Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of the Environment, Transport, Energy and Communications DETEC

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech Division

REEL Demo – Romande Energie ELectric network in local balance Demonstrator

Deliverable: 3a4 Protection of electrical infrastructure from cyber-attacks

Demo site: Chapelle

Developed by
Peter Gysel, FHNW
Michael Roth, FHNW
Dominik Link, FHNW

[Olten, 28.08.2020]

1. Description of deliverable and goal

1.1. Executive summary

DSOs use their IT systems to operate critical infrastructure which can be considered a high-value target for attackers such as terrorists and cyber criminals. Therefore they need to protect their IT infrastructure in various ways. We identified that an attacker will most likely try to get access through the computer systems in the office or operator networks. By protecting these computers, we can protect the electrical grid against attacks which can cause outages or even lasting damage to hardware components. To reach this goal, we present a new approach where we monitor system activity on computers, identify potential malicious actions and make them subject to authorization.
An action can be malicious or benign depending on the reason it has been executed or the intention of the responsible process. Since the reason and the intention cannot be evaluated automatically, applications need to have permission to execute critical actions such as taking a screenshot which can be a legitimate user action or data exfiltration.
We have implemented a prototype which has shown good results. Our prototype can fend off attacks by a custom malware which is undetected by commercial anti-malware products. However, a complete implementation requires a lot of effort.

1.2. Research question

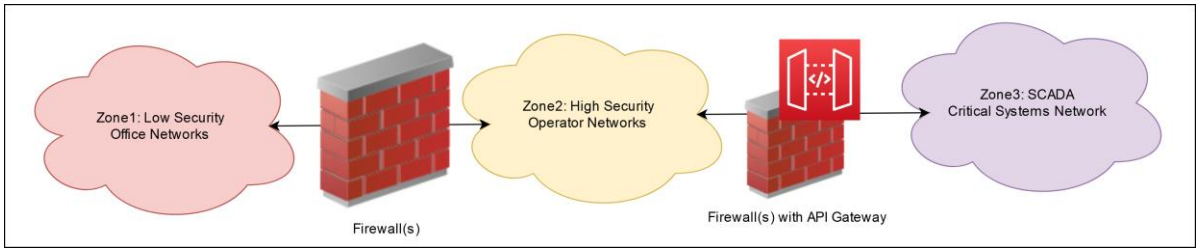
What does the threat landscape look like and how can we protect DSOs from targeted attacks?

1.3. Novelty of the proposed solutions compared to the state-of-art

Current anti-malware systems leverage various methods for malware detection. These include recognizing signatures, behavior analysis, machine learning, and many more. However, they all rely on previously learned and observed data and patterns. Therefore, their protection against new and future attacks is limited as attack signatures can be obfuscated with small changes. Our solution takes a proactive approach by making actions a software may executes subject to authorization. This enables us to detect new and future attacks.

1.4. Description

DSOs need to protect their critical IT infrastructures to provide a reliable service and avoid damaging the electrical grid. The consequences of a successful attack of a DSO are devastating. Big parts of Switzerland could be without power for a longer period. This scenario is realistic in the modern ages of digital warfare.



First, the network needs to be designed in a secure manner. While this varies for each DSO, the basics remain the same. The figure shows this basic design principle of a secure network. There are different network-zones in which different security standards are enforced. The simplest secure network design has three different zones:

1. The "office" zone. Within this zone, multiple networks exist and are used for everything which is not related to operating the power network, such as HR, billings, public relations, etc.
2. The zone where everything related to operating the power networks. The SCADA network and its devices can be controlled and configured through this zone.
3. The zone where the devices are located. Usually, they are part of a "SCADA" network. These SCADA-networks often define very strictly how they have to be operated.

In some cases, there is not even a connection between the zones 1 and 2 for enhanced security.

We differentiate between two types of attacks:

1. **Bulk attacks**: The attack is executed against as many targets as possible. If one encounters such an attack, it was random or one out of many.
2. **Targeted attacks**: The attack is specifically designed for a single target and constructed for exactly its network and environment.

By applying a network architecture similar to the one explained above and using an up-to-date anti-malware software, DSOs can achieve a good protection against bulk attacks. However, targeted attacks are much more difficult to detect since the malware is designed for a specific victim and can be tested against the used anti-malware software in advance to ensure the malware is not detected at the victims network.

Targeted attacks can be carried out in many different ways. Social engineering is one of the most used technique to get initial access to a system. From there, the malware progresses and spreads within the victim's network until it has full control. Therefore, if the clients in the zones 1 and 2 can be protected against such malware, the SCADA systems stay safe.

To improve the clients within the DSOs' networks, we analyzed malware and its behavior. This led to new methods to mitigate targeted attacks. The results of our analysis shows that a malware executes two types of actions:

1) **Helper-Routines**: These are basic actions the malware needs to perform before being able to execute the actions leading to the actual goal. Helper routines consist of actions like hiding, persisting itself, escalating privileges, spreading in a network, and more.
2) **Goal related actions**: These actions actually fulfill the malware's goal which is one or a combination of stealing, destroying, or manipulating data. This can

potentially lead to controlling devices in a SCADA network and subsequently damage the electrical grid.

Unfortunately, from a technical point of view, many of these actions are executed by a lot of benign software too. For example, taking a screenshot can be a feature used by a user which is benign behavior. However, if a malware takes a screenshot to exfiltrate data, its behavior can be considered malicious.

Therefore, we conclude that not the actions themselves define whether a software is benign or not, but the *intention*, e.g. *why* a specific action has been executed, classify a benign or malicious behavior.

Since software cannot determine the reason nor the intention of other software, this problem cannot be solved easily. Therefore, any software that executes actions which can be used to conduct malicious behavior could harm a system. However, many benign applications and processes need to use such actions to work properly. Therefore, a method to distinguish between 'good' and 'bad' usages is required.

To solve this problem, we use an approach inspired by modern smart phone operating systems such as Android or IOS. We introduce permissions for applications. However, instead of giving permissions to access resources (like the camera, phone, etc.), we software to execute such potential malicious actions.

To implement this, we need to be able to recognize when a process executes such actions. For this, we monitor the syscalls a process makes to the Windows API and recognize the resulting actions. For example, one possible way for taking a screenshot requires calling the following functions in this order and supplying specific arguments:

- `GetWindowDC()`
- `CreateCompatibleDC()`
- `CreateCompatibleBitmap()`
- `BitBlt()`

By monitoring these calls, we can match this sequence of calls to the action 'taking a screenshot'. After the matching, the allow-list is consulted to query if the process is allowed to perform this action. If not, an incident is created and countermeasures such as killing the process or simply reporting the incident can be invoked.

The screenshot action is just exemplary, but the idea is valid for each action that can become dangerous for an IT network.

2. Achievement of deliverable:

2.1. Date

30. September 2020 (deadline)

28. August 2020 (delivered)

2.2. Demonstration of the deliverable

We were able to verify the effectiveness of such an approach. A prototype which recognizes a subset of all actions is implemented. Using this prototype, we can detect malware which is undetectable by commercial anti-malware software.

3. Impact

As successful attacks (for example, Stuxnet) show, security is not only a feature but an urgent requirement. It is absolutely necessary to successfully carry the demonstrators over to production.