Security research: ThingsPro Suite – IIoT gateway and device manager by Moxa

Alexander Nochvay

22.01.2019 Version 1.0

Contents

Analyzing the security of technologies that are included in the toolset of automation system developers and have the potential for being used at industrial facilities across the globe is a high-priority area of work for Kaspersky Lab Industrial Control Systems Cyber Emergency Response Team (Kaspersky Lab ICS CERT).

It is obvious that the security of products that are part of the industrial internet of things (IIoT) ecosystem requires special attention. This time, our research focused on ThingsPro Suite – an IIoT gateway and device manager by Moxa.

An important argument in favor of our choice was the fact that using ThingsPro Suite requires the solution to be remotely available over the internet, because ThingsPro Suite is designed for Moxa's UC-8100 Series industrial gateway computers and its platform is controlled via a web interface. In addition, when using ThingsPro Suite, most devices transfer data either directly via the platform or through an intermediary, such as another gateway.

This means that ThingsPro Suite is an exit point from the industrial network to the internet and, conversely, an entry point from the internet into the industrial network.

Object description

The opportunity to improve the enterprise's efficiency is the main argument in favor of implementing industrial internet of things technologies in various industrial sectors, agriculture, transport, logistics, city infrastructure management, utilities, and energy sectors. The prospects opened by these technologies to company management are indeed attractive. They include centralized monitoring of equipment operating at remote facilities, which makes it possible to assess the devices' wear and tear and, therefore, to take timely measures to prevent their failure. It also includes the ability to optimize the operation of remote industrial facilities – preventing down time, using resources more efficiently, improving quality, and increasing the output.

To implement these and many other attractive features, the modern concept of the industrial internet of things involves collecting data from industrial information systems – PLCs, SCADA, OPC servers and directly from 'smart' field devices (sensors and actuators) and sending that data to a remote system for centralized processing and analysis. This usually involves using cloud systems (hence the word 'internet' in the name of the concept). Essentially, this means limited integration of OT (operational technology) systems with remote IT systems that are available over the internet. To simplify and accelerate such integration, Moxa has developed a dedicated solution, ThingsPro Suite.

ThingsPro Suite is positioned as an off-the-shelf solution designed to collect data from OT components and transfer it to a third-party cloud, and as a platform for developing in-house solutions for preprocessing and analyzing the data collected.

ThingsPro Suite includes such useful functionality as collecting data over Modbus TCP/RTU protocols and transferring it to the cloud over the MQTT protocol, including via 4G networks. It also provides integration with popular cloud platforms – MS Azure and Amazon. System administrators get convenient functionality related to monitoring the state of ThingsPro Gateway installations, configuring and updating them, with support for displaying them on a map. ThingsPro Suite implements a role model for access and supports some security features, such as TLS v1.2 – encryption and VPN access.

Application developers have a C/Python API at their disposal for preprocessing and filtering data, as well as a RESTful API for accessing the built-in functions of ThingsPro Suite.

ThingsPro Suite is designed to be installed on Moxa UC-8100 Series industrial computers.

Moxa first released ThingsPro Suite in 2017 and continues to release updates for the solution. We analyzed ThingsPro Suite 2.1 Build 17072504.

Research conditions

After studying the documentation, setting up the demo stand and familiarizing ourselves with the ThingsPro Suite application, we allocated two weeks for preliminary research to be done by one expert.

The solution was analyzed from the perspective of a remote attacker, since this is the most common and likely type of attacker. Consequently, we began our research by analyzing the security of the web service, since it is the most likely target for an attack. The analysis focused on the web administration application, which uses ports 80(HTTP) and 443(HTTPS).

It is also worth noting that our research was based on the "black box" method, i.e., it was conducted:

- Without consulting the Moxa development team;
- Without using non-public documentation;
- Without analyzing the source code.

The preliminary stage of the research yielded the results described below.

Research results

Over the two weeks of research, seven vulnerabilities were identified, some of which are critical: their exploitation results in the execution of arbitrary code in the Linux system. In addition, exploiting the vulnerabilities identified in combination allows remote attackers to take complete control of the device and elevate privileges in the system to a maximum level without initially having access to any authentication data.

It should be noted that Moxa has released a new version of the product – ThingsPro Suite 2.3 – to fix the vulnerabilities described below. For more detailed information, please refer to Moxa's security advisory.

The following IDs were assigned to the vulnerabilities identified:

KLI D	CVE	Brief vulnerability description
KLCERT-18-018	CVE-2018-18390	User enumeration
KLCERT-18-019	CVE-2018-18391	User privilege escalation
KLCERT-18-020	CVE-2018-18392	Broken access control
KLCERT-18-021	CVE-2018-18393	The server side does not require the old password when changing the old password
KLCERT-18-022	CVE-2018-18394	Cleartext storage of sensitive information
KLCERT-18-023	CVE-2018-18395	Privilege escalation (hidden token) - 1 backdoor (10 CVSS)
KLCERT-18-024	CVE-2018-18396	Remote code execution - 1 RCE (10 CVSS)

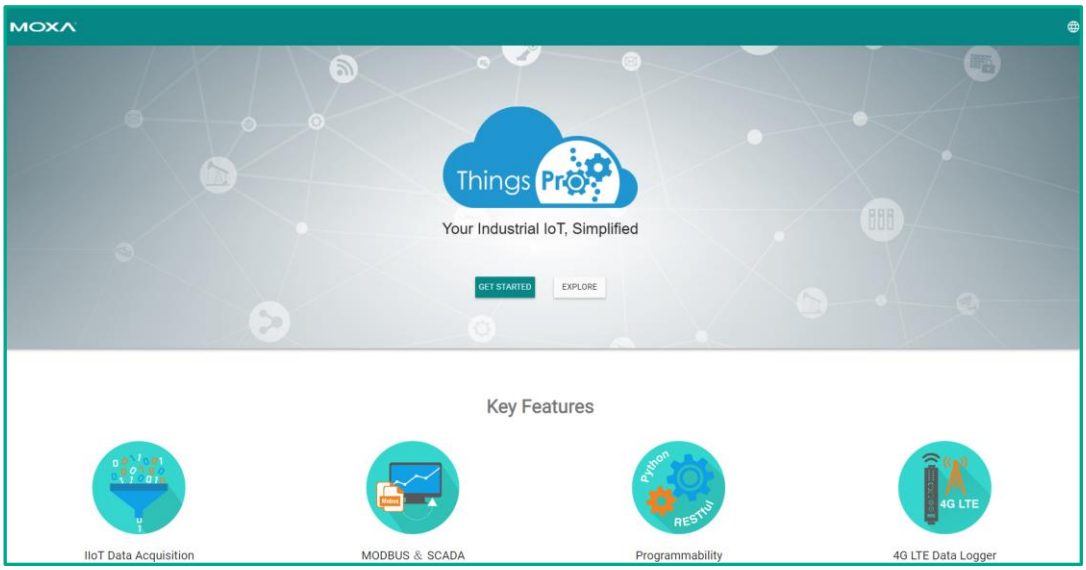
Below we discuss what an attacker can do with ThingsPro Suite upon exploiting the above vulnerabilities.

Some sensitive information on screenshots has been blurred at Moxa's request.

Exploitation stages

It should first be emphasized that the vulnerabilities we have identified can only be exploited by those attackers who are able to send requests to the ThingsPro server and receive responses from it, i.e., those who can access the device's administration panel. Only then does an attacker have the necessary conditions to carry out a successful attack on the device.

ThingsPro Suite administration panel home page



We divided an attack on ThingsPro Suite via the web service into four stages, from a minimal level of privileges in the web application to maximum privileges in the operating system.

1. Obtaining user authentication data;
2. Privilege escalation;
3. Arbitrary code execution;
4. Privilege escalation in the operating system.

We identified two ways in which an attack of this kind can be carried out – an easy way and a hard way.

We will start by describing the harder way of exploiting the vulnerabilities.

The hard way

Obtaining user authentication data

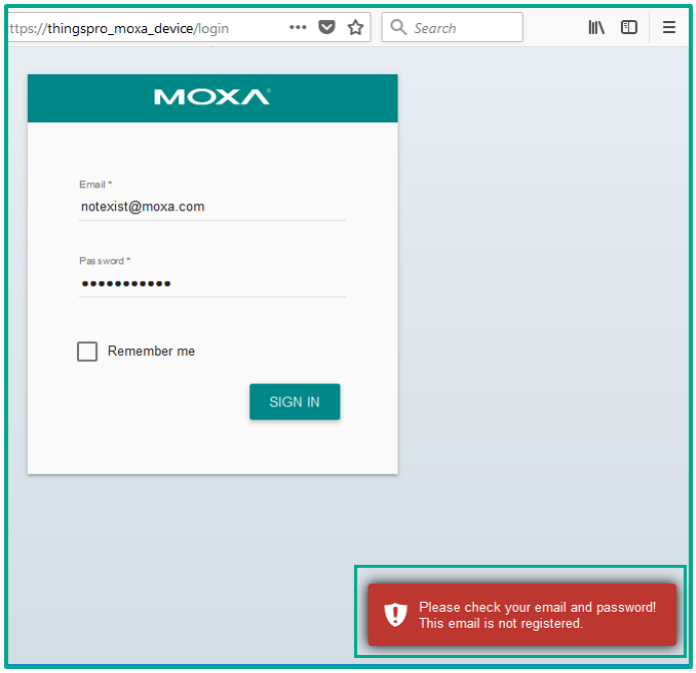
The hard way involves obtaining user authentication data by exploiting a vulnerability that we have identified – KLCERT-18-018/CVE-2018-18390.

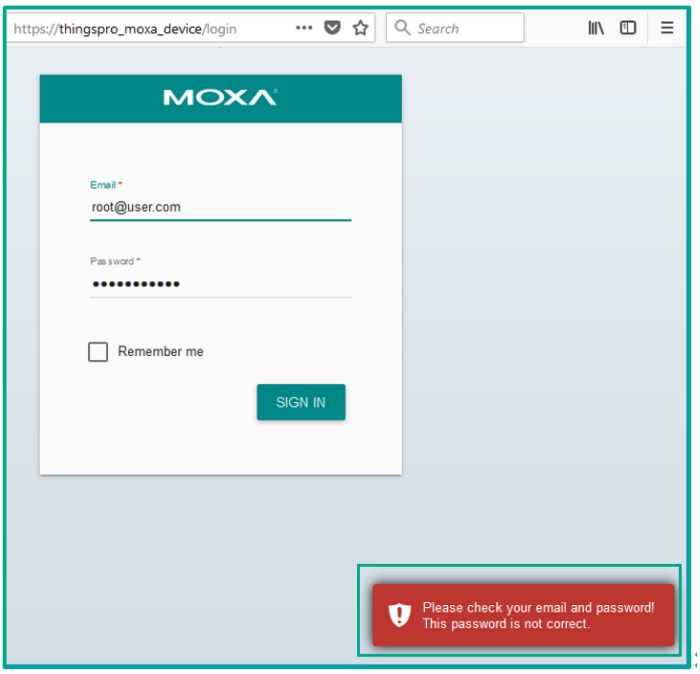
Since the ThingsPro Suite administration panel uses an authentication mechanism, attackers need authentication credentials to gain authorized access to the web interface.

An attacker can try to get confirmation that a specific user exists in the system by exploiting a vulnerability that is well described in OWASP-AT-002: the server's response to authentication data received can provide sufficient information to determine whether a specific user exists in the system or not.

If the authentication data sent to the server includes a user name that does not exist in the system, the server will indicate that by providing one response, and if the user name is correct but the password does not match it, the server will return a different response.

Exploitation of KLCERT-18-018/ CVE-2018-18390





This enables an attacker to identify users that exist in the system by enumerating their names.

If an attacker is able to obtain a valid user name for the system, this will make obtaining the relevant password significantly easier. And if the attacker is able to find the matching password, the next step will be to escalate privileges in the system.

Privilege escalation

After obtaining authentication data, the attacker needs to escalate privileges in the system to the highest level, because:

- Users with high privilege levels can do more in the system than other users;
- Attackers with high privilege levels can access more functionality that potentially contains vulnerabilities, providing the attackers with new opportunities.

Below, we look at two methods of escalating privileges – exploiting KLCERT-18-019/CVE-2018-18391 and exploiting KLCERT-18-020/CVE-2018-18392.

Example of escalating privileges by exploiting KLCERT-18-019/CVE-2018-18391

Authenticated ThingsPro Suite users can change their account data in the web panel. The data that can be changed includes user login, password, email address and company name. To modify that data, an HTTP request is sent to the web service. Part of the request is shown below:

```
PUT /api/v1/users/{:id} HTTP/1.1
[...]
{"id":{:id},"name":"user","role":"user",
```

It can be seen from the request that, in addition to the data specified on the web application's page, the request includes the user's current role.

To our surprise, after changing the role value specified in the request from user to root and resending the request, the server's response specified that the current user's role had been changed from user to root. Logging on to that account again confirmed that the privileges had been elevated to root. This vulnerability is categorized as Broken Access Control, a vulnerability type that ranks fifth on OWASP TOP 10 2017.

Example of escalating privileges by exploiting KLCERT-18-020/CVE-2018-18392

During our analysis of ThingsPro Suite we discovered that authenticated users can change data not only for their own accounts but also for any other account in the system. This can come in handy if a user needs to change the password for the root or admin account in case the password is forgotten.

However, this provides attackers with a privilege escalation opportunity. The easiest way of exploiting this vulnerability would be for an arbitrary user with any privilege level to change the password for the root account. The request fragment shown below is similar to the previous request fragment, the difference being that the user identifier equal to one, which corresponds to the user with the name root, is specified in the body of this request.

```
PUT /api/v1/users/5 HTTP/1.1
[...]
{"id":1,"name":"root","role":"root",
```

Upon receiving such a request, the server will change the password for the root user to the value specified in the request, after which the attacker can log on to the system as the root user.

This vulnerability is also categorized as Broken Access Control, ranking fifth on OWASP TOP 10 2017.

Arbitrary code execution

KLCERT-18-024/CVE-2018-18396 can be exploited in the following way.

Users with high privilege levels can access ThingsPro Suite functionality that modifies system settings or the behavior of ThingsPro Suite as a whole. To handle this level of requests, the web application has to use the capabilities of the Linux command line.

One such request is the request to extend the web application's RESTful API. (ThingsPro Suite can extend the web application's functionality by adding the relevant handler.)

The handler of this request does not check data received from the user for various special characters, passing the data directly to the command line for processing. This means that, by manipulating that data, an attacker can call any command from the Linux command line in addition to the command usually called by the ThingsPro web server for that handler.

This vulnerability is categorized as the Injection type and ranks as number one on OWASP TOP 10 2017.

A request fragment used to exploit this vulnerability is shown below:

```
POST /api/v1/market/upload HTTP/1.1
[...]
Content-Type: multipart/form-data; boundary=--------886584073

----------886584073
Content-Disposition: form-data; name="file"; filename="
"

----------886584073--
```

Privilege escalation in the system

As a rule, privilege escalation is necessary to develop an attack on the web server, after gaining control of the Linux command console. This is because web servers are usually run under a user with restricted privileges created in the system for that specific purpose. This is how such web servers as apache or nginx work. However, the ThingsPro Suite web server is already running in the system under the root user, which means that when an attacker has gained the ability to execute arbitrary commands, there is no need for privilege escalation.

The fragment below shows the output of the id command. The command returns the user's privilege level in Linux achieved after exploiting the vulnerability.

```
$ ncat -nklvp 8080
Ncat: Version 7.40 ( https://nmap.org/ncat )
Ncat: Listening on :::8080
Ncat: Listening on 0.0.0.0:8080
Ncat: Connection from
Ncat: Connection from
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
uname -a
Linux  Thu Apr 20 15:22:58
```

The uid, gid and groups values are equal to zero, which means that the user who executed the id command had the maximum level of privileges in the system.

This weakness is a significant flaw for the device as a whole.

Exploitability limitations

The hard way of obtaining privileges required an attacker to do the following in order to authenticate successfully:

1. Obtain an existing user's system login;
2. Find the password matching the login obtained.

It may prove to be impossible to do these two things – for example, when very strong passwords have been set for users or when an attacker is not sufficiently lucky. In those cases, attackers have to look for other ways of breaking into the web application.

Exploiting a vulnerability we have identified, KLCERT-18-023/CVE-2018-18395, can be one such way.

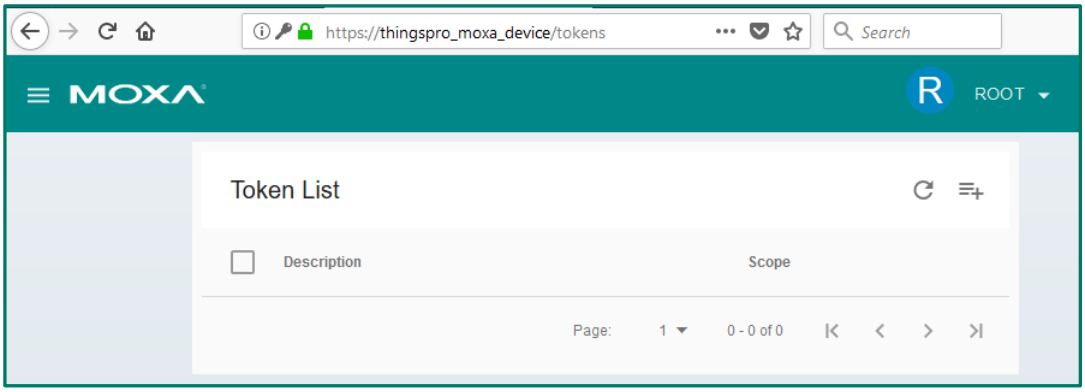
The easy way

Getting user authentication data

The easy way involves gaining user authentication data by exploiting KLCERT-18-023/CVE-2018-18395, another vulnerability that we have identified.

The vulnerability is caused by the ThingsPro Suite web server containing a "hidden" token for RESTful API. The web interface does not show the token on the list of tokens created, even for users with root privileges, i.e., the highest privilege level. This is demonstrated in the screenshot below:

List of pre-installed tokens in the web administration panel



The reason for this behavior is that upon receiving a request for a list of tokens for this tab, the request handler filters all tokens from the database for which the parameter hidden is set and does not display such tokens in the web interface.

A request for a list of all tokens existing in the database is shown below:

```
root@Moxa:/# sqlite3 /var/www/[redacted].db
SQLite version [redacted] 2014-10-29 13:59:56
Enter ".help" for usage hints.
sqlite> .mode line
sqlite> select * from tokens;
         id = [redacted]
    content = [redacted]
      scope = [redacted]
     hidden = [redacted]
description = Internal Local Token
  createdAt = 2000-01-15 22:41:52.669 +00:00
  updatedAt = 2000-01-15 22:41:52.669 +00:00
```

Although, according to the description, the “hidden” token is an “Internal Local Token”, it can also be used by an external attacker.

Privilege escalation

ThingsPro Suite uses tokens of two types – read and write. A token of the type “write” can perform all the operations that are available to users with root privileges in the web application. A token of the type “read” is a user with “user” level privileges.

The “hidden” token is a token of the type “write”, which means that no privilege escalation is necessary if that token is used.

Arbitrary code execution

With a “hidden” token available and the resulting maximum privileges in the web application, both the search for vulnerabilities and the vulnerability exploitation algorithm that leads to arbitrary code execution becomes similar to that described above in “The hard way” section – the difference being that the “hidden” token will be used instead of a session in request headers.

Privilege escalation in the system

Given the ability to execute arbitrary code in the system and the availability of the token, the algorithm of privilege escalation in the system is similar to that described in “The hard way” section.

Exploitability limitations

The easy exploitation method described above requires the attacker to know the value of the “hidden” token in order to pass authentication successfully.

The value of the “hidden” token is generated when the web application is first started and is the result of executing the bcrypt hash function for ten pseudorandom characters. This means that this value would be hard to obtain through a bruteforce attack; however, if the seed used to generate the ten pseudorandom characters can be obtained, these characters can be reconstructed and used to generate the token again.

Another possibility for obtaining the “hidden” token is to extract it from the file system: ThingsPro Suite saves the token’s value in a file that can be accessed via a static path, and as a record in its database.

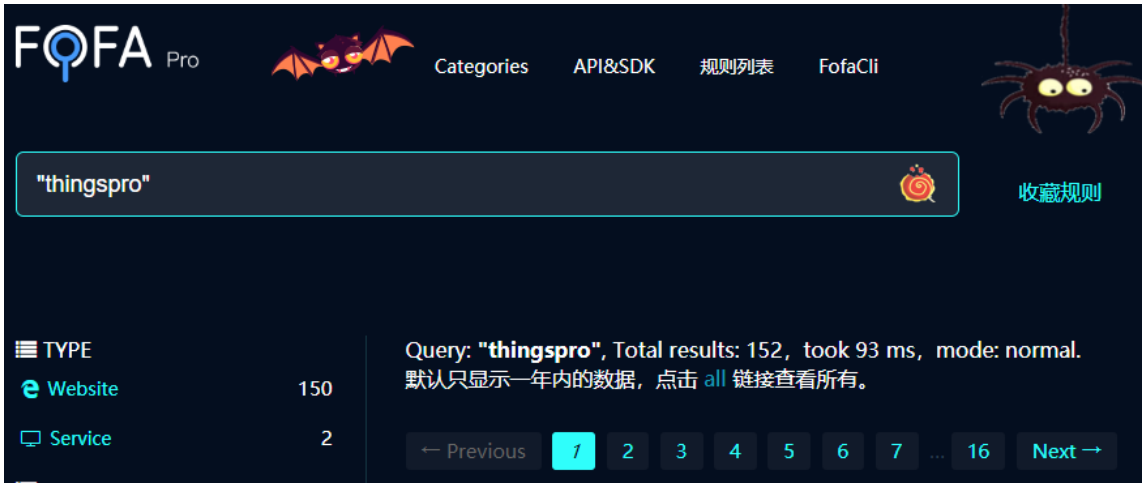
Attack phases

Detection

Platforms such as shodan.io and fofa.so scan network ranges for available network services, making it easier to detect available ThingsPro Suite IP addresses.

For example, in early November 2018, shodan.io returned 43 IP addresses at which services returned the string “thingspro” in response to a network request. The fofa.so platform returned 152 results in which “thingspro” is mentioned. These results are shown below:

Shodan and Fofa search results for ThingsPro Suite



Version identification

Some attackers skip the stage of identifying the version of the software being attacked and move right on to vulnerability exploitation. During penetration testing we are also sometimes forced to use this approach; however, successfully passing this stage improves the chances of subsequent successful vulnerability exploitation.

In addition to the main web service on ports 443 and 80, which provides the administration panel, the ThingsPro Suite web server has another web service available on port 8880. When a request is sent to that web service over HTTP, it returns a static XML page, which includes the following information:

- ThingsPro Suite version;
- MAC address;
- IP address.

The main problem with the web service available on port 8880 is that it has no authentication mechanism. This means that any internet user can get information which can be used to determine unambiguously which version of ThingsPro Suite is used and whether it is vulnerable.

A sample response from the web service on port 8880 is shown below:

Response from the web service on port 8880

Exploitation

The stage which follows detecting a ThingsPro Suite web service and identifying its version is vulnerability exploitation, which is described in detail in the previous chapter. It is worth reiterating, however, that consecutive exploitation of two vulnerabilities provides an attacker, who is initially only able to send requests to the main ThingsPro Suite web server, with access to the Linux command line on the ThingsPro Suite server with the privileges of a root user.

Other vulnerabilities

In addition to the five vulnerabilities mentioned above, our research yielded two more flaws, which are not required for the attack described above but are nevertheless worth describing.

KLCERT-18-021/CVE-2018-18393 is a password management issue: a user is not required to enter the old password to change it to a new one. It can be hypothesized that if this vulnerability did not exist in ThingsPro Suite, one of the vectors used to exploit KLCERT-18-020/CVE-2018-18392 (Broken Access control) to change the password for the root user (which was used to escalate privileges) would not be applicable.

KLCERT-18-022/CVE-2018-18394 is relevant following the exploitation of vulnerabilities such as SQL injection. The issue is that ThingsPro Suite stores sensitive data in its database in cleartext. That sensitive data includes information on the tokens generated, as well as the “hidden” token. The upshot of this is that, although passwords are stored in the database as hashes, an attacker can attempt to extract tokens and use them in place of authentication data, thereby eliminating the need to de-hash passwords.

Conclusions

By the end of 207, we had identified 7 vulnerabilities in the ThingsPro Suite product by Moxa. Those flaws included vulnerabilities with a CVSS v.3.0 base score of 10. Consecutive exploitation of two vulnerabilities enables a remote attacker who initially only has access to the ThingsPro Suite web application to gain access to the command line in the operating system in which ThingsPro Suite is running. The exploitation of these vulnerabilities can be automated to conduct a bulk attack on devices.

Importantly, we identified most of the above vulnerabilities within a very limited timeframe without performing a deep technical analysis or analyzing the web application’s source code. This indicates that these vulnerabilities can be found and exploited by an attacker who lacks high-level technical skills.

We reported all the vulnerabilities we had identified in Moxa software to the vendor. All of the vulnerabilities have been fixed.

- Our research started in December 2017.
- In January 2018, we prepared a report and provided it to Moxa’s security group.
- Moxa released a patch in late July 2018.
- A security advisory was released in October 2018.
- Vulnerability descriptions were published in October 2018.

We would like to extend our thanks to Moxa for their cooperation. The staff of the company’s security service were very professional and eager to address their product’s security issues. We would also like to mention the transparency in our negotiations with the company’s representatives and their prompt replies.

Kaspersky Lab Industrial Control Systems Cyber Emergency Response Team (Kaspersky Lab ICS CERT) is a global project of Kaspersky Lab 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 Lab ICS CERT devotes its efforts primarily to identifying potential and existing threats that target industrial automation systems and the industrial internet of things.

Kaspersky lab ICS CERT

cs-cert@kaspersky.com