@0b ·...1+·
0060	89 96 6f d4 26 00 00	·o·&·

In response to receiving the nonce, the PLC also sends a byte sequence (a nonce response) to EcoStruxure™ Control Expert, which is also 32 bytes long.

Modbus		
.101 1010 = Function Code: Unity (Schneider) (90)		
[Request Frame: 3]		
[Time from request: 0.009112150 seconds]		
Data: 00feaaaa25e423b15ba05f47c0b53cdf47868e4e33e3b0a110fd0d812231a5a85f7e9397		
0000	50 7b 9d 75 2c 75 00 80 f4 21 0a b7 08 00 45 00	P{·u,u· ·!·...·E·
0010	00 54 1d ac 40 00 40 06 9b 10 c0 a8 00 96 c0 a8	·T··@·@· ·...·
0020	00 01 01 f6 a7 1c f5 d6 7d 67 15 28 9a 14 50 18	·...·}g·(·P·
0030	22 08 fe 19 00 00 00 07 00 00 00 26 00 5a 00 fe	"·...·&·Z·
0040	aa aa 25 e4 23 b1 5b a0 5f 47 c0 b5 3c df 47 86	·%·#·[· _G·<·G·
0050	8e 4e 33 e3 b0 a1 10 fd 0d 81 22 31 a5 a8 5f 7e	·N3·...· ·"1·~
0060	93 97	·

Importantly, the nonce response computed on the PLC side depends only on the byte sequence received from EcoStruxure™ Control Expert (the nonce) and no additional random element is used to generate the response. In other words, the same response is always sent to the same nonce.

In the next step, EcoStruxure™ Control Expert uses the same nonce and response to compute the SHA 256 hash, which is required to reserve the PLC.

The hash is computed as follows:

```
SHA256 (PLC nonce response + base64 strings (password hash) from PLC's
memory block + EcoStruxure Control Expert nonce)
```

Using the data from the above examples, the hash computation looks as follows:

```
SHA256 ("\\x25\\xe4\\x23\\xb1\\x5b\\xa0\\x5f\\x47\\xc0\\xb5\\x3c\\xdf\\x47\\x86\\x8e\\
x4e\\x33\\xe3\\xb0\\xa1\\x10\\xfd\\x0d\\x81\\x22\\x31\\xa5\\xa8\\x5f\\x7e\\x93\\x97"
+ "\\x43\\x47\\x49\\x33\\x55\\x53\\x4f\\x39\\x49\\x00\\x43\\x30\\x69\\x6e\\x30\\x6a\\x
38\\x41\\x4c\\x78\\x55\\x3d\\x0d\\x0a\\x31\\x4f\\x6d\\x5a\\x42\\x33\\x31\\x77\\x57\\x5
7\\x6c\\x6c\\x67\\x47\\x45\\x4b\\x2f\\x75\\x36\\x45\\x43\\x7a\\x66\\x6f\\x39\\x48\\x55
\\x76\\x59\\x69\\x4e\\x44\\x6c\\x6a\\x2b\\x73\\x59\\x77\\x77\\x71\\x74\\x47\\x38\\x3d"
+ "\\xe7\\x62\\xed\\xdd\\x78\\x4b\\x61\\xf7\\x64\\xca\\xfa\\x84\\xaf\\xc6\\x7c\\x5b\\x
40\\x4f\\x62\\x10\\xa8\\xed\\x31\\x2b\\x82\\x94\\xc4\\x89\\x96\\x6f\\xd4\\x26") = 1b
c23b84e0989643965ef082869d17d5a8398b82fbc8e2775419a8a807f5fe04
```

Ultimately, PLC reservation is performed using the ASCII representation of the computer's name and the SHA256 hash computed earlier.

```
PWIN-FQIR7QT81KI + '\\x00' + 1bc23b84e0989643965ef082869d17d5a8398b82f
bc8e2775419a8a807f5fe04
```

Modbus	
.101 1010 = Function Code: Unity (Schneider) (90)	
Data: 0010372a00005057494e2d4651495237515438314b490031626332336238346530393839...	
Computer Name	
0000	00 80 f4 21 0a b7 50 7b 9d 75 2c 75 08 00 45 00 ...!..P{ .u,u..E.
0010	00 87 be a7 40 00 40 06 f9 e1 c0 a8 00 01 c0 a8@.@.
0020	00 96 a7 1c 01 f6 15 28 9a 14 f5 d6 7d 93 50 18(....}.P.
0030	fa c4 82 61 00 00 00 08 00 00 00 59 00 5a 00 10 ...a.... ..Y.Z..
0040	37 2a 00 00 50 57 49 4e 2d 46 51 49 52 37 51 54 7*..PWIN -FQIR7QT
0050	38 31 4b 49 00 31 62 63 32 33 62 38 34 65 30 39 81KI.1bc 23b84e09
0060	38 39 36 34 33 39 36 35 65 66 30 38 32 38 36 39 89643965 ef082869
0070	64 31 37 64 35 61 38 33 39 38 62 38 32 66 62 63 d17d5a83 98b82fbc
0080	38 65 32 37 37 35 34 31 39 61 38 61 38 30 37 66 8e277541 9a8a807f
0090	35 66 65 30 34 5fe04
SHA256	

If the request is successfully fulfilled, the PLC will return session ID (0xf8 on the screenshot below) to EcoStruxure™ Control Expert.

▼ Modbus		
.101 1010 = Function Code: Unity (Schneider) (90)		
[Request Frame: 5]		
[Time from request: 0.004372910 seconds]		
Data: 00fef8		
0000	50 7b 9d 75 2c 75 00 80 f4 21 0a b7 08 00 45 00	P{·u,u·· !····E·
0010	00 33 1d ad 40 00 40 06 9b 30 c0 a8 00 96 c0 a8	-3··@·@· -0·····
0020	00 01 01 f6 a7 1c f5 d6 7d 93 15 28 9a 73 50 18	····· }··(·sP·
0030	22 08 46 53 00 00 00 08 00 00 00 05 00 5a 00 fe	"·FS···· ····Z·
0040	f8	·

This session ID will subsequently be used to send Security function (0x38) protected commands to the PLC.

▼ Modbus		
.101 1010 = Function Code: Unity (Schneider) (90)		
Data: f83801086c0a7e894ace8ddad9a6b80508c509a3fdf288fe551d40dab95b67d1d683d65a...		
0000	00 80 f4 21 0a b7 50 7b 9d 75 2c 75 08 00 45 00	···!··P{·u,u··E·
0010	00 58 be a9 40 00 40 06 fa 0e c0 a8 00 01 c0 a8	-X··@·@· ······
0020	00 96 a7 1c 01 f6 15 28 9a 73 f5 d6 7d 9e 50 18	·····(·s··}··P·
0030	fa b9 82 32 00 00 00 09 00 00 00 2a 00 5a f8 38	···2···· ····*·Z·8
0040	01 08 6c 0a 7e 89 4a ce 8d da d9 a6 b8 05 08 c5	··1·~·J· ······
0050	09 a3 fd f2 88 fe 55 1d 40 da b9 5b 67 d1 d6 83	·····U· @··[g···
0060	d6 5a f8 41 ff 00	-Z-A··

It can be seen from the above analysis of the PLC reservation process using the new and improved mechanism that the new method is by no means secure since all computation is performed on the client side (i.e., by EcoStruxure™ Control Expert), while the "secret" can be obtained from the PLC without authentication.

The fact that the PLC always sends the same response to the same nonce received from the client is an additional shortcoming of the mechanism, which enables an attacker to carry out a Replay attack using network traffic between a legitimate client (the operator) and the server (the PLC) captured earlier in the process of PLC reservation.

Updated reservation procedure with Application Password

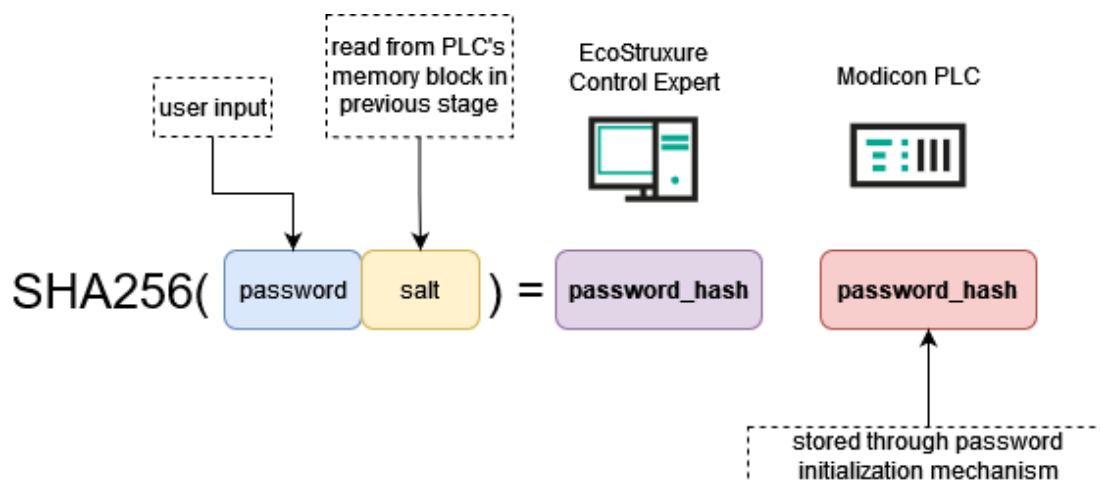
At the time of publication, Schneider Electric had released updates for EcoStruxure™ Control Expert software (version 15.1), Modicon M340 PLC firmware (version 3.50), and Modicon M580 PLC firmware (version 4.10). These updates fix the vulnerability described in the [Authentication bypass with Application Password](#) section.

This section describes the updated reservation procedure used after the vendor had fixed the vulnerability.

During the PLC reservation procedure, a total of 0x534 bytes are read from memory block 0x14 in two requests, using UMAS function `pu_ReadMemoryBlock(0x20)`, which does not require authentication. In the earlier version of the reservation mechanism, that memory segment contained the password hash, but in the new version it contains a salt and a ciphertext.

0120	00 00 00 00 00 00 00 00 00 00 00 00 00 4b 6c 4esalt.....KIN
0130	33 6e 49 4c 4c 4e 4a 63 3d 0d 0a 0d 0a 00 00 00	3nILLN3c =
0140	61 54 70 70 62 35 74 6c 74 6a 49 54 41 44 55 55	aTppb5t1 tjITADUU
0150	65 77 77 71 4e 76 6e 36 78 63 66 37 32 41 59 57	ewwqNvn6 xcf72AYW
0160	5a 45 30 69 56 79 58 61 32 43 63 47 6d 4d 7a 34	ZE0iVyXa 2CcGmMz4
0170	42 46 65 69 4e 32 62 6f 31 37 32 6f 6b 54 7a 7a	BFeiN2bo 172okTzz
0180	0d 0a 6e 37 54 79 38 48 36 53 51 35 71 50 6d 51	..n7Ty8H 6SQ5qPmQ
0190	56 58 4f 55 4c 42 34 71 56 55 35 76 61 4c 79 57	VXOULB4q VU5vaLyW
01a0	63 51 51 7a 30 36 6a 79 6f 6d 36 74 49 6f 4a 67	cQQz06jy om6tIoJg
01b0	78 63 6d 31 6f 79 6e 2b 39 30 6d 76 7a 62 4c 30	xcm1oyn+ 90mvzbl0
01c0	58 59 0d 0a 4d 33 79 68 74 31 67 61 43 2f 67 35	XY..M3yh t1gaC/g5
01d0	2f 41 6e 42 72 35 61 6f 52 44 45 71 54 37 37 30	/AnBr5ao RDEqT770
01e0	61 44 37 38 4c 6d 72 79 6f 79 76 31 6f 69 4a 4e	aD78Lmry oyy1oiJN
01f0	54 31 38 4f 66 6c 73 54 38 65 51 6e 66 78 30 34	T180f1sT 8eQnfx04
0200	76 49 34 62 0d 0a 66 6b 2b 34 63 42 59 44 2f 6b	vI4b..fk +4cBYD/k
0210	50 56 6a 66 36 33 53 39 69 65 54 54 50 6e 79 77	PVjf63S9 ieTTPnyw
0220	64 30 65 51 70 6d 7a 67 53 49 46 71 79 50 5a 34	d0eQpmzg SIFqyPZ4
0230	67 75 51 33 2b 38 6f 2f 53 59 58 65 63 4e 71 32	guQ3+8o/ SYXecNq2
0240	51 33 63 4e 33 4f 0d 0a 45 4c 64 45 47 79 35 65	Q3cN30.. ELdEGy5e
0250	31 36 34 46 69 6b 39 77 6a 75 6d 31 6b 51 58 6b	164Fik9w jum1kQXk
0260	42 53 76 56 4a 43 2b 6e 75 36 4a 37 6d 72 69 38	BSvVJC+n u6J7mri8
0270	61 43 69 78 53 39 6f 4d 73 68 6d 33 64 2b 32 70	aCixS9om shm3d+2p
0280	5a 57 6b 4e 39 4a 52 52 0d 0a 67 64 71 54 35 75	ZWkN9JRR ..gdqT5u
0290	50 64 6d 4c 6e 48 31 51 7a 65 79 48 4d 58 7a 77	PdmLnH1Q zeyHMXzw
02a0	3d 3d 0d 0a 00 00 00 00 00 00 00 00 00 00 00	=====
02b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

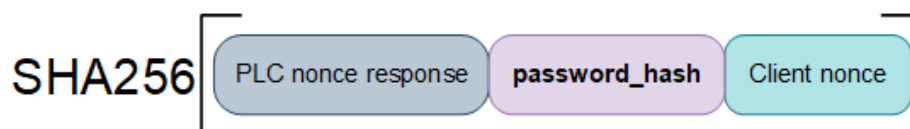
When the client (EcoStruxure™ Control Expert) has obtained the salt value for hashing, it can start the PLC reservation procedure. The client computes the SHA256 hash of the password entered by the user with the salt obtained at the previous step to compute the password hash.



The next step is the nonce exchange between the client and the PLC.



In the final step, the client computes the SHA256 hash that is to be used to reserve the device. The hash is computed from the PLC nonce, the password hash and the client nonce.



The screenshot below shows a request from the client (EcoStruxure™ Control Expert) to the PLC to reserve the device using the SHA256 hash computed in the final step.

Modbus			
.101 1010 = Function Code: Unity (Schneider) (90)			
Data: 0010f03d0000495245534541524348006265303163363531613236386465343632646236...			
0000	00 80 f4 21 0a b0 04 42	1a 85 ca 78 08 00 45 00	...!...B...x...E..
0010	00 80 09 5e 40 00 80 06	00 00 c0 a8 00 7d c0 a8	...^@...
0020	00 96 0c a7 01 f6 72 24	26 43 32 9c 48 1c 50 18r\$ &C2·H·P·
0030	fe 12 82 d6 00 00 00 11	00 00 00 52 00 5a 00 10R·Z..
0040	f0 3d 00 00 49 52 45 53	45 41 52 43 48 00 62 65	...IRES EARCH be
0050	30 31 63 36 35 31 61 32	36 38 64 65 34 36 32 64	01c651a2 68de462d
0060	62 36 32 63 32 63 37 33	34 64 30 35 31 31 32 65	b62c2c73 4d05112e
0070	39 31 39 30 37 35 30 62	39 36 37 30 34 38 30 30	9190750b 96704800
0080	64 62 36 38 66 65 64 38	66 30 61 35 32 63	db68fed8 f0a52c

SHA256

In the earlier version of the reservation mechanism, the main problem was that the “secret” used to reserve the device was computed entirely on the client side (i.e., by EcoStruxure™ Control Expert). In the corrected implementation of the mechanism, memory block 0x14 of the PLC does not contain the password hash used to compute the “secret”, i.e., the final SHA256 hash.

Conclusion

Our analysis shows that the implementation of the UMAS protocol in Modicon M340 firmware versions prior to 3.50 had significant shortcomings that critically affected the security of automation systems based on Schneider Electric solutions.

According to shodan.io data, the number of Modicon M340/M580 devices available online is greater than 1000. This is obviously just the tip of the iceberg.

Vendor recommendations

The vendor recommends following the remediation provided for EcoStruxure™ Control Expert in the [SEVD-2021-194-01](#) security advisory and using the Application Password mechanism to ensure complete remediation of this issue.

Kaspersky ICS CERT recommendations

In addition to the recommendations provided by the vendor, Kaspersky ICS CERT strongly recommends monitoring critical UMAS functions at traffic level, for example, using IDS or dedicated solutions for monitoring industrial network traffic, identifying and analyzing network anomalies. It is obvious that such functions as device reservation, stopping the device or downloading/uploading strategies are critically important and can be abused by an attacker to disrupt the industrial process.

Kaspersky Industrial Control Systems Cyber Emergency Response Team (Kaspersky ICS CERT)

is a global project of Kaspersky aimed at coordinating the efforts of automation system vendors, industrial facility owners and operators, and IT security researchers to protect industrial enterprises from cyberattacks. Kaspersky ICS CERT devotes its efforts primarily to identifying potential and existing threats that target industrial automation systems and the industrial internet of things.

[Kaspersky ICS CERT](#)

ics-cert@kaspersky.com