Kill Chain Attack Modelling for Hidden Channel Attack Scenarios in Industrial Control Systems

Tom Neubert [*,**] **Claus Vielhauer** [*]

[*] *Brandenburg University of Applied Sciences, Brandenburg (Havel), Germany (e-mail: tom.neubert@th-brandenburg.de, claus.vielhauer@th-brandenburg.de).*
[**] *Otto-von-Guericke-University Magdeburg, Research Group Multimedia and Security, (e-mail: tom.neubert@iti.cs.uni-magdeburg.de)*

Abstract: The protection against Advanced Persistent Threats (APTs) is an important topic in nuclear and industrial information technology security since the last decade. Nowadays steganography, i.e. information hiding techniques are increasingly used by attackers in order to operate without being detected. The usage of hidden channel communication in APTs creates a novel form of attack scenarios for which the current defense mechanisms are usually ineffective. In order to defend industrial control systems against those attacks, it is necessary to understand and comprehend the attacks. Thus, this paper presents how attack modelling based on the Lockheed Martin Cyber Kill Chain can be used to analyze hidden channel APT attack scenarios and how it can be used to elaborate defense mechanisms and to reveal attack indicators along all phases of those attack scenarios.

Keywords: Methodologies and tools for analysis of complexity, Kill Chain, Steganography, Hidden-Channel-Communication, Detection

1. INTRODUCTION

Since the `Stuxnet`-attack (see Kushner (2013)) in 2010, the protection of industrial control systems (ICS) against advanced persistent threats (APTs) is of great importance for nuclear and industrial information technology (IT) security. Nowadays, a trend in IT is the usage of steganographic techniques to implement hidden channel network communication (e.g.: the paper of Schmidbauer et al. (2019)). Hidden channel communication can be used by cyber attackers in order to operate without being detected as long as possible. The combination of hidden channel communication (implemented by steganographic techniques) and APTs creates a novel form of hidden channel attack scenarios (as introduced in Hildebrandt et al. (2020)). For those novel scenarios, current defense mechanisms are usually ineffective due to the high level of sophistication associated with steganographic techniques to conceal an attack.

In order to defend ICS against these novel APT attack scenarios, an attack modelling is necessary to understand and comprehend these attacks. As a result of the attack modelling, defense mechanisms (for prevention, detection and reaction of attack scenario) can be elaborated according to the attack scenarios.

Thus, in this paper we present an attack modelling based on the Lockheed Martin Cyber **Kill Chain** by Hutchings et al. (2011) of three exemplary APTs with hidden channel communication for ICS. With our attack modelling we want to show how to analyze APTs with hidden communication, how to derive defense mechanisms and how to reveal attack indicators along different phases of attack scenarios. The Kill Chain (see Hutchings et al. (2011)) is a multi-stage model of the U.S. defense, security and advanced technology company Lockheed Martin Corporation. It is used to analyze cyber attacks, especially APTs and to develop defense mechanisms along the different phases of the attack.

This work contributes Kill Chain attack modelling of three exemplary hidden channel attack scenarios, which have been derived from security incidents in other kinds of IT systems and are likely to represent attack scenarios for ICS. Additionally, defense mechanisms and attack indicators based on the Kill Chain attack modelling along every phase of the three attack scenarios will be discussed. The novelty of this kind of attacks is the combination of hidden channel communication in APTs for ICS. We want to show that attack modelling based on the Kill Chain is a useful tool to identify vulnerability of ICS in presence of hidden communication and to counter APT attacks. Additionally, we intend to raise the awareness for those attacks and to show how attack modelling can be used to elaborate defense mechanisms and to reveal attack indicators. Furthermore, these scenarios can be used as the basis for future experimental evaluations. We are aware that we present only a limited number of attack scenarios but we assume that our emphasized defense mechanism can be applied for other hidden channel attack scenarios in ICS. The malware of our three exemplary attack scenarios

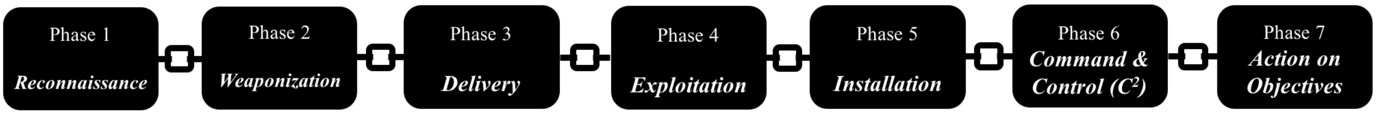


Fig. 1. Visualization of Lockheed Martin Cyber Kill Chain based on Hutchings et al. (2011)

use four different channels/carriers for the steganographic communication:

- Modbus-TCP over Ethernet (see ACROMAG (2005)),
- OPC-UA over Ethernet (see OPC-Foundation (2008)),
- syslog server (see Parker (2016)) and
- NTP server (see Williams (2008)).

The work is structured as follows: Section 2 describes the state of the art of the Lockheed Martin Kill Chain and of hidden channel communication. In Section 3, 4 and 5 three exemplary hidden channel attack scenarios will be described and modelled. Additionally, defense mechanisms and attack indicators for the attacks based on the Kill Chain attack modelling will be discussed. Section 6 concludes the paper with a summary and future work.

2. STATE OF THE ART

In this section we introduce the basics of the Lockheed Martin Cyber **Kill Chain** based on the work of Hutchings et al. (2011) and we give a short overview about hidden communication in networking systems.

2.1 Lockheed Martin Cyber Kill Chain

The Lockheed Martin Cyber Kill Chain is a 7-phase-model by Hutchings et al. (2011) for the modelling of cyber attacks, especially advanced persistence threats invented by Lockheed Martin Cooperation. The kill chain is visualized in Fig.1. It is based on the F2T2EA (find, fix, track, target, engage, assess) military kill chain (see Azuma et al. (2006)). Cyber attacks with a high complexity need to be modelled to understand the tactics, techniques and methods of the attackers, in order to structure the attack and to install defense mechanisms to defend those attacks and for this purpose the Kill Chain is introduced.
The Kill Chain is an end-to-end process described as a “chain” because an interruption will stop the entire process. This means that attackers have to go through the entire chain to reach their goals, while a defender can interrupt the attack on every phase of the chain. Consequently, it is desirable to install defense mechanisms for every phase of the Kill Chain. The seven phases of cyber kill chain are defined as follows (see Hutchings et al. (2011)):

(1) **Reconnaissance** - Identification, research and selection of attacking targets. This is typically accomplished by
    - spying webpages,
    - fishing mails,
    - social engineering and
    - sniffing attacks.

(2) **Weaponization** - Prepares the operation by assembling attacking tools (e.g.: coupling exploit with backdoor into deliverable payload).

(3) **Delivery** - Transmission of the assembled attacking tools to the targeted environment. Relevant delivery techniques include
    - email attachments,
    - websites,
    - USB removable media and more.

(4) **Exploitation** - Triggers the delivered attacking tools. Mostly, it targets an application or operating system vulnerability, but it can simply be triggered by users or by delivery.

(5) **Installation** - Installs delivered attacking tools with a backdoor or remote access trojan horse on the victims system.

(6) **Command and Control (C2)** - Sets up a C2 channel to remotely control the installed attacking tools. Mostly, compromised hosts are used from an internet controller server to establish the C2 channel. Typically, encryption is used to conceal the content of the C2 communication.

(7) **Actions on Objectives** - Achieves the goals of the attack scenario. Only in this phase, the attackers can take actions to achieve their actual attack goals. Typical goals of APTs are:
    - data exfiltration,
    - compromise additional systems and
    - trigger malfunctions.

The Kill Chain is a guide to analyze APTs and should provide defense mechanisms for the modelled attack scenarios. Based on the detection in a certain phase, defenders can assume that prior phases have already been executed successfully. In our APT scenarios, we assume that a detection of the hidden communication is most likely in Phase 6, because steganographic channels aim to conceal the mere existence of illegal communication, which seems to be a desirable property for attackers. However, only a complete analysis of prior phases can prevent or at least mitigate future intrusions (see Hutchings et al. (2011)). This is visualized in Fig. 2. Mostly, incident response processes start after Phase 4. This reveals that defenders are naturally disadvantaged and inevitably too late. Advanced tools, technologies and processes should be used along the kill-chain to offset this disadvantage.

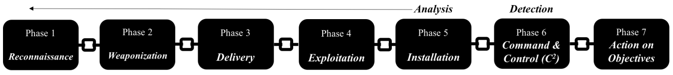


Fig. 2. Visualization of analysis and detection in Kill Chain based on Hutchings et al. (2011)

2.2 Hidden Communication in Cyber Attacks

Hidden communication implemented by steganographic techniques is increasingly used in cyber attacks in order to stay undetected as long as possible by embedding and hiding actual communication between the malware and the attacker in unsuspicious information exchange. The criminal use of information hiding is subject of the CUIng initiative[1] which cooperates with the Europol European Cybercrime Center. In advanced persistent threat scenarios steganographic techniques can be used for example by hiding communication within the normal network traffic. The purposes of hidden communication in APTs are mainly:

- command & control,
- data exfiltration and
- data injection.

To achieve these goals, the steganographic communication channel can be formed with different storage and timing methods (see Mazurczyk et al. (2018)). Storage channels are one class of network steganography methods which modify sub-carriers in a so-called carrier to create a storage hidden channel by e.g. modifying protocol fields, such as unused bits of a protocol header. Timing channels are the other class of network steganography methods. These methods modify the timing of events of a carrier to create a hidden channel by e.g. modifying the timing of protocol messages (see Mazurczyk et al. (2018)). For both methods it is essential that the purpose of the carrier is not effected and that the modifications performed by the steganographic embedding do not produce suspicious anomalies. Otherwise, it is presumably easier to detect by steganalysis. “Steganalysis is the science of detecting hidden information” (see Boehme (2010)).

3. HIDDEN CHANNEL ATTACK SCENARIO 1 ($HCAS_1$)

In this section the first of our three attack scenarios $HCAS_1$ is presented. This and the following two chapters (4 and 5) have the same structure: they start with a description of the attack vector, followed by the Kill Chain attack modelling and a discussion about defense mechanisms and attack indicators for the respective attack. We focus on a more comprehensive description of defense mechanisms in Phase 6, because we consider the best chance to detect the attack in this step.

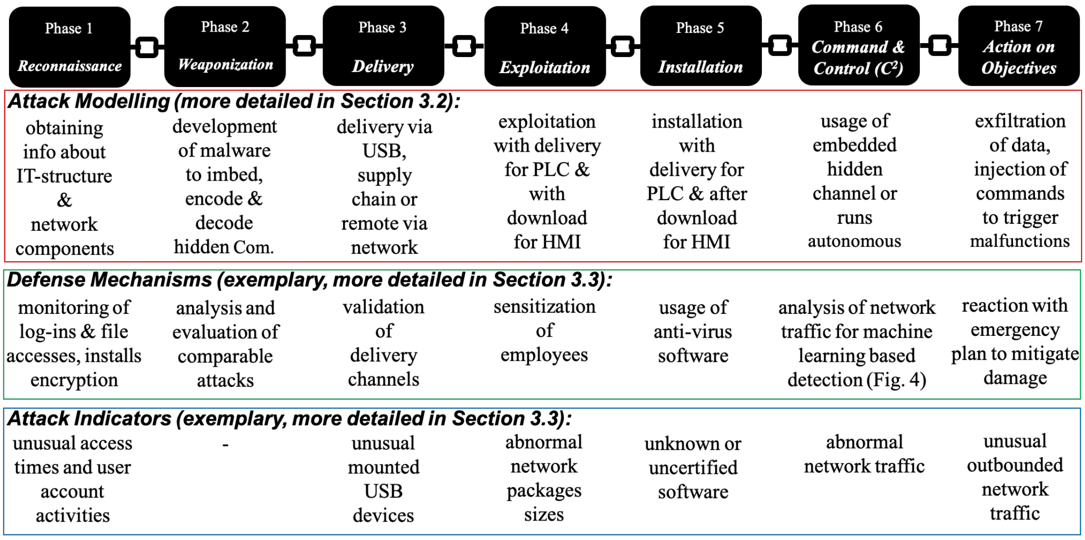
3.1 Attack Vector for $HCAS_1$

In $HCAS_1$ we propose to focus on the communication between a programable logical controller (PLC, see Bolton (2015)) and a Human-Machine-Interface (HMI, see Gonzalez (2015)). This is a classic communication scenario in an industrial control system (ICS), because HMIs are usually used to set inputs and to display inputs and outputs of PLCs. For this scenario, we suppose that the attacker uses the network traffic between PLC and HMI to embed hidden communication with the goal of command & control, data exfiltration or data injection. Typical network protocols used in this application field are for example Modbus-TCP (see, ACROMAG (2005)) and OPC-UA (see OPC-Foundation (2008)), which are the carriers for our hidden communication in $HCAS_1$. To send, receive, encode, decode and embed the hidden communication, malware has to be installed on the PLC and HMI. With $HCAS_1$ the attacker is able to exfiltrate data or to inject commands to trigger malfunctions. We note that one has to be aware, that all presented attack scenarios are complex attacks which can take multiple months or years. A related attack scenario is introduced in Hildebrandt et al. (2020). Hildebrandt et al. use OPC-UA as a carrier to embed a hidden communication channel in ICS.

3.2 Kill Chain Attack Modelling for $HCAS_1$

The Kill Chain for $HCAS_1$ is visualized in Fig. 3. In **Phase 1** the attackers have to obtain information about the network and IT infrastructure of the objective, because for the attack they need to know where Modbus-TCP or OPC-UA communication between PLC and HMI in the system takes place to embed their hidden channel communication. To get these necessary information, two options can be considered for the attackers. The first option is an insider, who gives the information to the attackers. The second option is a sniffing attack on the enterprise level of the objective.
During **Phase 2** the attackers develop their attacking tools for the HMI and the PLC to send, receive, encode, decode and embed the hidden communication into the ICS specific network protocols (Modbus-TCP and OPC-UA).
In **Phase 3** the malware developed in Phase 2 will be delivered. Therefore, the attackers have to deliver the malware to the HMI and to the PLC. For the HMI, an insider could simply use an USB removable disk or it could be delivered through a remote access over the network of the objective. For the PLC, we assume that a supply chain attack (corrupts objective through outside partner with access to objective) is most likely, which corrupts the firmware of the PLC. In addition, we mention that the PLC could also be corrupted via its SD-card or via an engineering-workstation (where the malware is downloaded to the PLC).
Phase 4 starts with delivery for the PLC because a supply chain attack already delivers the PLC with corrupted firmware. For the HMI it starts with copying the malware from USB to HMI or with downloading the malware from the network to HMI.
Phase 5 installs the malware and provides the hidden channel communication between PLC and HMI. For the PLC the installation is done with delivery and for the HMI, it is successful when the malware runs on the HMI.
Phase 6 uses the installed malware (which has embedded the hidden communication channel) to infiltrate commands and to exfiltrate data. When the hidden channel is only used to exfiltrate data, Phase 6 could also run autonomously because there is no need for a bidirectional communication.
In **Phase 7** the attackers want to achieve their goals. So, in this phase, data is exfiltrated or commands are infiltrated to trigger malfunctions.

[1] https://www.cuing.org



Phase 1 Reconnaissance	Phase 2 Weaponization	Phase 3 Delivery	Phase 4 Exploitation	Phase 5 Installation	Phase 6 Command & Control (C^2)	Phase 7 Action on Objectives
Attack Modelling (more detailed in Section 3.2):						
obtaining info about IT-structure & network components	development of malware to imbed, encode & decode hidden Com.	delivery via USB, supply chain or remote via network	exploitation with delivery for PLC & with download for HMI	installation with delivery for PLC & after download for HMI	usage of embedded hidden channel or runs autonomous	exfiltration of data, injection of commands to trigger malfunctions
Defense Mechanisms (exemplary, more detailed in Section 3.3):						
monitoring of log-ins & file accesses, installs encryption	analysis and evaluation of comparable attacks	validation of delivery channels	sensitization of employees	usage of anti-virus software	analysis of network traffic for machine learning based detection (Fig. 4)	reaction with emergency plan to mitigate damage
Attack Indicators (exemplary, more detailed in Section 3.3):						
unusual access times and user account activities	-	unusual mounted USB devices	abnormal network packages sizes	unknown or uncertified software	abnormal network traffic	unusual outbounded network traffic

Fig. 3. Visualization of Kill Chain for $HCAS_1$

3.3 Defense Mechanisms and Attack Indicators for $HCAS_1$ along the Kill Chain

In this section, we discuss potential defense mechanisms and attack indicators for $HCAS_1$ along all phases of the Kill Chain (see Fig. 3). In **Phase 1** sensible files in the network architecture (such as documents about IT infrastructure) have to be protected for example with encryption methods. Furthermore, we propose to train and sensitize the employees of the objective regarding possible attack scenarios based on network protocols. Additionally, the defenders should monitor network traffic, log-ins and file accesses. All these actions, could prevent the attack. Here, possible attack indicators could be for example irregularities in log-ins or network accesses. When an insider attacks in Phase 1, indicators like unusual access times (e.g. late at night), anomalies in privileged user account activities or unusual requests or number of requests for a file can be noticed.

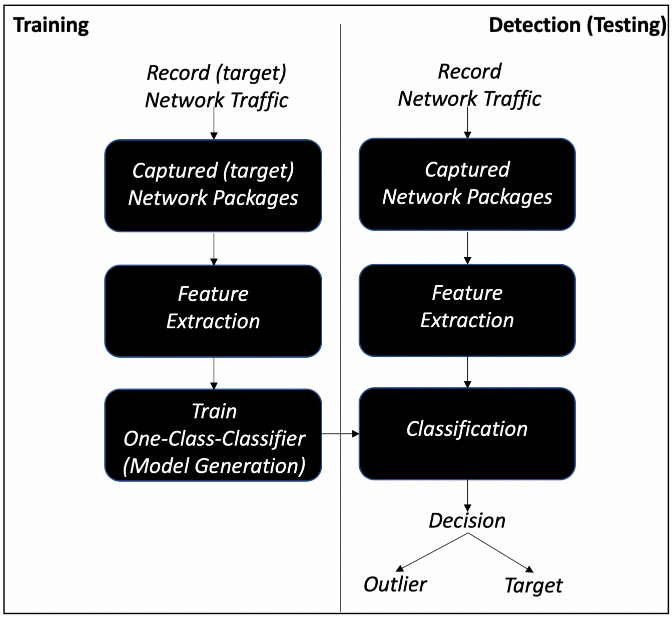

Training
Detection (Testing)
Record (target) Network Traffic
Captured (target) Network Packages
Feature Extraction
Train One-Class-Classifier (Model Generation)
Record Network Traffic
Captured Network Packages
Feature Extraction
Classification
Decision
Outlier
Target

Fig. 4. Pipeline for pattern recognition based anomaly detection approaches in Phase 6 for $HCAS_1$

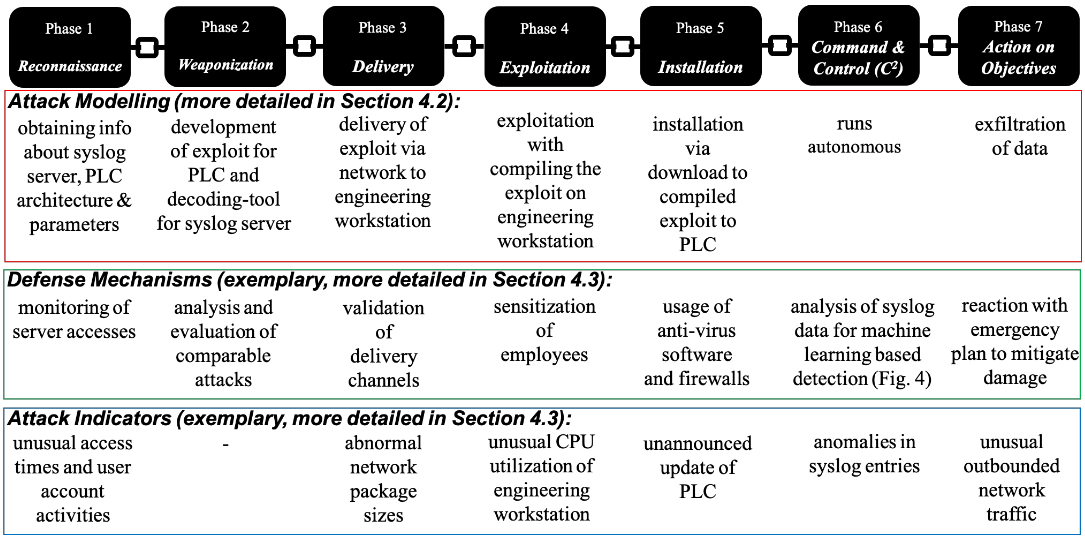
During **Phase 2** the defenders can not prevent the attack directly. However, the defenders should analyze and evaluate comparable attacks, to detect the malware or hidden communication in later phases of $HCAS_1$.

In **Phase 3** all possible attack vectors (in this case: delivery channels) have to be validated to secure that no malware infiltrates the system and to prevent the attack. Attack indicators for Phase 3 are for example unusually mounted USB devices or network flows.

In **Phase 4** we propose to sensitize the employees of the objective for possible attack indicators. Abnormal network packages sizes or data flows to the HMI caused by the download of the malware could be indicators for an attack.

In **Phase 5** the installed malware on the HMI can be possibly detected with anti-virus software. Unregistered, uncertified or unknown software on the HMI are indicators for malicious software.

During **Phase 6** the embedded hidden communication channel is used by the attackers. We propose to detect the hidden communication by statistical pattern recognition and machine learning approaches. Therefore, we recommend to record network data and to design a handcrafted feature space based on network package characteristics from that data. We propose to train statistical models only with authentic (target) network traffic to subsequently train a one class classifier which will be able to detect communication data with anomalies (outliers) in the observed network. A detection pipeline (based on this description) for pattern recognition based anomaly detection approaches is introduced and visualized in Fig. 4. A specific detection approach, designed for a comparable attack scenario with a comparable detection pipeline is published in Hildebrandt et al. (2020). For this phase, there could be different indicators for an attack like an abnormally large network package size or unexpected information flows. In **Phase 7** the defenders can not prevent the attack anymore. In this phase an emergency plan has to be activated to react to the attack scenario and to mitigate the caused damage. Here, unusual outbound network traffic is an indicator for Phase 7.



	Phase 1 *Reconnaissance*	Phase 2 *Weaponization*	Phase 3 *Delivery*	Phase 4 *Exploitation*	Phase 5 *Installation*	Phase 6 *Command & Control (C^2)*	Phase 7 *Action on Objectives*
Attack Modelling (more detailed in Section 4.2):	obtaining info about syslog server, PLC architecture & parameters	development of exploit for PLC and decoding-tool for syslog server	delivery of exploit via network to engineering workstation	exploitation with compiling the exploit on engineering workstation	installation via download to compiled exploit to PLC	runs autonomous	exfiltration of data
Defense Mechanisms (exemplary, more detailed in Section 4.3):	monitoring of server accesses	analysis and evaluation of comparable attacks	validation of delivery channels	sensitization of employees	usage of anti-virus software and firewalls	analysis of syslog data for machine learning based detection (Fig. 4)	reaction with emergency plan to mitigate damage
Attack Indicators (exemplary, more detailed in Section 4.3):	unusual access times and user account activities	-	abnormal network package sizes	unusual CPU utilization of engineering workstation	unannounced update of PLC	anomalies in syslog entries	unusual outbounded network traffic

Fig. 5. Visualization of Kill Chain for $HCAS_2$

4. HIDDEN CHANNEL ATTACK SCENARIO 2 ($HCAS_2$)

4.1 Attack Vector for $HCAS_2$

$HCAS_2$ describes an APT which uses the communication between a PLC and a System Logging Protocol Server (syslog server, see Parker (2016)) as a hidden communication channel. Therefore, the PLC is corrupted via its engineering workstation. The corrupted PLC embeds, encodes and sends the hidden messages to the syslog server and the server receives the hidden messages. This is a scenario to exfiltrate data from the PLC (for example the operating status) with a unidirectional communication from PLC to syslog server.

4.2 Kill Chain Attack Modelling for $HCAS_2$

We visualize the Kill-Chain for $HCAS_2$ in Fig 5. In **Phase 1** the attackers need to obtain information about the syslog server and PLCs in the IT-structure of the objective. Here, they need especially information about parameters of the server like network protocols (UDP/TCP), server version, encryption methods (e.g. SSL, TLS) and so on. A likely scenario is that an insider releases the information to the attackers or a sniffing attack on the enterprise level used to get important documents with information about the syslog server.

In **Phase 2** the attackers develop exploit code for the PLC which embeds, encodes and sends corrupted messages to the syslog server. So, the PLC must be able to manipulate syslog-messages and send the corrupted messages to the server. The corrupted PLC could send for example additional messages about critical states of the PLC or it could imitate PLC messages for the syslog server, this is a possibility to embed hidden messages on the syslog server. Furthermore, the attackers need to develop a decoding-tool to receive and decode the hidden messages from the syslog server.

In **Phase 3** the exploit is delivered to an engineering-workstation which has access to the targeted PLC. Therefore, the exploit is delivered via the network (or USB removable disk if an interface is available) to the engineering workstation. The decoding-tool is most likely delivered via USB removable disk somewhere in the IT-infrastructure on a system with access to syslog server.

The exploitation (**Phase 4**) is done when the exploit code for the PLC and the decoding-tool for the syslog server are compiled successfully. The installation in **Phase 5** is successful when the compiled exploit is downloaded from the engineering-workstation to the PLC and the compiled decoding-tool is running on a system with access to the syslog server.

Command and control (**Phase 6**) is usually used when the attackers need a bidirectional communication for their attack. In $HCAS_2$ there is no classic command and control phase because the attacker hides messages from the PLC in the syslog server which runs completely autonomous because it is an unidirectional communication (from PLC to syslog server) with no need of command and control.

In **Phase 7** the hidden messages are embedded and saved on the syslog server. Now, the messages can be decoded and exfiltrated.

4.3 Defense Mechanisms and Attack Indicators for $HCAS_2$ along the Kill Chain

We discuss different defense mechanisms and attack indicators along each phase of the Kill Chain of $HCAS_2$ in this section and it is visualized in Fig. 5. To prevent the procurement of information in **Phase 1**, it seems reasonable to monitor and evaluate server access to detect unusual user behavior or anomalies. This phase has equivalent attack indicators as Phase 1 in $HCAS_1$ in Section 3.3.

In **Phase 2** the attack can not directly be prevented. But defenders should evaluate state-of-the-art attacks for syslog servers to prepare defense mechanisms and to close potential security gaps.

For the **Phases 3, 4** and **5**, defense mechanisms equivalent to Section 3.3 can be used. Thus, defenders should check attack vectors and raise the awareness for the attack among the employees. Furthermore firewalls and anti-virus-software should be installed. In Phase 3 abnormal network package sizes to the engineering-workstation and network traffic with abnormal behavior can be noticed as indicators for an attack. An attack indicator in Phase 4 could be an unusual CPU utilization on the engineering-workstation during the compilation of the malware. Attack indicators for Phase 5 could be abnormal network traffic from engineering workstation to PLC or an unannounced update of the PLC firmware.

Phase 6 runs autonomous for $HCAS_2$. In this phase defenders should install a detector which analyzes the syslog data with anomaly detection approaches. In future, machine learning approaches could be trained with integer syslog data to detect anomalies in the syslog data set. The introduced detection pipeline in Fig. 4 could also be used for $HCAS_2$ under consideration of suitable training data (authentic syslog data for training). The chance to detect hidden communication increases by the amount of embedded information because this concludes to an increasing number of detectable anomalies in the syslog data. Anomalies in the behavior of a syslog server or anomalies in syslog entries are attack indicators for Phase 6.

If the attackers reach **Phase 7**, the hidden communication is embedded by steganographic techniques into the syslog server. Thus, steganalysis tools could possibly detect the hidden data on the syslog server. As for $HCAS_1$, abnormal outbound network traffic is an attack indicator this Phase.

5. HIDDEN CHANNEL ATTACK SCENARIO 3 ($HCAS_3$)

5.1 Attack Vector for $HCAS_3$

Our third exemplary attack scenario $HCAS_3$ uses Network Time Protocol server (NTP server, see Williams (2008)) as hidden communication channel to inject commands to a PLC. In this scenario the broadcast interval of an NTP broadcast server is used as a steganographic timing channel to send hidden commands to a PLC which is corrupted by a supply chain attack.

5.2 Kill Chain Attack Modelling for $HCAS_3$

The Kill Chain of $HCAS_3$ is presented in Fig. 6. During the reconnaissance (**Phase 1**) the attackers need to collect data about the NTP server (for example: version of the protocol, NTP broadcast available or unavailable and so on) used in the objective and information about which PLCs are NTP-broadcast clients. As already described for attack scenario $HCAS_1$ and $HCAS_2$, the attackers could use an insider or a sniffing attack on the enterprise level to obtain the required information.

In **Phase 2** the attackers have to prepare an corrupted update for the NTP server which embeds the corresponding broadcast intervals to inject the steganographic commands to the PLC. For the PLC, the attackers develop a decoder which processes the broadcast intervals of the server.

In **Phase 3** the attackers have to deliver the malware for the PLC and the NTP server. For the PLC a supply chain attack could be used to deliver a corrupted firmware to the PLC which is able to decode and interpret the NTP server broadcast intervals into commands. The NTP server will be corrupted through an update via the network. The exploitation (**Phase 4**) starts with delivery for the PLC and with downloading the corrupted update for the NTP server. The installation (**Phase 5**) for the PLC is done with delivery (supply chain attack). For the server, the phase is completed when the update has been downloaded successfully and is running on the server.

Phase 6 runs autonomous, because there is no bidirectional communication. It is only used for the hidden command injection via the embedded timing channel and this is an automatic process. The attack ends with **Phase 7** when the hidden channel finally injects the commands into the PLC.

5.3 Defense Mechanisms and Attack Indicators for $HCAS_3$ along the Kill Chain

The defense mechanisms and attack indicators for $HCAS_3$ are similar to $HCAS_1$ and $HCAS_2$ and are visualized in Fig. 6. In **Phase 1** defenders should monitor server accesses in the enterprise level to detect unusual server activities and we propose to protect sensible files with encryption to prevent data theft. For attack indicators of Phase 1 look at Section 3.3.

In **Phase 2** defenders should study attack scenarios for NTP server and basics for hidden timing channels.

In the **3rd Phase** defenders can not prevent the supply chain attack for the PLCs but they can install firewalls and anti-virus software to detect the corrupted update for the NTP server. Unusual NTP server update files in the network could be an attack indicator.

In **Phase 4** we propose to raise the awareness for corrupted updates via the network, that employees additionally check updates with anti-virus software. The download of the corrupted server updates creates unusual network traffic and package sizes, which are indicators for an attack in this phase.

The installation (**Phase 5**) of the corrupted update for the NTP server can only be prevented with the installed anti-virus software in Phase 3. The unusual server timeout during update installation is an attack indicator.

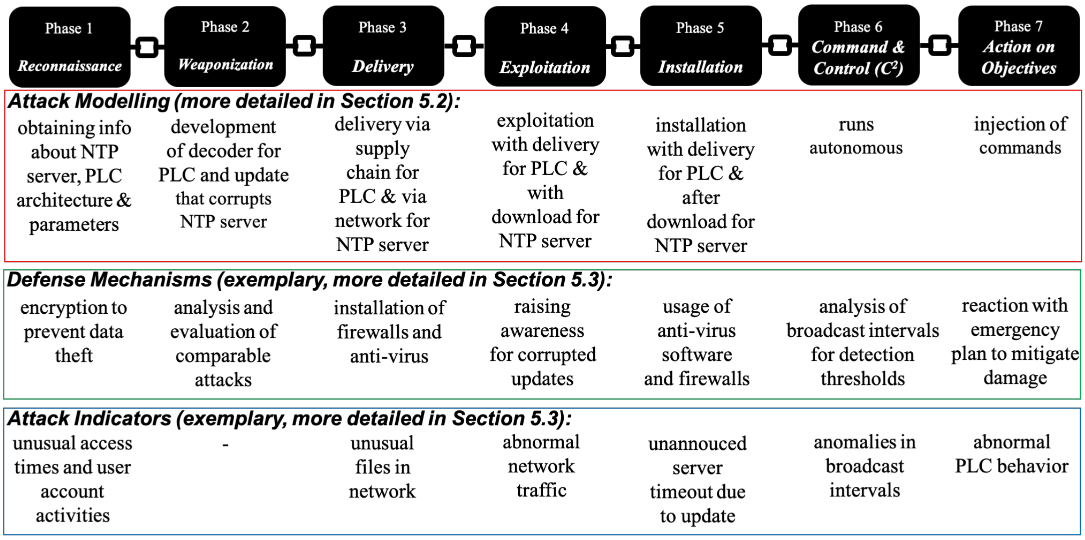
During **Phase 6** defenders should analyze the broadcast intervals to train statistical models or thresholds for anomaly detection approaches. Then it is possible to detect unlikely patterns in the broadcast intervals and the NTP server can be restored with a backup. Attack indicators for Phase 6 are the abnormal broadcast intervals from the NTP server.

Phase 7 could only be noticed when the command injection has already been successful and sends corrupted commands to the PLC. Due to this, the PLC has received unlikely commands that defenders must react with an activation of a developed emergency plan to shut down the corrupted parts of the network to mitigate the damage. Anomalies in PLC behavior is an attack indicator in the last phase of the attack.

6. CONCLUSION

In this work we propose the modelling of three exemplary attack scenarios with the Lockheed Martin **Kill Chain** to show how advanced persistent threats with hidden communication channels could took place in an industrial control system. Additionally, we discuss defense mechanisms and attack indicators along every phase of every proposed attack based on the presented Kill Chain analysis.

The goals of this work are to raise the awareness of the



Phase 1 *Reconnaissance*	Phase 2 *Weaponization*	Phase 3 *Delivery*	Phase 4 *Exploitation*	Phase 5 *Installation*	Phase 6 *Command & Control (C^2)*	Phase 7 *Action on Objectives*
Attack Modelling (more detailed in Section 5.2):						
obtaining info about NTP server, PLC architecture & parameters	development of decoder for PLC and update that corrupts NTP server	delivery via supply chain for PLC & via network for NTP server	exploitation with delivery for PLC & with download for NTP server	installation with delivery for PLC & after download for NTP server	runs autonomous	injection of commands
Defense Mechanisms (exemplary, more detailed in Section 5.3):						
encryption to prevent data theft	analysis and evaluation of comparable attacks	installation of firewalls and anti-virus	raising awareness for corrupted updates	usage of anti-virus software and firewalls	analysis of broadcast intervals for detection thresholds	reaction with emergency plan to mitigate damage
Attack Indicators (exemplary, more detailed in Section 5.3):						
unusual access times and user account activities	-	unusual files in network	abnormal network traffic	unannouced server timeout due to update	anomalies in broadcast intervals	abnormal PLC behavior

Fig. 6. Visualization of Kill-Chain for $HCAS_3$

community for these novel attack scenarios and to present, how Kill Chain attack modelling can be utilized to elaborate defense mechanisms and to reveal attack indicators for different phases of an attack scenario because current defense mechanisms are usually ineffective against those attacks. In future work we will perform simulations for our attack scenarios to create authentic test data (including authentic and corrupted communication data). With that created test data we are planning to design and evaluate detection approaches based on pattern recognition.

ACKNOWLEDGEMENTS

The work in this paper has been funded in part by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the research programme STEALTH-SZENARIEN under the contract no. `FKZ: 1501589C`. The authors would like to thank our project partners for discussions on the attack scenarios presented in this paper.

REFERENCES

ACROMAG (2005). *Introduction to Modbus TCP/IP.* ACROMAG Incorporated; Technical Report; `https://www.prosoft-technology.com/kb/assets/intro-modbustcp.pdf` , last checked:11/11/19.

Azuma, R., Daily, M., and Furmanski, C. (2006). *A review of time critical decision making models and human cognitive processes.* IEEE Aerospace Conference; DOI: 10.1109/AERO.2006.1656041.

Boehme, R. (2010). *Advanced Statistical Steganalysis.* Springer-Verlag Berlin Heildelberg, DOI: 10.1007/978-3-642-14313-7.

Bolton, W. (2015). *Programmable Logic Controllers.* Publishey by Elsevier Newnes; Boston; Programmable Logic Controllers (Sixth Edition); ISBN: 978-0-12-802929-9; DOI: 10.1016/B978-0-12-802929-9.00001-7.

Gonzalez, C. (2015). *What are Human Machine Interfaces and Why Are They Becoming More Important?* `https://www.machinedesign.com/iot/what-are-human-machine-interfaces-and-why-are-they-becoming-more-important`, last requested: 02/03/20.

Hildebrandt, M., Altschaffel, R., Lamshoeft, K., Lange, M., Szemkus, M., Neubert, T., Vielhauer, C., Ding, Y., and Dittmann, J. (2020). *Threat Analysis Of Steganographic and Covert Communication in Nuclear I&C Systems.* In Proceedings of IAEA ICONS 2020: International Conference on Nuclear Security: Sustaining and Strengthening Efforts, 10-14 February 2020, Vienna, Austria, `https://event.do/iaea/a/#/events/3301/f/29007`.

Hutchings, E., Cloppert, M., and Amin, R. (2011). *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.* Leading Issues in Information Warfare & Security Research; ISBN: 978-1-908272-08-9.

Kushner, D. (2013). *The Real Story of Stuxnet.* IEEE Spectrum; DOI: 10.1109/MSPEC.2013.6471059.

Mazurczyk, W., Wendzel, S., and Cabaj, K. (2018). *Towards Deriving Insights into Data Hiding Methods Using Pattern-based Approach.* ARES 2018, 13th International Conference on Availability, Reliability and Security; Hamburg, Germany, August 27 - August 30, ISBN: 978-1-4503-6448-5.

OPC-Foundation (2008). *Unified Architecture.* Technical Report; `https://opcfoundation.org/about/opc-technologies/opc-ua/`, last requested: 02/03/20.

Parker, J. (2016). *What is Syslog, includuing Linux and Windows Servers, Ports and more.* `https://www.pcwdld.com/what-is-syslog-including-servers-and-ports`, last requested: 02/03/20.

Schmidbauer, T., Wendzel, S., Mileva, A., and Mazurczyk, W. (2019). *Introducing Dead Drops to Network Steganography using ARP-Caches and SNMP-Walks.* ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and Security August 2019 Article No.: 64 Pages 1–10 DOI: 10.1145/3339252.3341488.

Williams, R. (2008). *How an NTP Server Works.* `https://www.galsys.co.uk/time-reference/ntp-time-servers/how-ntp-works.html`, last requested: 02/03/20.