How Does Triton Attack Triconex Industrial Safety Systems?

By IoT Security Research Lab

Published: 2021-02-23 · Archived: 2026-04-05 21:09:14 UTC

Triton, also known as TRISIS or Hatman, is a piece of malware specially crafted to attack industrial safety systems. In particular, Triton exploits vulnerabilities on the Triconex safety instrumented system from Schneider. Despite this system being deployed at more than 15,000 sites across the world, the malware allegedly only targeted one critical energy industrial site in the Middle East in 2017.

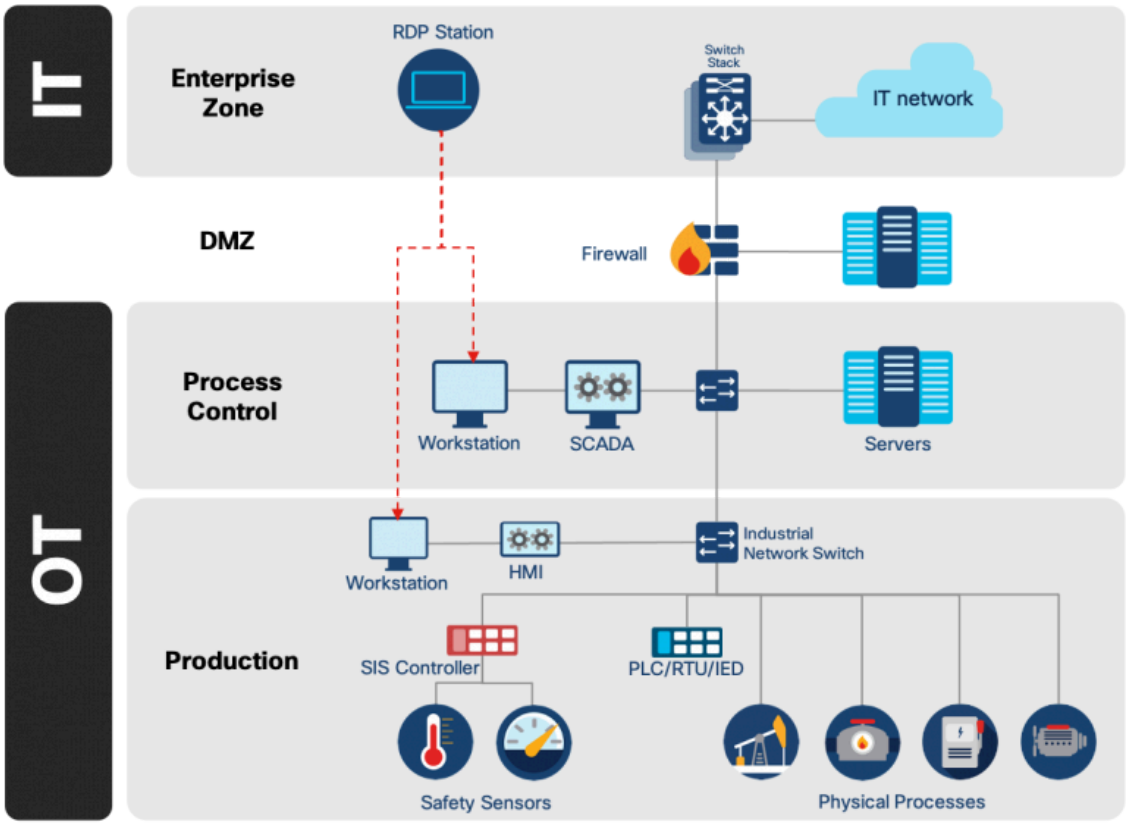
Safety instrumented systems (SIS) are used within industrial plants to monitor process values and parameters in order to assess whether the process remains within predefined operational limits. When the process drifts away from its safety limits, the SIS triggers alarms and performs actions to put the plant in a safe state. Thus, SIS such as Triconex play an important role in protecting a plant from accidents that might lead to significant environmental, health or economic damage.

The complexity of Triton, the nature of its target as well as its attack techniques and procedures have led many security researchers to assess with some confidence that it is consistent with a nation state threat actor. As such, the U.S. Department of Treasury, through sanctions announced by its Office of Foreign Assets Control, alleged that this attack was supported by a Russian government research institution, namely the State Research Center of the Russian Federation FGUP Central Scientific Research Institute of Chemistry and Mechanics (TsNIIKhM).

The Triton attack can be broken down into different phases:

1. After gaining an initial access into the IT network, the malware attempted to move towards the OT network using standard techniques (custom tools with functionalities close to those of Mimikatz or PSExec). As a result, Triton managed to compromise an engineering station within the safety system. An example of an industrial network architecture including safety systems is shown in the figure below.
2. From the compromised station, Triton was able to launch a *dropper* (trilog.exe) with the aim of delivering *backdoor* files to the PLC.
3. The *backdoor* consists of two files. One file makes use of a *0-day* exploit to inject the contents of the second file into the PLC's memory. Subsequently, the attackers had total control of the target.
4. In the final phase, the attackers' aim was to take control and manipulate the device, leading to concrete effects on the safety systems of the physical installation. However, this phase did not take place and the associated sources were not identified. A handling error in the target forced it into *fail-mode* which stopped the plant's production. The attack was identified following this shutdown.

Dropper



Example of an industrial network architecture. Although the safety systems are autonomous, they are often connected to Windows engineering stations for updates and maintenance.

The attackers first wrote the dropper as a Python script named *script_test.py*. Then, using the Py2Exe utility, the attackers compiled the script to an executable called *trilog.exe* . When compiling the dropper, the attackers used Py2Exe's default settings. This enabled all of the files related to this implementation to be easily recovered.

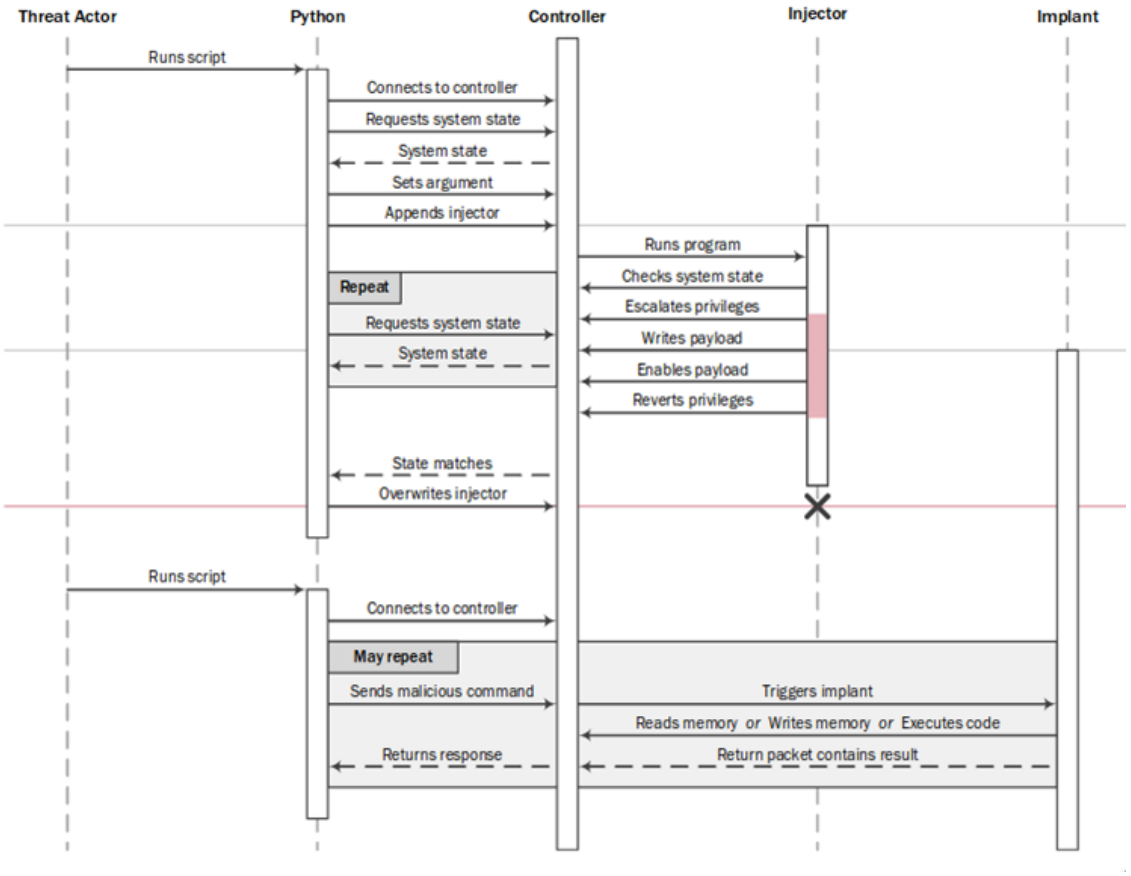
In order to craft the dropper, the attackers reverse-engineered the proprietary TriStation Protocol used to program the system. The first analysis, published in December 2017, identified Triconex MP3008 models with software versions between 10.0 and 10.4 as vulnerable.

Using this implementation, the dropper first connects to the target to transfer the malicious files that are used to control the machine. The dropper then regularly scans the target to establish whether the injection is complete.

However, at this stage, two significant weaknesses impact the attackers:

- The Triconex machine only accepts program downloads when in "*PROGRAM*" mode whereas other modes (such as "*RUN*") prevent it. These modes are changed manually using a physical key system.
- The injected code is not persistent. For example, the code is deleted when downloading a new program ("*DOWNLOAD ALL*").

While network discovery options were present in the malware, the target's IP address was hardcoded in the recovered files. This reflects the attackers' deep preliminary information gathering phase.



General operation of the Triton attack. (source CISA ICS-Cert)

Backdoor

The backdoor comprises an injector and an implant. First, the injector (a program that aims to *inject* a payload) checks whether the controller is vulnerable to the exploit. If the tests are conclusive, the exploit is used to raise privileges. These new permissions enable the content of the implant to be written in the system memory area. This memory area is not wiped when downloading new programs. Moreover, since safety systems are critical components, they are almost never restarted. Therefore, the risk of the implant being deleted is low.

Once this payload is in memory, the original program is patched to include a conditional break to the address of the payload. A RAM integrity verification feature is also patched to avoid detection.

The implant allows specific commands to be listened to and processed without any restriction with respect to the operating mode (which can be changed using the physical key mechanism). More precisely, it analyzes the messages that are usually used for debugging processes (GetMPStatus messages) and executes the expected command (read, write, or execute). This communication allows, for example, the memory of the machine to be modified even if it is in a mode that prevents the standard download of a program ("*RUN*" mode).

0-day exploit

The injector used a vulnerable system call to obtain a 2-bytes arbitrary write ability which it then used to increase its privileges. During a system call, when switching from *USER* mode to *SUPERVISOR* mode, registers are saved to a known memory area. One of these registers, *MSR*, contains a bit which indicates the privilege level. Using the

vulnerable system call, it is possible to overwrite this bit, thus giving supervisor privileges to the user once the system call ends and the registers are reloaded from the memory.

Redundancy functions

Among the files discovered during the first analysis, the file *CRC.pyc* implemented some cyclical redundancy control functions. The use of such functions is normal and is found in many protocols. This does not necessarily mean, however, that all protocols use the same function.

Various functions related to different protocols (such as Modbus or XMODEM) were identified when analyzing the file. While not every single function was useful during the attack, this implies that the attackers could target other types of machines or industrial systems.

Detection of the attack

At this stage, the attackers were in a position to control the target remotely, regardless of its mode of operation. However, before any potential physical consequences were observed, the system went into a safe mode (*failed safe state*). This behavior resulted in a shutdown of the production line and the discovery of the attack. The ultimate aim of the attackers has not been identified.

The triggering of this safe mode was probably linked to a handling error in the machine. This might occur, for instance, when writing in the incorrect memory area. Various analyses have shown that the attackers had difficulty implementing and handling the TriStation protocol and the associated features.

Detection measures

To detect the behavior of the malware, Cisco Talos published Snort rules#45260, #45477 and #45478) that can be activated in a next generation firewall, such as Cisco Firepower firewalls and, in particular the ISA 3000 Industrial Security Appliance.

However, these rules might fail to detect another Triconex attack as the attackers would probably not use the exact same components. Therefore, it is necessary to analyze the UDP communications on port 1502. Various industrial security products such as Cisco Cyber Vision can analyze the communications over the Triconex protocol to detect anomalies and to identify potential attacks.

> **Visit our IoT Security Research Lab for**
>
> **more technical reports on IoT/OT Security**

Subscribe to the Cisco IoT Security Newsletter.

Source: https://blogs.cisco.com/security/how-does-triton-attack-triconex-industrial-safety-systems