A Mobile Honeypot for Industrial Control Systems

Master-Thesis von Shreyas Srinivasa aus Bangalore, India
Tag der Einreichung:

1. Gutachten: Emmanouil Vasilomanolakis
2. Gutachten: Prof. Dr. Max Mühlhäuser
3. Gutachten:

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Telekooperation
Prof. Dr. Max Mühlhäuser

A Mobile Honeypot for Industrial Control Systems

Vorgelegte Master-Thesis von Shreyas Srinivasa aus Bangalore, India

1. Gutachten: Emmanouil Vasilomanolakis
2. Gutachten: Prof. Dr. Max Mühlhäuser
3. Gutachten:

Tag der Einreichung:

Zusammenfassung

The number of attacks towards the cyberspace is constantly increasing with their complexity. All areas including a smartphone to a nuclear power plant are targeted by hazardous malware. The advent of better smartphones has changed the way people interact and also leverage its resources to constantly provide better productivity to users. Enterprise systems like nuclear power plants have complex network infrastructure and also need to be highly secure. Over the years such systems have been infiltrated by perilous malware like Stuxnet that had capabililtes to shut down critical system thereby causing hazards. Detecting such malware is very complex and challenging because of their propagation behavior. Network security devices like Firewalls and Intrusion Detection Systems (IDS) provide front line defense against cyber attacks in enterprse networks. These systems work on the Rules and Signature based frameworks to identify and to detect malicious traffic. An active mechanism that provides alert mechanisms on malicious packets and features an analysis framework is necessary. Honeypots are active monitoring systems that act as decoy mechanisms for attackers. As Honeypots simulate target devices with vulnerabililtes, every connection attempt to it can be considered as an attack. Critical

Enterprise Infrastructure like Nuclear Power Plants, Water Treatment and Distribution Plants and Gas Distribution Plants form an important part of everyday necessities of mannkind. The above infrastructure are classified as Industrial Control Systems(ICS). ICS are a part of Enterprise Infrastructure that invlove Programmable Logic Controllers(PLCs), Command and Control systems, Real Time Units and a private network of hosts. The systems are managed over wide geographical areas which expose these devices to the external networks. Preventing these systems from being compromised is complex and paramount. A Honeypot framework which provides attack detection, analytic capabilities along with signature generation mechanisms can be considered for monitoring in ICS systems. This strategy is proved effective in infrastructire where both IDS and Honeypots work in co-ordination.

Smart phones today provide great resources and computation power. They also provide superior networking capabilities and flexibility. The greatest advantage of these devices are the mobility and ease of use. We leverage these features of smart phones to develop a Mobile Honeypot mechanism for ICS systems. We extend HosTaGe to HosTaGe ICS, an Android based mobile honeypot to accomodate ICS profiles like a Nuclear Power plant. ICS systems are further classified to SCADA (Supervisory Control and Data Aquisition) systems. SCADA forms a standard for majority of ICS systems deployed worlwide.

The protocols and the target systems that are specific to ICS SCADA systems like Modbus and S7 are also implemeted to simulate the behavior of end systems. The behavior and response mechanisms are also implemented to facilitate the attacker to pursue his attacks. Malware detection is a crucial challenge that involves studying the behavior and progation techiniques of malware. HosTaGe ICS creates the environment for the propagation of STUXNET, a popular malware that can bring down ICS systems and also detect its propagation. The results are further verified by the Virustotal database.

The attack strategies used by modern day attackers involve distributed attacks, single level protocol attacks, multi level protocol attacks and payload level attacks. Multistage attacks refer to attacks that originate from the same source and attempt to exploit different types of protocols within a small window of time. It is one of the prevailing and effective strategies used by attackers today to perform social engineering and compromise end systems. HosTaGe ICS features the detection mechanism for multistage attacks. This detection mechanism also identified interesting results during the evaluation course of the thesis.

We previously discussed about the network infrastructure where IDS and Honeypots work together. HosTaGe ICS incorporates a signature generator module which creates signatures based on the attacks recorded with Modbus, S7 and SMB protocols. The signatures that are generated can be mounted to the Bro Network Monitor to check packets for signature. This functionality can be leveraged where Bro and HosTaGe are deployed on the network to be monitored.

Inhaltsverzeichnis

1 Introduction

Mobile devices today have better communication capabilities. They enable dynamic and faster communication. Users are able to access internet and web applications through their smart phones anywhere, anytime. Smarter applications offer better social interaction and online presence to the users. This creates an urge to stay connected and be online seamlessly to be updated.Public infrastructures like airports, coffee shops, shopping malls provide free access to their networks to its customers to facilitate their connectivity and of course, for some information exchange. With free access to networks, attackers are now concentrating on the possibility of exploiting users in the same network. Securing open networks is very challenging and complex. It is however possible to detect these attacks. A pro-active approach is a better way for detecting the attacks.

Huge industries like nuclear power plants, water treatment and distribution plants, manufacturing plants have many complex critical machines and require constant monitoring. They rely on process automation on these machines and are dependent on sensors for making this automation possible. This sensor-to-machine-to-human communication and automation is achieved with the help of PLCs[21] or Programmable Logic Controllers. This communication is usually not secure and is open to attacks. As this hardware has limited computing resources, encryption of data is an expensive option. There have been many attacks detected over the years on SCADA[1] ICS, most notable being STUXNET[6]. Securing and detecting attacks in these networks is necessary as it is responsible for communication in critical machines. Failure of such machines could cause a devastation to the environment and human life because of the wide spread use of PLCs in infrastructures like airports, coffee shops and also in prisons.

There are two approaches for detection of attacks. One is by using a NIDS[13] (Network Intrusion Detection System) and the other is by using Honeypot[12] . NIDS are installed on the server machines or hosts. The requests are scanned and analyzed for exploit-forged packets before they are sent to the server. NIDS are suitable for systems with high resources. The Honeypot approach, rather could be used where there are lesser resources. The idea behind Honeypot, is to pose as vulnerable hosts connected to the network, which could be tempting for exploits, thereby trapping the attacker by collecting as much information possible to backtrack, or good enough to detect that the network is under attack.

1.1 Motivation

The applicability of a Honeypot in a mobile environment is prodigious, considering the public network infrastructure services offered. Network connectivity has become more of a necessity than a luxury, as technology is continuously evolving. Better services, data management and accessibility draw a lot of users having online space and in the need to stay connected. This need is rendered by some businesses and public infrastructure like airports, malls and cafeterias. With smart phones, people have the power to stay connected and do the majority of the tasks efficiently at their fingertips. Mobile devices today are considered personal devices because of the capability to store, share and process private data. This data is valuable and private to a user and has to be secured. Connecting to public networks can result in lot of vulnerabilities, as there is not always security considered in public networks. With the help of scripts crafted to exploit these vulnerabilities, an attacker can exploit users personal data.

Attacks are not limited to the above protocols. Airports, malls, enterprise hotels and huge industries use PLCs[21] (Programmable Logic Controllers) as for many applications such as conveyor belts, elevators, lighting control systems, fire and safety detection systems in order to automate the tasks quickly without human intervention. PLCs can be programmed logically to specify the methods to be called, based on inputs provided by sensors. SCADA (Supervisory Control and Data Acquisition) is a system operating with coded signals over the communication channels so as to provide control of remote equipment like PLCs. A study made by DELL[1] showed that the attacks on Industrial components like PLCs doubled over

the years, and even more dangerously, such incidents going unreported. The research found a 100 percent increase in attacks against industrial control systems like SCADA. Figure 1 gives an understanding of the Key SCADA Attack Methods.

It shows that about half of the total attacks were based on improper assignment on bounds of a memory buffer, improper input invalidation, vulnerabilities in credentials management. These vulnerabilities pose as a huge threat to ICS. Figure 2 represents the number of attacks performed over the months. There is a steep increase in the number of attacks performed over the months, expressing the need to safeguard ICS systems and also detect these attacks.

[1] http://www.schneider-electric.com/solutions/ww/en/med/20340568/application/pdf/1485se-whitepaper-letter-scadaoverview-v005.pdf

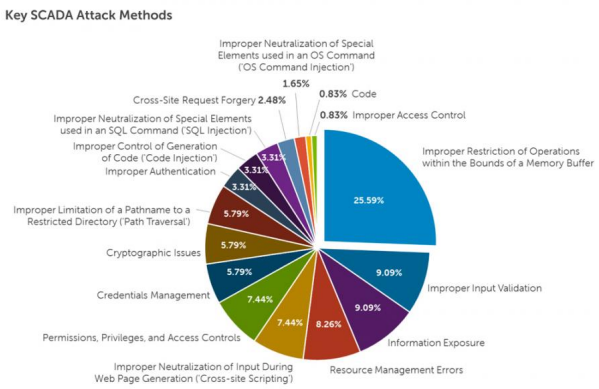

Abbildung 1: SCADA attack methods[1]

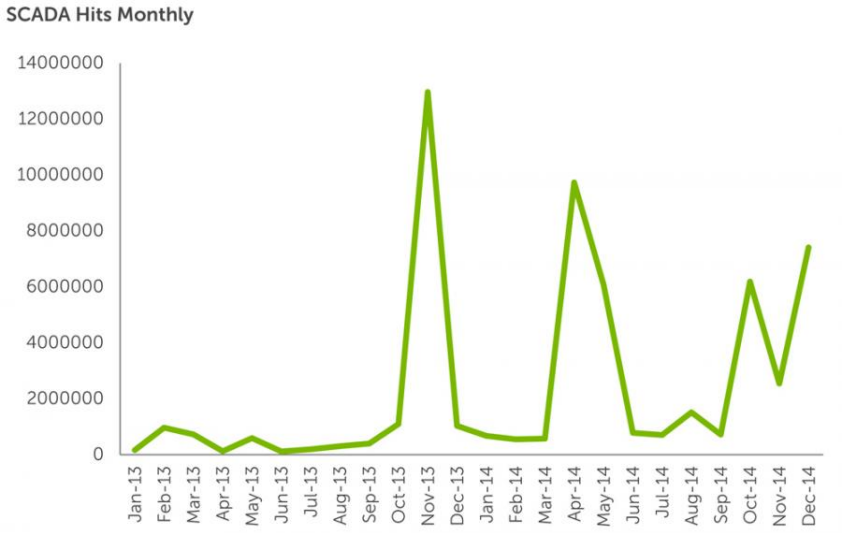

Abbildung 2: SCADA hits on a monthly basis.[1]

The majority of industrial systems today use SCADA for controlling and automating their processes. Securing these devices is as much important like any other hosts in the network because these devices are programmable and could affect the normal automatized working. STUXNET[6], a computer worm discovered in 2010 was designed to attack industrial programmable logic controllers (PLCs). STUXNET reportedly compromised PLCs in power plant at Iran. The design and architecture of STUXNET is not domain-specific and it could be forged for exploiting modern SCADA and PLC systems.

1.2 Contribution

This theses aims at identifying and detecting the ICS SCADA attacks using a low interaction mobile Honeypot platform, using which master and slave profiles will be simulated. An analysis of the communication paradigm and the security loopholes of a SCADA ICS system is made, to simulate the services offered by the system. The thesis also concentrates on contributing to many security related research questions of SCADA ICS systems like identifying the targets, analyzing the malware, assessing the consequences and defending ICS systems. Finally, the a mobile ICS Honeypot, HosTaGe ICS is implemented as an extension for HosTaGe.

1.3 Outline

This thesis topic also aims at adding more capabilities to detect attacks through different malware, mainly focussing on simulating industrial level SCADA PLC to determine malware attacks on them. The rest of the expose is structured as follows. Section 2 will specify the requirements to develop the protocol emulation for mobile Honeypot. In Section 3,

related work in the area of mobile Honeypot and SCADA Honeypot are discussed. Section 4 describes a proposed system for a mobile Honeypot for ICS systems and Section 5 concludes with a time plan for the thesis.

2 Background - ICS SCADA and Mobile Honeypots

ICS (Industrial Control Systems) form a dominant portion in present day industries. Strange, yet astonishing, the fact that ICS is also a part of everyday life is also true. ICS components include actuators, sensors, networking devices, controlling systems and PLC's . The sensors form a major part of ICS as they provide continuous feed of critical information which is used to automate and control other systems. The other important component is the PLC. This interface allows a programmer to implement a logic to automate the systems based on the data received from sensors. There are a few different kinds of ICS. One of the major types is SCADA (Supervisory control and data acquisition) which is deployed on geographically widespread and controlled using a central location. Examples to this type include nuclear power plants, water distribution , power distribution where there is a need constant monitoring and critical automation. SCADA systems are mainly deployed where is a need for alarm systems. The other kind of ICS system is the Distributed Control Systems (DCS). On the contrary these systems are not centralized, but distributed across a network. We shall focus more on SCADA ICS systems as they are being deployed in major infrastructures today.

Infrastructures discussed above have a lot of components and devices which need constant communication between them. // Complete this para

2.1 ICS SCADA

SCADA is an industrial automation control system at the core of many indutries today including Energy, Oil and Gas, power, Water and Recycling , Manufacturing and many more. They are used by both private sector industries and the public sector service providers. It provides the benefit of simple configuration and usability. The basic architecture of

SCADA involves communication of information from sensors or manual inputs to PLCs or RTUs. These PLCs process the information as per the logic deployed in them and then forward this information to workstations/servers running SCADA applications. Figure 3 describes the basic architecture of a SCADA system which contains sensors, PLCs, master hosts and the controller hosts. Data is fetched from the sensors and is processed on PLCs. This data is communicated further to other PLCs in the network for further processing. The HMI displays the processed data to the user.

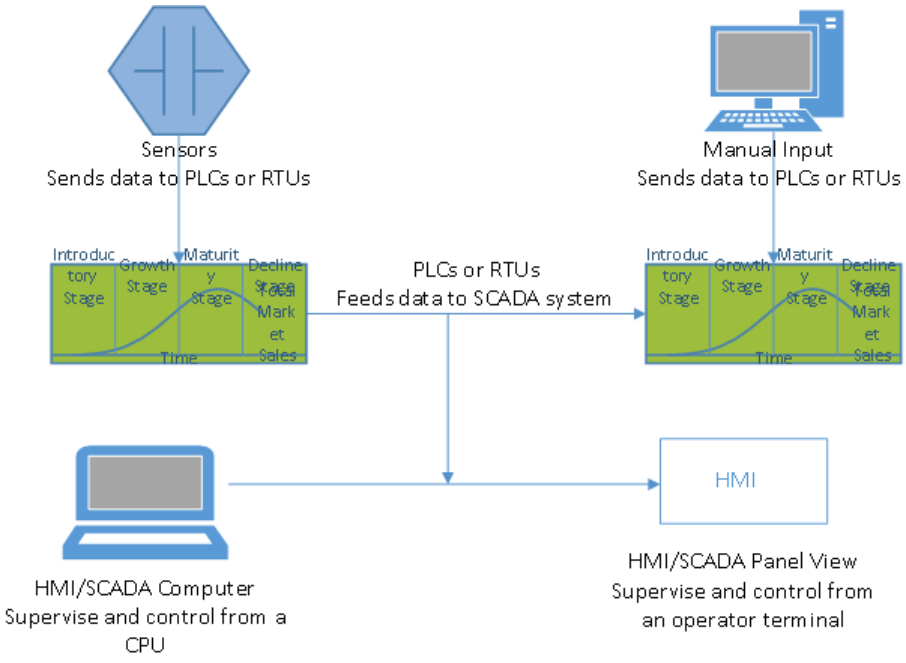

Abbildung 3: SCADA Architecture

SCADA systems involve control components and network components. The following is a list of control components in SCADA:

- **Remote Terminal Units (RTU):**These units connect to sensors in the process and convert sensor signals to digital data. They have telemetry hardware capable of sending digital data to the supervisory system, as well as receiving digital commands from the supervisory system. RTUs often have embedded control capabilities in order to accomplish boolean logic operations.

- **Programmable logic controller (PLCs):** These devices connect to sensors in the process and convert sensor signals to digital data. PLCs have more sophisticated embedded control capabilities than RTUs. PLCs do not have telemetry hardware, although this functionality is typically installed alongside them. PLCs are sometimes used in place of RTUs as field devices because they are more economical, versatile, flexible, and configurable.

- **Telemetry system:** It is typically used to connect PLCs and RTUs with control centers, data warehouses, and the enterprise. Examples of wired telemetry media used in SCADA systems include leased telephone lines and WAN circuits. Examples of wireless telemetry media used in SCADA systems include satellite (VSAT), licensed and unlicensed radio, cellular and microwave.

- **Data and Control Server:**A data acquisition server is a software service which uses industrial protocols to connect software services, via telemetry, with field devices such as RTUs and PLCs. It allows clients to access data from these field devices using standard protocols.

- **Human Machine Interface (HMI):** It is the apparatus or device which presents processed data to a human operator, and through this, the human operator monitors and interacts with the process. The HMI is a client that requests data from a data acquisition server.

- **Historian software:** A software service which accumulates time-stamped data, boolean events, and boolean alarms in a database which can be queried or used to populate graphic trends in the HMI. The historian is a client that requests data from a data acquisition server.

Different network characteristics exist for every layer within the control systems. The network topologies vary by vendors or manufacturers and also on different implementations. Modern day SCADA systems are open to Internet communication and enterprise integration can be achieved. The control networks work in hand with the corporate enterprise networks to better manage and control the systems from outside networks. The following are the major network components of an ICS network:

- **Fieldbus Network:** The fieldbus network links sensors and other devices to a PLC or other controller. Use of fieldbus technologies eliminates the need for point-to-point wiring between the controller and each device. The devices communicate with the fieldbus controller using a variety of protocols. The messages sent between the sensors and the controller uniquely identify each of the sensors.

- **Control Network:** The control network connects the supervisory control level to lower-level control modules.

- **Communications Routers:** A router is a communication device that transfers messages between two networks. Common uses for routers include connecting a LAN to a WAN, and connecting MTUs and RTUs to a long-distance network medium for SCADA communication.

SCADA applications help in monitoring, analysing the data to help the device controllers and operators work efficiently. Modern SCADA systems allow real time data from the plants to be accessed from anywhere in the world. This also means that it provides attackers an opportunity to exploit this data and availability. Exploiting SCADA systems can cause catastrophic as it may result in huge damage to the environment and people in the plant. We try to identify the attacks and exploits that could be made and detect them using a mobile Honeypot.

2.2 Security Perspective of SCADA ICS

ICS SCADA systems are highly distributed. They are used to control and manage geographically dispersed plants, often scattered over thousands of kilometers. In these areas centralized data acquisition and control are critical to system operation. They are applicable in distribution systems such as water distribution and wastewater collection systems, oil and natural gas pipelines and electrical power grids.on systems. A SCADA control center provides centralized monitoring and control for field sites over long-distance communications networks, including monitoring alarms and processing status data. Based on information received from remote stations, automated or operator-driven supervisory commands can be pushed to remote station control devices, which are often referred to as field devices. Field devices control local operations such as opening and closing valves and breakers, collecting data from sensor systems, and monitoring the local environment for alarm conditions. The control center is responsible for managing and controlling the devices at the

field site and thus there is a need to have a critical communication network between them. This is usually established through the Modbus TCP/IP over the Ethernet. It is usually advised [15] to place the SCADA devices on a network that is not physically connected to any other networks.

// Refer to paper Plausible Solution to SCADA security for more info

2.3 Honeypots

A Honeypot is a decoy server or a system in a network which is closely monitored for adversaries. It is also defined as: *A Honeypot [14] is an information system resource whose value lies in unauthorized or illicit use of that resource*. They are mostly deployed inside firewalls, but they could be deployed in any part of the network. It is designed to be a system with vulnerabilities and services that are offered by a real target system. Any attempt to connect to these systems could be considered as an attack. All the activities are logged and further traced. The general idea is that once an adversary detects a vulnerable system and tries to attack it, he would come back with more sophisticated attacks. The initial part of discovery and knowing the general services and loopholes is called system social engineering. Honeypots provide are active monitoring components that wait for attacks and respond to the attacks by luring the attacker to pursue more. There are certain main functionalities that the Honeypots must possess in order to perform their main functionality.

1. Honeypots must simulate the system that they are intend to focus on. This gives the attacker a feeling of approaching a real system. The Honeypot may simulate the complete functionality of the system or just the services offered by the system.

2. A proper response mechanism which keeps the attacker engaged to the Honeypot. This makes better logging of the attack and also provides more data to analyze the attacks.

3. It mainly has three perspectives. Firstly, an attacker perspective, by posing as a vulnerable system; second an administrator who can log identify and log the attacks made by the attacker and third, being able to present and analyze the attacks logged by the administrator.

4. Honeypots must not induce additional load on the infrastructure. Honeypots must be designed to be lightweight and having no influence on the production systems.

5. Honeypots are basically exposed to exploits, threats and malware. It is very important to see that this data is not leaked through the network which can later infect other systems in the network.

6. Based on the previous condition, Honeypots must be robust to withstand the attacks and exploits and not fail.

It is very clear are valued because of the interaction mechanism that they provide for any communication request. They can be used to study and gather exploits, malware and threats in an early attack phase. There are many advantages that one could consider for using Honeypots as an additional security monitor. There are various advantages of Honeypots:

- **Effective Data Sets:** Honeypot collects data only when there is a communication requested with it. The data collected at Honeypots may not be immense but is good enough to analyze and detect attacks. The logs provide information about the attacker IP, time of attack and protocol used to carry out the attack. This makes lesser false positives.

- **Reduced False Positives:** Among other security approaches like IDS and firewalls, false positives are quite common. The biggest challenge is to reduce false positives. Honeypots reduce false positives or could be designed to reduce false positives. Any communication with the Honeypot is unauthorized. This makes Honeypots efficient in detecting attacks.

- **Catching False Negatives:** Honeypots have advantages over signature based detection systems. Signature based systems do not categorize unknown attacks. They rely on a signature system to be updated on their local database to identify and detect uknown attacks. The probability of a detecting a new exploit is low. Honeypots detect all attacks irrespective of their signatures, hereby increasing the possibility of detecting new attacks.

- **Encrypted Communication:** The current standards in Transport layer includes using encrypted TLS communiction between nodes. Some attacks fail to detect because of the encrypted data and communication. All enterprise employ secure protocols like SSH,IPSec, HTTPS, TLS in their infrastructure. This may cause problems in detecting exploits and analyzing the attacks later. Honeypots solve this issue as they are end points in the communication. The hosts directly interact with the node and hence all the traffic and data can be decrypted and analyzed later.

- **Compatibility to new architecture:** Technology evolves every moment. It is very essential to consider future compatibility with newer standards and technology. Most of modern day IDS or firewalls are not compatible with IPv6 which promises to be the next standard on Internet addressing. Honeypots can be made compatible to newer standards and technology as they are not mediators or devices but act as end points. However, devices could be simulated by Honeypots.
- **Flexibility:** Honeypots can be deployed locally or open to the external network. Honeypots could be deployed on any environments based on the requirements. Honeypots could be used to simulate any software, hardware, servers, workstations and devices.
- **Minimal Resource Consumption:** Honeypots can run on low resource machines as they are just simulations and are may not depict full functionality of the system simulated. Honeypots today can run on smartphones as they possess the required resources which are good enough to run a Honeypot.

There has been extensive research going on in the field of Honeypots. This section describes related works on Honeypots.

2.3.1 Types of Honeypots

Honeypots can be classified into two types based on the ability of the attacker to interact with the application or services. They can be categorized to High-Interaction Honeypots and Low-Interaction Honeypots. This classification is mainly based on the Honeypot's interaction with the attackers. Highn Interaction Honeypots typically composed of the actual device, its operating system and all the applications that run on that device. In short, the exact machine is used as a Honeypot with all its services. This provides better interaction as we are using the device itself as a Honeypot. There are also better chances that based on the vulnerability known, all the exploits work on the device. The main advantage of such Honeypots is that it is the machine itself that is being exposed and has greater chances of attracting attackers. The disadvantage would be that if the Honeypot is completely compromised, then it has to be rebuilt in order to log other attacks. The validity of such Honeypots is not guaranteed. A low interaction honeypot on the other hand is a software based or simulation based Honeypot approach. The system to be subjected to attack is simulated by the Honeypot along with its main services. The Honeypot can run on any system, for example it can run on a Linux machine and simulate a Honeypot for a Windows IIS server. It can simulate or mimic the network stack and the operating system of the targetted system. All connections and communication with this device is logged. The advantage of low interaction Honeypots is that they are completely flexible and easy to maintain. Low interaction Honeypots are also likely not to get compromised as they just mimic the services or in short the basic communication mechanism. It is on the researcher to design these Honeypots accurately to get productive results.

2.3.2 Honeynets

Honeynets [5] are a networked collection of honeypots that look like common network services and servers. It could be a collection of Honeypots depicting as a Domain Controller, web server, application server, file server and so on which provide a facade of a enterprise network. Honeynets usually consist of high -interaction honeypots, low - interaction honeypots, or a combination of both. Using high interaction Honeypots only for this approach would be more expensive. Honeynets are placed behind a Honeywall , which acts as a bridge to the honeynet. It includes network monitoring, packet capture, and IDS capabilities.

2.3.3 Mobile Honeypots

Modern day smart phones are context sensitive and collect a lot of data from the users perspective. This data is both private and critical to the user. There is a need to protect this data. The phones also have enormous computing resources in terms of hardware and also efficiently built software kernels that are capable of processing huge data. We are also able to stay online every moment and can connect to various hotspots providing us Internet facilities to stay connected. This also is huge security concern as the networks and the apps that are deployed on our phones may not be secure and leak sensitive data with respect to the user. The power of mobility, computing resources, usability and flexibility make Mobile

devices a good platform to host low interaction Honeypots. Such capabilities make it possible to host a low interaction Honeypot on the devices.Some researchers believe that Mobile Honeypots are still not well defined and could be used to define either a probe deployed on a mobile device or on a mobile operating system. It can also be defined for a system that is controlled in the network of mobile devices [20].

Early research on Mobile Honeypots focused only on Bluetooth communications[5,17]. The continuous advances in the field of smartphone technology has enabled better opportunities towards Honeypot research on smart phones.

//Write about Mobile Honeypots There has been existing work that focused on detection of mobile specific malware. The first to discuss the idea of a Honeypot for smartphones were Mulliner et al., by providing the initial ideas, challenges and an architecture for their proposed system[10]. Nomadic Honeypots[9] concentrates on mobile specific malware and also trades off with a lot of personal information.

- **HoneyDroid**) HoneyDroid [10] is a smartphone Honeypot for Android operating system which claims to be the first ever Honeypot in the Mobile Honeypots category which makes use of smart phone hardware to host the Honeypot.It is built on a Linux micro-kernel and is customized to impose restrictions on the Android operating system for monitoring its activities. The architecture is comprised of a Event Monitor, to monitor active connection requests and also system calls in the kernel level; Filters to mitigate any attempts of malware trying to affect the system and a log software to log all the activities. This Honeypot is also focused on detecting attacks from apps installed in the device which try to infiltrate the kernel for gaining unauthorized access. The system also involves virtualization which enables simulation of various services. This could also result in an overhead, hereby causing a signature which can be detected by attackers and malware. However, the direction of HoneyDroid was to introduce the concept of Mobile Honeypots.

- **Cellpot:** Cellpot [8] concentrates on detection and defence of attacks in the cellular network. It comprises of a collection of Honeypots, or Honeynets that are deployed on mobile phones. Cellpot consists of applications like SMS spam prevention, mobile phone theft and malware protection. The Honeypot mainly is concentrated towards Small Cells(cite from paper), wireless infrastructure deployed in customers site and operated in licensed bands. The main use of Small cells is to support the need of coverage and capacity. These points are a good place to deploy the Honeypots to detect malware and other intrusion attacks. Denial Of Service is the most common category of attack in the area of cellular networks, and with the help of few devices,this attack can be executed successfully. Introducing a Honeypot approach for detecting such attacks at small cells is a feasible solution.The concept of Cellpot is to detect, collect intelligence and mitigate threats against the cellular network directly on the base stations. Further, it has the ability to deploy countermeasures against detected threats, and enables a wide area of applications. It provides a good platform for mobile network operators to deploy and run additional applications to reduce signaling.

- **Nomadic Honeypots:**Nomadic Honeypots [9] propose a concept that provides an infrastructure to enable mobile network operators to collect threat intelligence directly on smartphones. It was the first proposed Honeypot on a smartphone platform. It places the smart phone user as a key role and requires the device to be used in a normal way. The Honeypot is deployed on the smartphone in a separate partition which serves the purposes of mediating all communication of the mobile OS, hosting infrastructure for data collection, facilities for snapshots and logging and lastly provides a secure backchannel for the operator. The main partition hosts the mobile OS and strict isolation is followed between the partitions to ensure the robustness of the Honeypot infrastructure. The operator can further use the collected data to gain intelligence on mobile threats. Snapshots of the mobile OSâĂŹs file system to do an offline forensic analysis of attacks could be easily provided which can gain an thorough insight on the nature of the threats and use the findings to protect his customers.

- **HosTaGe:**[17],[18] is an Android App which acts as a Mobile Honeypot, determined to detect malicious networks and probe for attacks. It is user centric and aims at creating security awareness to its users. The results obtained in this process are synchronised with a global repository and also can be shared locally through bluetooth. The current version has capabilities of emulating as Windows, Unix, Apache Server, SQL and Paranoid host. Attacks through HTTP, SMB, SSH, HTTPS, Telnet and FTP can be identified.

2.4 SCADA Honeypots and Related Work

Analysing the security concerns of ICS SCADA systems and the advantages of Honeypots, a solution could be implemented to combine the needs and features. SCADA Honeypots could be deployed in ICS Networks for monitoring and analysis. They act as an additional line of defense providing warnings and notifications for attacks. Designing a SCADA Honeypot involves studying the architecture of the SCADA systems and the components, protocols involved in communication and processing of data. Further, as discussed before, SCADA networks comprise of hardware devices like PLCs and RTUs which play a very critical role in processing and communication of data. SCADA systems rely on PLCs for data processing. If PLCs are targeted by attackers to compromise their working, it could bring down the entire plant, hereby resulting in a huge catastrophe. Modern day PLCs offer TCP/IP communication which can used to control and manage the data flow

between other PLCs and control servers. On investigating attacks that have occured in the past, STUXNET a malware, was found to be injected in a Nuclear Enrichment Facility in Iran. STUXNET was found to be injected into the network using a USB drive to one of the host control systems. The malware spread from that system to other systems through intranet and remained hidden from operators. STUXNET was able to interfere with the working of a PLC that controlled centrifuges and managed to compromise the conditions on which the PLC depends. It was only by the observation of an operator that the PLC was causing the centrifuges to run more fast than usual was detected. But nobody could determine what caused the centrifuges run abnormally. Detecting such kinds of attacks is not only complex but also very necessary.

Such kind of attacks cannot be detected neither by signature based systems, nor by firewalls. Some organisations took initiative to design Honeypots for SCADA systems. They are elaborated in futher secctions.

2.4.1 SCADA Honeynet

SCADA Honeynet Project[3] is a project aimed at building Honeypots for industrial networks. It was the theb first of the type. SCADA Honeynet was designed to simulate the PLCs and detect attacks performed on them.The short-term goal of the project was to determine the feasibility of building a software-based framework to simulate a variety of industrial networks such as SCADA, DCS, and PLC architectures. It provided scriptable industrial protocol simulators to test actual protocol implementation. The design was a ingration of stack level, protocol level, application level and hardware level. The Honeypot was carefully designed to cover all the services offered by the SCADA systems, including the networking devices like routers and a direct serial device.

2.4.2 Trend Micro SCADA Honeypot

Trend Micro a global security software company conducted an experiment[2] to detect attacks on SCADA by setting up 12 Honeypots in 8 countries. The Honeypots camouflaged a municipal water control system based on SCADA that was connected to the internet. Attacks were basically focussed on meddling with the pump system. The objective of this experiment is to assess who/what is attacking Internet-facing ICS/SCADA(Industrial Control Systems) devices and why. In addition, the research set out to identify if the attacks performed on these systems were targeted, by whom, and for what purpose.

The Honeypot architecture design used a combination of high-interaction and pure-production Honeypots. A total of three Honeypots were created to ensure as much of the target surface as possible. All three Honeypots were Internet facing and used three different static Internet IP addresses in different subnets scattered throughout the United States.

2.4.3 Digital Bond

Digital Bond a security research and consulting firm created a Honeypot system that comprised of two virtual machines. The Honeypots are open source. One of the virtual machine acts as a PLC Honeypot and the other is a monitoring engine that logs all the traffic information. This system is also called a Honeywall. Honeywalls can also be used to monitor High Interaction PLC Honeypots. The Honeywall comprises of Snort IDS and signatures with respect to PLC. The services that are simulated are FTP, TELNET, HTTP, SNMP and Modbus TCP.

2.4.4 Conpot

Conpot[3] is a low interactive server side ICS Honeypot designed to be easy to deploy, modify and extend. It provides a range of common industrial control protocols capable of emulating complex infrastructures to convince an adversary that he just found a huge industrial complex.To improve the deceptive capabilities it also provides the possibility to server a custom human machine interface to increase the Honeypots attack surface. The default configuration of Conpot simulates a basic Siemens SIMATIC S7-200 PLC with an input/output module.

2.5 Modbus

Modbus denoted IETF RFC 2026 is a serial communications protocol published by Modicon for using in its PLCs. It is now a standard that connects industrial devices together. The basic configuration involves connecting a SCADA supervisory

[2] http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf
[3] http://conpot.org/

control system to a PLC or RTU. Many of the data types are named from its use in driving relays: a single-bit physical output is called a coil, and a single-bit physical input is called a discrete input or a contact. The device requesting the information is called the Modbus Master and the devices supplying information are Modbus Slaves. In a standard Modbus network, there is one Master and up to 247 Slaves, each with a unique Slave Address from 1 to 247. The Master can also write information to the Slaves. Modbus TCP/IP specification was introduced to Modbus to integrate corporate intranet with PLC systems. This made the network better manageable, scalable and also cost-effective.Modbus TCP/IP offers many advantages:

- **Simplicity:** The TCP is wrapped with Modbus instruction set. The setup involves simple driver initialization at end devices to communicate. Low development cost, hardware and compatibility with many OS makes it simple.
- **Standard Ethernet:** Ethernet ingrates easily into simple chipsets and boards. The cost of implementing Ethernet to Modbus is low and also provides ample resources as there are many developers are working on optimizing the technology. Ethernet port 502 is used by the Modbus TCP/IP protocol.
- **Open:** The Modbus protocol has been open source since 2004 and a dedicated organization working towards develpoment,optimization and maintenance.
- **Compatibility:** Modbus provides interoperability among various vendors and also compatibilty with devices of other manufacturers.

Modbus TCP/IP is an Internet protocol. This makes the devices open to the Internet. This was a particular feature that was incorporated to facilitate better control and making device maintenance through remote systems over the internet. Modbus is also industrial networks protocol and the industries are geographically separated. Modbus TCP/IP helps in better management of distributed industrial systems throughout the world.

3 Proposed System

In this work, a low interaction Mobile Honeypot mechanism to simulate an industrial PLC will be designed and implemented. The design also aims at detecting attacks and making inferences about the adversaries and attacks. The final implemented version will be integrated to the HosTaGe app along with advanced mechanisms that HosTaGe already provides to its users. As the proposed system deals with implementing a low interaction Honeypot, the challenge involves implementing only the essential components or services, that satisfy the discovery and vulnerability to attack them, for example, the network stack. Along with basic attack detection, the system must also have a short response time, robust design to withstand the attacks and also maintain a log of the exploit for further analysis and backtracking. An attempt will be made to detect attacks forged with popular identified worms like STUXNET. The conclusions on the attacks made will be pushed on to a central repository where the details of the attack are made public for users worldwide.

The proposed system also aims to identify Multistage attacks and generate signtures for IDS. The attacks containing malware are formally modeled to analyze their propagation through the network and their dropping mechanism. This model helps in identification and detection of such malware and also generating respective signatures.

There are several systems that propose the idea of Multistage attacks and their detection.

// Mention related work about Multistage here

The overlay of the proposed system, mechanisms and the evaluation are followed below.

4 System Design

In this section we discuss the design of the proposed HosTaGe ICS mobile Honeypot. The discussion involves the ICS perspective for HosTaGe, the simulated Siemens SIMATIC S7-200 PLC system , protocols supported by the PLC, the formal model, detection mechanisms and signature generation mechanisms.

4.1 HosTaGe ICS Perspective

HosTaGe has implemented mechanisms to emulate different kind of hosts like a windows host, linux host, webserver, FTP server, SSH server and more. The simulation of industrial level SCADA based PLC will be added to the the existing list of simulated hosts and services. To simulate PLCs it is important to understand their communication and control infrastructure. PLCs have network interfaces that support Ethernet, TCP/IP, Modbus[4], DeviceNet[7], ControlNet[7], Foundation Fieldbus[16]. The manufacturers have their own in built shells to support FTP commands. The Ethernet communication module of the PLC typically runs an embedded operating system that includes standard network protocol as well as implementations of industrial network protocols such as Modbus/TCP or EtherNet/IP. Telnet and FTP servers are common and have identifying information which can be used to determine the vendor and version of software. The network components that need to be simulated in a PLC are the TCP/IP stack, Modbus/TCP server, FTP server, Telnetd server and a HTTP web server which provides an interface to manage the functioning and control of PLC.

The discovery and identification of the PLC in the network can be through a network nmap scan that reveals information about the host name, ports 21, 80 and 502(Modbus) open.

The main objective is to detect attacks made using the protocols offered by the Siemens Simatic S7 200 PLC . A logging mechanism logs the information about the attacker in pursuit.

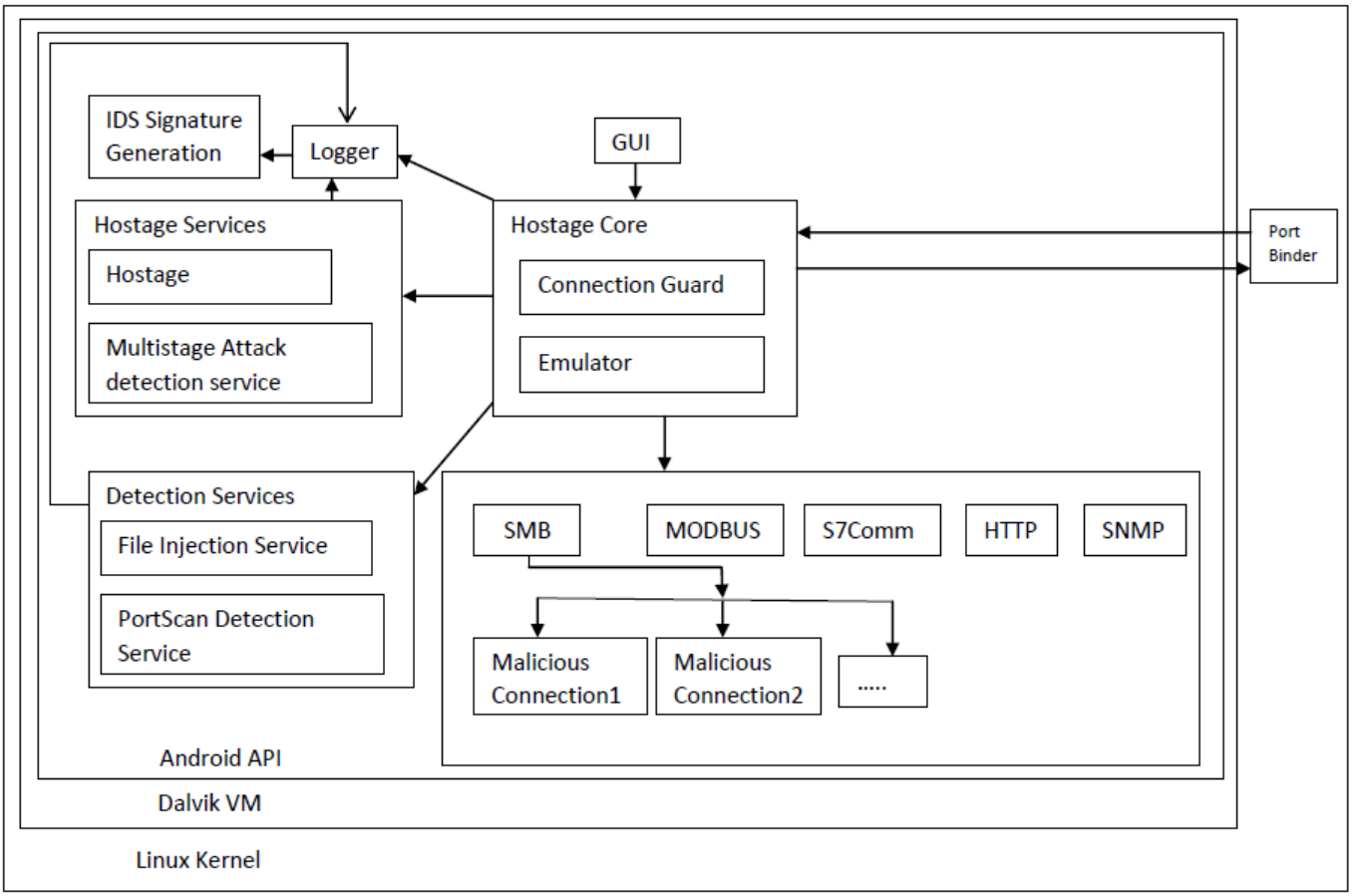

Abbildung 4: HosTaGe ICS Architecture

The architecture of HosTaGe ICs is as shown in Figure 8. A discussion of the components is followed in the below sections.

4.1.1 HosTaGe Core

The HosTaGe Core forms the basic core mechanism of HosTaGe ICS. It is responsible for running the core mechanism and functionality of HosTaGe. It provides an interface for the activation and deactivation of implemented emulated protocols. The core interacts directly with the GUI to provide notifications to the user of the connections made. The main components or sub modules of HosTaGe Core are Emulator and Connection Guard.

- **Emulator** The Emulator is responsible for the emulation of protocols in the Protocol Emulation . It is a multi threaded module which dedicates a thread for every protocol to be emulated and also actively listens to the incoming malicious traffic for the respective service ports. It calls the Logger module to log all the activities occuring at that port. The enulated protocol can accept multiple simultaneous connections at the same instance. A Connection Handler is started for every incoming connection request. Every Connection Handler communicates with the initiating client, providing a basic protocol interface based on the selected protocol for emulation. The Port Binder module is responsible for binding the ports .

  The emulator provides a selection interface for the user to choose the protocol to be emulated. For the ICS perspective we provide additional protocols like Modbus, S7, SNMP, SMTP and also a dynamically adapting version of SMB and HTTP which form a mainstream for attacking in ICS systems.Adding new protocols to the Emulator module is hassle free.

- **Connection Guard** A Honeypot has to be robust itself to detect malicious attacks and survive them. However, it is always possible to compromise a system through tailor made strategies. The connection Guard mechanism limits the number of connections that can be received by HosTaGe ICS. This measure makes sure that our system is safe under the Denial Of Service attacks. In such cases, the number of connections are limited by the source ip or for the destination port. The connections are also terminited over a period of time.

4.1.2 Logger

The Logger module is responsible for logging the attack and connection data into the SQLite database. It also supports export of the logs generated in different formats. The formats include plaintext and in JSON for data processing by third party applications.

4.1.3 Graphical User Interdace

HosTaGe ICS provides a user friendly and an interactive GUI for its users. The Overview shows the current state of the HosTaGe service with respect to the secure state of the network in which the device is placed. The Overview could be sleep, scanning- without any attack previously detected on the network, scanning- with an attack previously on the network and on attack detection. This provides an overview of the network health condition for the users. Figure** shows the GUI of the Overview screen of HosTaGe.

(Insert Fugure of GUI Overview)

HosTaGe offers various functional modes through the Profiles module. This module enables the user to select the Target system type that has to be emulated. Every profile has predefined protocols that have to be emulated which combine together to form the emulation of a target system. However, users can also select from a list of services that they want to include in the profile. HosTaGe also provides protcols to be emulated as per users choice. Users can decide upon the protocols to be emulated. HosTaGe ICS adds Modbus, S7, SNMP and SMTP to the list of protocols supported by HosTaGe. Figure** shows the the Profile view and the Services view of HosTaGe ICS.

4.1.4 Port Handling

The Android OS has a security policy of allowing only signed applications access to privileged network ports(below 1024). As HosTaGe is a third party application, by default it does not have access to these ports. The policy sounds fair with respect to the security enforcements but adds additional challenges in the implementation of the protocols for HosTaGe. ICS specific profiles include emulation of protocols like Modbus, S7, HTTP which fall in the network previlege ports category. A script was implemented to achieve this challenge in native C, cross compiled for Android OS. This program binds a port passed to it as a parameter and sends the file descriptor back to the caller through a UNIX domain socket. (cite This network is infected Honeypot). The program uses a Java server sockets to create a socket from a file descriptor

which ensures connection to the privilege ports. Some applications already bind to some previlege ports. In this case, HosTaGe notifies user that the port is bound to some other service.

However, a root privilege or a rooted Android device is required for the above mentioned program to work. We also look forward to overcome this situation for the app to work in the future.

4.1.5 HosTaGe Services

HosTaGe relies on services that run in background to achieve its constant attack detection activities. These services ensure monitoring the network, managing the connections, handling of attacks and notifications to the user through the app GUI. The HosTaGe app currently has 3 services running and are described as follows:

- **HosTage service:**The main service that HosTaGe relies on is the HosTaGe Service that controls and co ordinates the mainstream activities of HosTaGe. It is responsible for managing connections, listening on ports, enabling protocol services, attack notifications and logging of data. The HosTaGe service is resposible for the basic functionality of HosTaGe.
- **Multistage Attack detection Service**
- **Synchronization Service**

4.2 Siemens SIMATIC s7 200 - Overview

The Siemens S7 300 is a micro-programmable logic controller which can control a wide variety of devices to support various automation needs. The S7-200 monitors, inputs and changes outputs as controlled by the user program, which can include Boolean logic, counting, timing, complex math operations, and communications with other intelligent devices. It can control and communicate with devices like automatic pressure controllers, centrifuge pumps, water cooling systems. The STEP 7–Micro/WIN programming package provides a user-friendly environment to develop, edit, and monitor the logic needed to control the application that monitor devices. The Siemens Simatic S7 PLC's use PROFINET which is based on Ethernet for communication. There are over 3 million PROFINET devices deployed worldwide.

Siemens S7 200 PLCs boasts of a compact design, powerful performance, optimum modularity and open communications. This Micro PLC has been in successful use in millions of applications around the world,in both standalone and networked solutions.

This PLC uses communication protocols such as PROFINET, an advanced version of Modbus communication protocol. This protocol is also based on Ethernet. It also supports TELNET, HTTP, FTP, SNMP, SMTP, Modbus and S7 Comm protocols. Though this PLC is designed to be used to control critical systems, security was not a part of its design. The above mentioned protocols were not customized to facilitate secure communication. The standards were defined to create an interconnected environment between industrial automation devices and common networking protocols.Security was either ignored or rather was thought to be expensive on these devices. This makes it an easier target for attackers.

The Simatic S7 PLC is also subjected to various vulnerabilities and attacks including the STUXNET as discussed earlier. We simulate the Siemens SIMATIC S7 200 PLC as our target system to attract attackers.

4.3 Protocols

The ICS SCADA systems include the master and slave devices. Our design must be capable of simulating the services of both the master and slave devices. The Siemens SIMATIC S7 supports a wide range of protocols which include Modbus/PROFIBUS TCP, HTTP, TELNET, FTP, SNMP, SMTP and S7Comm. Modbus TCP and S7Comm are the communication protocols and the rest of the protocols are enabled as added features. Considering the other protocols form an important part of the security analysis as malware are usually designed keeping the flaws of the network and the software bugs in mind.

- **HTTP:** HTTP is supported by the majority of PLCs for remote configuration purposes. The HTTP web server in the PLC enables GET/POST messages for information exchange. This HTTP server is simulated by HosTaGe through a dynamic HTTP protocol implementation. A default welcome page is displayed when the adversary tries to navigate to the device's webpage.

- **Telnet:** The Telnet protocol allows accessing a basic shell on the devices in which users are able to dump memory, delete files and execute commands. It provides command and control to the target remote devices. It enables file system based commands and directory listing. Users or applications can communicate with the PLC for file and backup operations.
- **FTP:** FTP provides file transfer and communication between end devices. These are usually files containing sensor readings and logs.
- **SNMP:** The Siemens S7 family of PLCs the configuration of client devices through SNMP. This allows to remotely manage devices on the network.
- **SMTP:** SMTP is mainly enabled for notification servivce in case of device failure or data inconsistency.
- **SMB:** Although SMB is not a part of Siemens S7, it is a main component of the Master devices that control the slave PLCs. The SMB protocol is used to share files in a network of hosts. This protocol is typical on enterprise intra-networks due to file sharing requirements.
- **Modbus/PROFIBUS TCP:** Modbus TCP acts as a strong communication mechanism between the slaves and the master devices. It forms a backbone for industrial systems automation. Modbus has instruction sets for the interaction of devices. PLCs have registers 1 as memory units. The instruction sets are specified as functions which denote Read/Write (R/W) operations on the registers of the PLCs The protocol is used for communication exchange between PLCs and control systems.
- **S7:** The S7 protocol (S7 Communication) is a Siemens proprietary protocol utilized in PLCs of the Siemens S7 family. It is used for PLC programming, exchanging data between PLCs, accessing PLC data from SCADA systems and for diagnostic purposes. The protocol forms as a base for accessing the registers for R/W operations and also programming the PLC for user defined tasks.

4.4 Formal Model

The detection mechanism of HosTaGe ICS is formalized with an Extended State Machine (EFSM). An EFSM has all the properties of a normal Finite State Machine (FSM) with the added feature of utilizing *if* -conditions, instead of only boolean conditions, to specify how a state transitions to a new state [**?**]cite from bib Marcelo Fantinato). The formal model of our proposed detection mechanism is given by Attack Detection EFSM $M = (S, s_0, I, O, V, P, \delta, \lambda)$

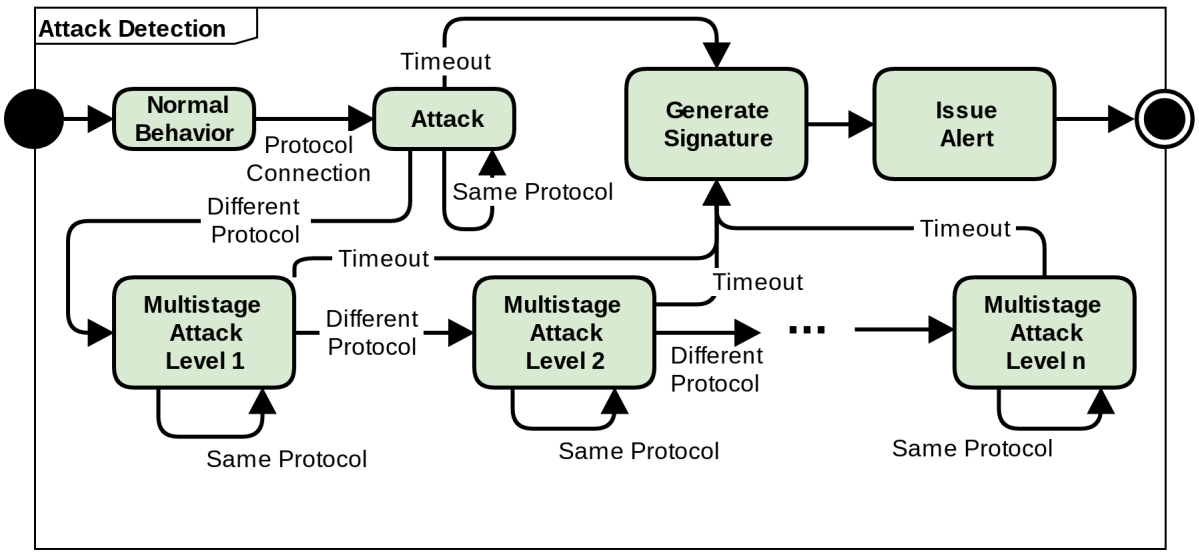

Abbildung 5: EFSM of the attack detection and signature generation mechanism [2]

and is illustrated in Figure 4. The set of all states are represented with S. The EFSM starts in the *Normal Behavior* state, represented by s 0 . If any protocol communication is detected by the Honeypot, the EFSM transitions to the *Attack*

state. For as long as the same protocol attack is observed, the state remains the same. If a timeout occurs the EFSM transitions to the *Generate Signature* state followed by the *Issue Alert* state. The signature generation is optional and will capture either single attack or multistage attack types. After an initial attack, observing attacks originating from other protocols (but the same host) that have not yet been observed moves the state to the next *Multistage Attack Level x*, where *x* corresponds to the number of different protocols observed after the first one.

The inputs I, outputs O, variables V and predicates P are tightly linked together. State transitions are carried out wherever specific inputs $i \in I$ are received.This transitions may also generate an output $o \in O$. In the *Normal Behavior, Attack* and *Multistage Attack Level x* states, the supported protocols are used as inputs and outputs. As such, $\{Modbus, S7, SNMP, HTTP, Telnet, SMB, SMTP, HTTPS, SSH, FTP\} \in I \in O$ for these states. The inputs of I are not limited, however, to only ports. Special activities of interest on a protocol are also considered inputs. For instance, the act of requesting a file through the *SMB* protocol (cf. Section 4.3) is an input on itself. V is a finite set of variables. These variables are used to construct a set of predicates P used for determining if a state transitions to another one. Each attack state hold a boolean variable $v \in V$ for each emulated port. If a particular port has been observed in the entire life of the EFSM, the corresponding variable for that port will be set to true. Besides variables, predicates P consist of the logic operator *AND* and the arithmetical operator $=$.We define the *Protocol Connection* predicate as the condition where a new protocol is observed without having observed other protocols yet. The *Different Protocol* predicate indicates, as the name suggests, that a new protocol has been observed after having seen at least one other. If any of these predicates is true a state transition takes place.

The final element of our model is the set of transitions $\delta(s_i, i, p) = s_j$ and the outputs $\lambda(s_i, i, p) = o$ generated by the transition itself. The set of transitions specify that whenever state $s_i \in S$ receives the input i and the predicate $p \in P$ is satisfied, the EFSM transitions to state s_j and outputs $o \in O$. The outputs are used by the *Generate Signature* state to create signatures for misuse analysis.

4.5 Detection Mechanisms

HosTaGe ICS is designed to identify three different class of attacks: Single-Protocol Level Detection (SPLD), Multi-Stage Level Detection (MSLD) and Payload Level Detection (PLD).

- **SPLD:** SPLD attacks refer to those that occur on a single-protocol, eg:HTTP connection attempts without observing other protocols or any extraordinary payload-level information. This is the simplest type of detection which still contains interesting analysis potential.
- **MSLD:** MSLD refers to attacks that originate from the same source and attempt to exploit different types of protocols within a small window of time. These type of attacks are identified by the Honeypot with the EFSM shown in Figure 4. An important factor in MSLD is the time-window (tw) that determines whether an attack should be mapped as the SPLD or the MSLD class. This means that when the EFSM is on the Attack state and no further activity is detected (for a maximum of tw) a timeout will occur and the attack will be identified as SPLD. The tw can be adjusted with respect to the monitored network and its requirements.
- **PLD:** Payload Level Detection is enabled for the HTTP, SMB and Modbus protocols. These protocols carry critical payload in the case of ICS environment. There could be malware in the form of executables injected in the payload which can trick the end systems to execute or process them. Studies show that majority of the malware existing today spread through HTTP payloads. PLD extends the applicability of the EFSM with respect to the inputs I. Referring back to our formal model, the outputs $o \in O$ from the Attack and Multistage Attack Level x states are used in the Generate Signature state to create signatures. Signatures are also EFSMs that comply with the presented model. As we already mentioned, the input is not limited only to a port or protocol but also to potentially interesting payload- level information. Figure 6 can be considered as an example of an EFSM that represents a signatures generated by PLD. This signature identifies Stuxnet attacks from the set of outputs O obtained from the Attack Detection EFSM shown in Figure 5. The Detection of Stuxnet EFSM assumes an initial Normal Behavior state and transitions to SMB Attack if an SMB protocol is observed. Stuxnet tries to inject an infected file through SMB. After a file is received, it (or its hash value) is sent to VirusTotal and, if the file is indeed malicious, the EFSM transitions to the Stuxnet Attack state where its presence can be reported.

4.6 Signature Generation

An Intrusion Detection System relies on its signature set and static response analysis in order to determine malicious packets and traffic. IDS depending on the signature set are called signature based IDs and the one's relying on heuristics

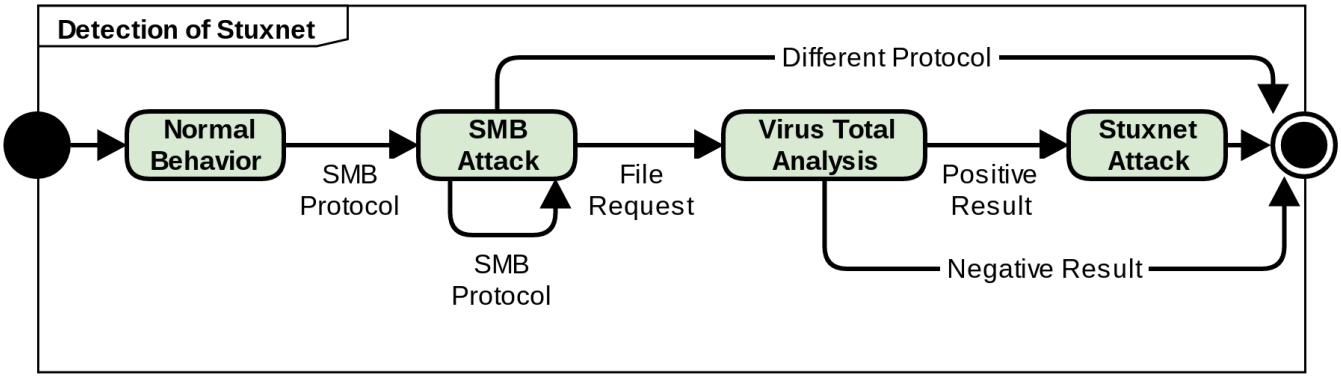

Abbildung 6: EFSM for PLD in the case of Stuxnet propagation [2]

are called anomaly based IDS. Signature based IDS monitors the packets and compare them with a database of signatures. An example for signature based systems could be antivirus systems that detect malware by comparing the files with a known set of malware signatures. Anomaly based IDS detect intrusions by monitoring the system activity and classifying it as normal or anomalous. This classification is based on rules or heuristics to determine anything which is different from normal system behavior.This type of IDS also relies on neural networks and artificial intelligence algorithms to classify normal traffic.

Majority of Enterprise Networks include IDS on their network and are dependent on them for identifying and detecting malicious attacks and traffic. It is also true that due lack of expertise or internal constraints, many Enterprise Networks do not prefer having Honeypots as decoys to determine malicious attackers and traffic. One such feature why IDS are preferred is because of commercial support and maintenance. IDS developers constantly monitor for newer malware and develop signatures to keep the IDS signature database updated. The administrators have to just update the IDS signature database to defend against newer malware. This update system would fail under some circumstances where the developers fail to recognise the malware or if its a tailored attack strategy against an organization.Thus we propose a model of HosTaGe ICS where our honeypot has the feature of generating signatures for an IDS which could be deployed on a active IDS that monitors the network.

The Signature Generation model of HosTaGe ICS concentrates on utilizing the attack results logged by HosTaGe. The active attack correlation techniques determine newer attack strategies followed by attacker, help in understanding the payload of malicious traffic and also instantly generate signatures that can be deployed to an IDS. This model also binds the usage of both an IDS and a Honeypot in a network for gaining stronger security aspects by utilizing the advantages and features of both an IDS and Honeypot.

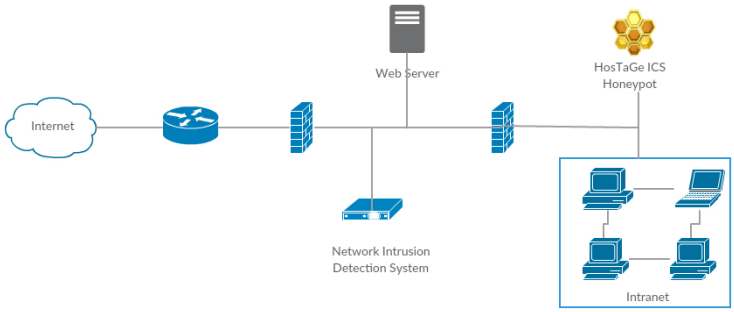

Abbildung 7: Network Architecture of IDS and HosTaGe

Figure 7 depicts the suitable network architecture for having both an IDS and HosTaGe. The firewall acts as a first line of defense allowing only required traffic. The Network Intrusion Detection comes second monitoring every packet for any malicious entries. HosTaGe can be placed either at the internal network or at the DMZ depending on the required area of interest for monitoring. It functions well at both areas. Any attacks towards HosTaGe is logged. This logged data can be analysed further and if found malicious, a signature file for matching the content of the payload can be generated. This signature can be mounted on the Intrusion Detection System for monitoring the packets along with other attack signatures.

4.7 SCADA PLC Profiles

SCADA ICS devices can be classified into master and slave device types based on the interaction and functionality. The master system is responsible for controlling the slaves and send them appropriate commands for a task. These systems are usually control servers or host systems connected to PLCs or slaves, that receive critical information and updates from the sensors placed on devices and PLcs. The other most important systems are the automation PLCs. Slave devices interact with many other devices and collectively process information to perform a task assigned by the master. When a Modbus master wants information from a device, it sends a message that contains the device address, the data it needs and the checksum for integrity. The network is typically like a hub structure. The data is broadcast in the network and the device from which the information was requested only responds. The slave devices cannot initiate communication and only can respond to a request made from the master.

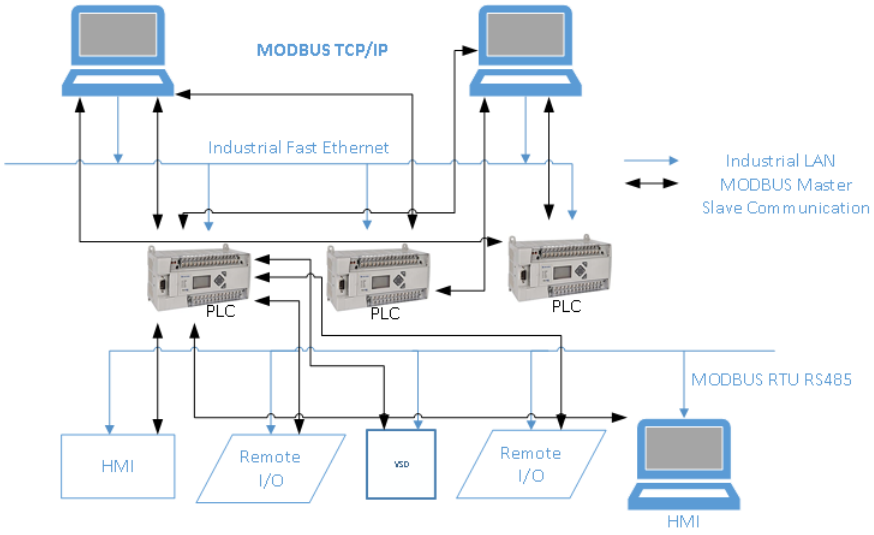

Abbildung 8: SCADA Master and Slave profile

Modbus/TCP allows multiple masters to poll the same device in parallel. A unit can be either a master or a slave but not both.

The Figure 8 represents devices connected on the industrial LAN and the Modbus master-slave communication. The master devices poll the slave devices and request information. The information is processed and sent back to the master. There is also possibility that a PLC acting as a master polls its data to the other devices like HMI and other PLC's in the network.

In the past there have been attacks both internal and external on SCADA systems. Popular attacks using STUXNET, were carried out internally by deloying the malware on a host computer with the help of a USB drive. However, the malware made use of the vulnerabilities of the host system to replicate and spread through the network. Detecting such kind of attacks are very important and cannot be ignored. These attacks are more dangerous than the external attacks as there are various mechanisms to detect attacks from external sites. Internal attacks have proved to be more catastrophic. We also concentrate on the slave profile. This is required as the slave devices today have Ethernet communication and can communicate with the Internet. Due to some network configuration loop holes, the device may be accessed due to the internet or the device itself may be configured to be accessed through the internet by the administrator. For example, the slave devices also run HTTP servers which can display the sensor information in the form of a webpage. This device may be configured to be accessed through the internet to check and monitor the sensor readings from a external system. There is no doubt about the possibilities of attack of such systems from the internet. Thus we concentrate on simulating both the master profile, to check internal attacks and also slave profile to check external attacks.

5 Implementation

5.1 Exploit Areas

We discussed the architecture, features and protocols facilitated by the Siemens SIMATIC S7 200 PLC and also security concerns of ICS SCADA systems. There were many exploit areas that were discovered. The PLC was subjected to various exploits and attacks. However, large scale attacks like STUXNET were successful because of vulnerabililtes that existed on the Host controllers as well, that is, Windows OS hosts. It made used of zero day exploits from both Windows OS and the Siemens PLCs. The attack was well designed and strategised considering vulnerabililtes on both systems. There are also small attacks like information leakage from an internet facing PLC, hosting a webserver. Over the years many vulnerabilities have been identified on the Siemens PLCs. It becomes a great challenge to make these systems secure. The PLCs have limited resources and thereby security measures like data encryption may prove expensive. Hence data encryption was avoided. This decision of ignoring secure features induced several exploits for the device.

The Honeypot must be designed keeping all the discovered exploits inorder to be more effective in attracting the attackers. We consider both the external and internal attack approaches hereby devising strategies to capture both kind of attacks. Before we design our Honeypot, it is very important to understand the previous known attacks on PLCs, their impact and the vulnerabilities that caused those attacks.

5.2 Implementation of HosTaGe ICS Honeypot

Based on the design decisions made in the Section 4, HosTaGe is implemented to simulate the services of the Siemens Simatic S7 200 PLC. The services include the simulation of HTTP, Modbus, S7, SMTP, Telnet, FTP, SNMP protocols with respect to the PLC. HosTaGe also supports creation of Honeypot environment profiles. We leverage this feature to create profiles of Nuclear Power Plant that includes a Modbus slave open to the internet with the other protocols. The services offered by th other protocols are also customized based on the profile chosen by the user. The services implementation involve handling incoming connection requests on respective ports of the protocols. Recollecting the main objective of a honeypot is to keep the attacker engaged and allow him to pursue his attack strategy, a proper response mechanism is required which provides satisfying responses to the attacker commands and packets. For example, if the attacker tries to access the webpage by sending a GET request to port 80 of the Honeypot, a webpage must be displayed to the attacker. Further, the webpage must also be dynamic depending on the honeypot profile chosen on HosTaGe. This feature has been implemented to keep the attacker engaged to HosTaGe while all the packets are captured and stored as records in HosTaGe attack database.

We also implement the detection of malware propagation through the SMB protocol. This detection forms an important aspect of detecting popular malware like STUXNET through the network. This forms as an extension to the implementation of the SMB protocol for the Modbus Master profile. As discussed in the exploit areas, the most dangerous incident reported with respect to ICS SCADA systems in STUXNET which propagated in the intranet of the plant. We try to replicate the scenario to facilitate STUXNET propagation by simulating HosTaGe as a shared network drive. We discuss this module further in the section //write number.

During the course of thesis, an attack pattern was recognized and observed amongst the attacks recorded by HosTaGe. The attacks were later analysed to be Multistage attacks. Multistage attacks are attacks that originate from the same source ip and attempt to attack multiple active protocols on the target system within a time window. This helps in reducing false positives and acts as a means to detect genuine attacks. The detection of Multistage attacks labelled as MSLD in section 4.5 has been implemented based on the formal model shown on Figure 4. The approach to detect MSLD is discussed further in section //mention section.

The signature generation module is one of the important component. It focuses on creation of signatures for an IDS based on the attack data captured and stored on HosTaGe. Signatures are generated and exported as files which can be mounted on an IDS for signature based traffic analysis. The creation of policies or rules is also implemented for the detection of MSLD. The implementation is further discusses in section //provide number.

5.3 Detecting Internal Attacks

As discussed previously, ICS SCADA systems have master and slave profiles. Though the devices are subjected to attacks from external attacks, when made open to the internet, it is proved that major attacks in the past were triggered by systems in the internal network. Attacks from malware such as STUXNET spread from host systems in the same network. Attacks from internal systems have proved to be more effective and dangerous as they do not leave any fingerprints, also their signature cannot be identified by the anti-virus softwares and other protection tools. The STUXNET worm was

reported to be injected through a USB flash drive. It made use of zero day vulnerabilities of the Windows operating system, the most popular one being how the Windows operating system handles the LNK files, which are used by the operating system to interpret devices capable of AUTORUN functionality, and to detect the software to run the file based on its format.

An anatomy of similar kind of viruses and malware revealed that they made use of as many zero day vulnerabilities as possible to make the malware attack more effective and stealthy. Identifying such malware attacks through our Honeypot mechanism is a challenge, as it involves careful design and simulation of services involved in such attacks. To achieve this, the conditions under which such worms propagate and try to sneak into the network is studied. Analysis of the studies made by researchers (cite STUXNET: Dissecting a Cyberwarfare weapon) shows that the worm looks for (complete the rest by referring the paper)

As discussed above STUXNET exploits the zero day vulnerabilities on a Windows host and is dormant without it. Hence it is required to simulate atleast one of the zero day vulnerabilites. The best suited amogst the five was the propagation through the network shared drive. This service could be simulated like on a WebDav server. We could then wait for the virus to propagate itself into this simulated location.

5.4 Detecting malware

5.5 Detection of Multistage Attack Approach

5.6 Signature Generation

As discussed previously in Section ** Signature Generation module generates signatures for the attack traffic that is logged. We consider the Bro Network Security Monitor for generating signatures. The signature generation module is a combined model involving modules of HosTaGe. Additional modules have been implemented to support signature generation for the Bro IDS. We highly make use of Bro's features to incorporate the signatures generated through HosTaGe. A brief discussion about the Bro IDS and the signature generation is provided in the below sections.

5.6.1 Bro Network Security Monitor

Bro is a powerful network monitor that provides a strong analysis framework. It provides a good platform for traffic analysis, signature matching and forensics. Bro is opensource and Unix based. It is originally written by Vern Paxson, and now maintained by researchers at International Computer Science Institute, Berkeley, CA and worldwide. Bro is implemented using the Bro scripting language. it is often associated to being a framework and could be used to build effective IDS. Bro is opensource and provides lot of documentation with researchers involved from all over the world. Bro can analyse traffic both live and offline to perform analysis, take network measurement, forensic investigation and traffic baselining.

Bro's has many features that aim at building better IDS. The features are discussed below:

- **High-Speed and large volume monitoring:** Bro is capable of handling huge amounts of traffic. It is also quick in analysing the traffic for the policies and signatures. Bro can also be deployed to work as a distributed security monitor, where it can share states and also propagate event data to notify other distributed peers.
- **No Packet Filter Drops:** As discussed in the previous point, Bro is capable of handling huge traffic flows. There is very minimal packet drops observed. If an application using a packet filter cannot consume packets as quickly as they arrive on the monitored link, then the filter will buffer the packets for later consumption. However, eventually the filter will run out of buffer, at which point it drops any further packets that arrive. From a security monitoring perspective, drops can completely defeat the monitoring, since the missing packets might contain exactly the interesting traffic that identifies a network intruder. Thus, Bro makes sure that the packet drops are to a minimum making it possible to capture traffic without any missing links.
- **Real Time notification:** Detecting attacks online as they happen is very much useful in taking relevant steps for mitigation and notification in real time. Offline detection leads to delay in analysing the attack and being proactive. Bro supports providing real time notifications of attacks. An attack detected live provides better trace back capabilities and to minimize damage.
- **Independent Policies:** Bro separates the working mechanism from its policies. This offers more flexibilities in customizing the policies and not the mechanism. This also leads to simplicity in analyzing the policies and rewriting them. Policies could also be referred to as rules as in a IDS representation.

- **Language Support:** The Bro language is designed keeping network communications as a base. It offers datatypes for network basics like ip address, subnet, port, source ip, destination ip, source port and destination port. It also supports conditional and non-conditional programming keywords like in , when and match which are very beneficial.

- **Scalable and Extensible:** As Bro can be deployed as a distributed environment, it can easily scale to monitor huge networks and share state information. Bro has co ordinators, peers and master systems to manage the Bro deployments. Bro is opensource and this enables to extend the platform for independent researchers. This is also provides the freedom to develop and test new policies and modules.

- **Protocol Support:** Bro processes both TCP and UDP packets. For each TCP packet, the connection handler verifies that the entire TCP header is present and validates the TCP checksum over the packet header and payload.

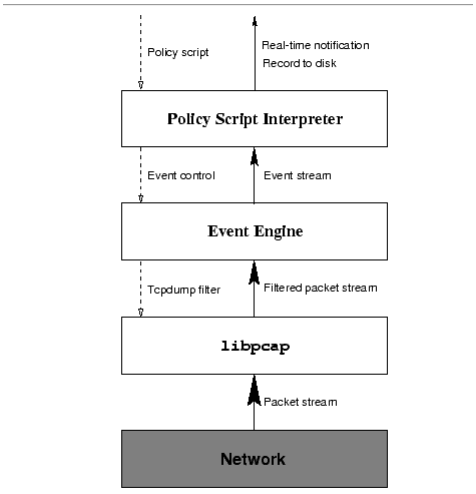

Figure 1: Structure of the Bro system

Abbildung 9: Bro Architecture.
[11]

Bro's architecture involves Bro Event Engine, Policy Script Interpreter and libcap. Figure depicts the architecture of Bro and the analysis framework. The role of each component is described below:

- **libcap:** libcap is the packet capture library used by Bro. Libcap isolates Bro from details of network link technology. It also enables to Bro to work offline by accepting packets to analyse threats and malicious content.

- **Event Engine:** This layer performs integrity checks to assure that packet headers are well-formed and also verifies the IP header checksum. Failure of checks generate events indicating the issue resulting in dropping of packets. If the checks succeed, then the event engine looks up the connection state associated with the tuple of the two IP addresses and the two TCP or UDP port numbers, creating new state if none already exists. It then dispatches the packet to a handler for the corresponding connection (described shortly). Bro maintains a tcpdump trace file associated with the traffic it sees. The connection handler indicates upon return whether the engine should record the entire packet to the trace file, just its header, or nothing at all.

- **Policy Script Interpreter:** The event engine checks if any events have been raised. If any events are generated it processes the events as per the event handler specified until last event. Bro's emphasis on asynchronous events as the link between the event engine and the policy script interpreter provides lot of extensibility. Adding new functionality to Bro generally consists of adding a new protocol analyzer to the event engine and then writing new event handlers for the events generated by the analyzer. This holds true also for the policies and the signature. We leverage this feature in HosTaGe to generate policies and signatures for Bro.

5.6.2 HosTaGe Records

HosTaGe stores all the packet data it receives in the form of records. The records include source ip, source port, attack type, protocol and the packet conversation. The data stored as conversation is the payload of the connection packets. This packet data can be considered for pattern matching and signature generation. This data is passed as a parameter for signature generation module. The records are a part of the Attack Logs module which is further discussed in Section 5.7.

5.6.3 Signature Generator

The Signature generator contains pre-defined templates which are implemented for specific protocols like Modbus and SMB for File Injection. These templates accept parameters from the conversation information stored. It also accepts the source ip address as a parameter. The source ip address is included to avoid false positives. As discussed previously in Bro architecture, there are also policies that Bro uses to check the incoming traffic. The signature generator module also generates policies for the Multistage attacks detected by HosTaGe. In the case of Multistage attacks, the conversation information is not considered for the signature generation. The source ip address and the protocols attacked are considered for the generation of policies for Multistage attacks.

5.7 Attacks Log

Data analysis of the attacks received also form an important part of a honeypot. This feature enables administrators to carry out forensics on the attack data and check for new malware types. Through attack analysis the attack vectors can be mapped which help us understand the strategies followed by attackers to compromise systems. The Attacks Log has an entry of all the incoming and outgoing connection information. Each connection received to HosTaGe is further divided as attack record, network record and the message record.

5.8 Challenges

6 Evaluation and Results

The evaluation was performed on important results derived which include evaluation of attack data with respect to individual protocols implemented for ICS SCADA profiles, evaluation of Multistage attacks and signature generation. Evaluating the performance of HosTaGe with respect to battery consumption and also comparing this performance with apps which utilize similar resources on the device is performed. In addition, we discuss the observed insights regarding the detectability of HosTaGe ICS.

6.1 Analysis of Individual Protocol Attacks

6.2 Conpot and HosTaGe attack comparison

In the Section 2.4.4 Conpot is described briefly. Conpot is an ICS Honeypot which simulates the behaviour of a Siemens Simatic S7 200 system. It has the S7, Modbus , HTTP and SNMP protocols implemented to support the simulation of the PLC. Conpot was deployed on one of our servers and was exposed to the internet for attacks. All the connection packets received for HTTP, SNMP, S7 and Modbus are logged. However, the complete packet with the payload information is not logged by Conpot. The logging mechanism is different for each protocol. For evaluation, HosTaGe ICS was also deployed and exposed to the internet for attacks. Both Honeypots were kept running for a specific evaluation period to analyse and compare the results at the end of the evaluation period. The honeypots were deployed in controlled environments with no firewalls between them and the Internet in the same /24 subnet.

The results of the analysis are shown in Figure. The results gathered from the two honeypots for the HTTP, Modbus and S7 protocols are compared. The Telnet protocol is a part of HosTaGe ICS simulation. In addition to the mentioned protocols, HosTaGe also received a lot of Telnet attacks. Telnet is considered important as it provides a shell access on the PLC devices, through which command and control is possible directly. We also show the Telnet attacks in Figure.

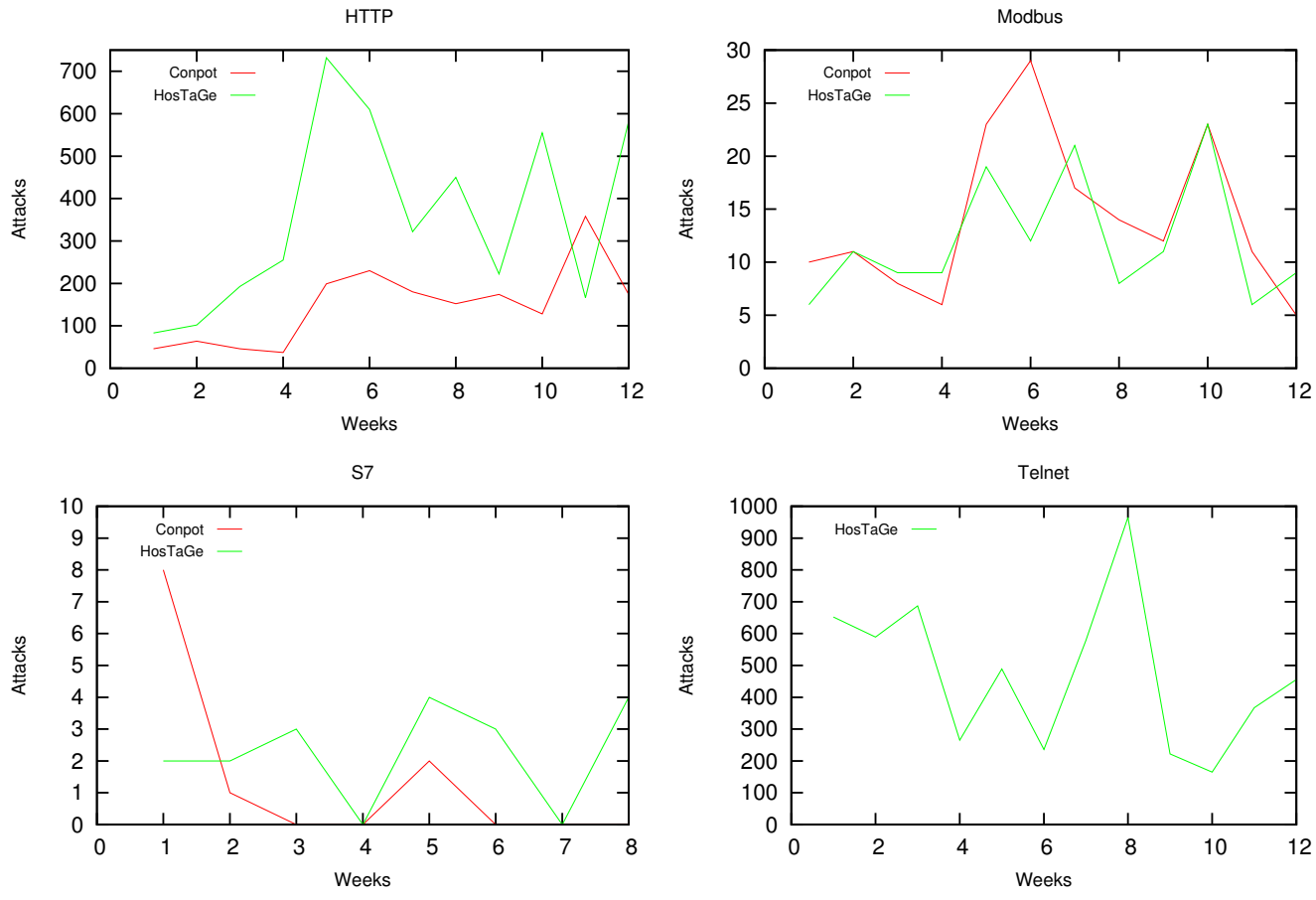

Abbildung 10: HosTaGe and Bro Comparison

Inferring some results from the Figure we can note that HosTaGe exhibits good and on-par detection accuracy when compared to Conpot. In the case of HTTP, HosTaGe is observed to have more attacks than Conpot and remains on par with Conpot on Modbus. It is important to mention that neither of the Honeypots were advertised in any form on the Internet. However, Shodan the online vulnerable device database and search engine managed to find and probe both Honeypots. This is further discussed in Section. The main goal of this evaluation was to compare the performance of HosTaGe with Conpot in terms of attack detection and gathering. It is notable to observe that Conpot runs on a full PC environment while HosTaGe functons equally better on a mobile environment. This inference forms a basis for proving that HosTaGe is a powerful Honeypot which runs efficiently on limited resources and is capable of simulating the services of real devices on a mobile device platform. It also encourages for addins and plugins for future enhancements.

6.3 Stuxnet Propagation

6.4 Multistage attack detection and Inference

6.5 Bro Signature Generation and Evaluation

6.6 Shodan Evasion

One of the most important and essential features of a Honeypot is to remain undetected as a decoy mechanism. This is a very hard feature to achieve and to be evaluated. During our experimental evaluation setup of Conpot and HosTaGe ICS, we received a lot of probes from Shodan, a popular online vulnerability device database and scanner. Shodan is one of the biggest search engines to find vulnerable devices on the Internet.It crawls the entire internet to check for vulnerable devices. On the Shodan website, users can search for ip addresses, specific devices, protcols and also vulnerabililtes like heartbleed or default password. Recently, Shodan started a new service called Honeypot Or Not? which can identify Honeypots. This service performs a series of probes or checks and subsequently creates a score, which is called Honeyscore for each probed device. Shodan determines if the host is a honeypot based on this score. During the evaluation period both honeypots received a lot of probes from Shodan. The probes were targeting HTTP, Modbus and S7 protocols. There were as many as 90 probes received over a period of 8 weeks. The probes were observed to be from 7 different subnets.

After the time period , Shodan identified Conpot to be a Honeypot and listed this on its website at the Honeypot or not site. It also showed the open services and protocols. HosTage ICS was not detected as a Honeypot by Shodan. The Shodan probes for HTTP and SSH were of low complexity, where the HTTP protocol involved a GET request and the SSH protocol was just the initial handshake message. The probes for Modnus and S7 protocols looked more sophisticated. An analysis made to the payload reveal that it could be a modified Nmap or Metasploit script to identify ICSs. The S7 attack involved checks for the device type, location, serial number, plant identification and module name. The Modbus attack involved fetching details of certain units (i.e., unit number 0 and 255) and their slave data. On the one hand, Conpot could not respond meaningfully in all the aforementioned requests (either due to static serial numbers or Modbus protocol simulation errors) and thus was classified as a honeypot. On the other hand, HosTaGe managed to respond successfully and hence remain undetected.

//put figure here of number of shodan probes

6.7 HosTaGe ICS - Performance Evaluation as an Android App

6.8 Results

[19]

Abbildungsverzeichnis

Literatur

[1] DELL. Dell security annual threat report 2015. 2015.

[2] Carlos Garcia Cordero Max Mühlhäuser Emmanouil Vasilomanolakis, Shreyas Srinivasa. Multi-stage attack detection and signature generation with ics honeypots. *NIST special publication*, pages 800–82, 2011.

[3] A. Galante, A. Kokos, and S. Zanero. Bluebat: Towards practical bluetooth honeypots. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–6, June 2009.

[4] Dao gang Peng, Hao Zhang, Li Yang, and Hui Li. Design and realization of modbus protocol based on embedded linux system. In *Embedded Software and Systems Symposia, 2008. ICESS Symposia '08. International Conference on*, pages 275–280, July 2008.

[5] Thorsten Holz and N Provos. Virtual honeypots: from botnet tracking to intrusion detection, 2008.

[6] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. pages 49–51, May 2011.

[7] Feng-Li Lian, James R. Moyne, and D.M. Tilbury. Performance evaluation of control networks: Ethernet, controlnet, and devicenet. *Control Systems, IEEE*, 21(1):66–83, Feb 2001.

[8] Steffen Liebergeld, Matthias Lange, and Ravishankar Borgaonkar. Cellpot: A concept for next generation cellular network honeypots. 2014.

[9] Steffen Liebergeld, Matthias Lange, and Collin Mulliner. Nomadic honeypots: A novel concept for smartphone honeypots.

[10] Collin Mulliner, Steffen Liebergeld, and Matthias Lange. Poster: Honeydroid-creating a smartphone honeypot. *IEEE Symposium on Security and Privacy (S&P)*, 2011.

[11] Vern Paxson. Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[12] Niels Provos. A virtual honeypot framework. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 1–1, Berkeley, CA, USA, 2004. USENIX Association.

[13] S. Rubin, S. Jha, and B.P. Miller. Automatic generation and analysis of nids attacks. In *Computer Security Applications Conference, 2004. 20th Annual*, pages 28–38, Dec 2004.

[14] Lance Spitzner. Honeypots: Catching the insider threat. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 170–179. IEEE, 2003.

[15] Keith Stouffer, Joe Falco, and Karen Scarfone. Guide to industrial control systems (ics) security. *NIST special publication*, pages 800–82, 2011.

[16] J.-P. Thomesse. Fieldbus technology in industrial automation. *Proceedings of the IEEE*, 93(6):1073–1101, June 2005.

[17] Emmanouil Vasilomanolakis, Shankar Karuppayah, Mathias Fischer, Max Mühlhäuser, Mihai Plasoianu, Lars Pandikow, and Wulf Pfeiffer. This network is infected: Hostage - a low-interaction honeypot for mobile devices. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pages 43–48. ACM, 2013.

[18] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, and Mathias Fischer. Hostage: A mobile honeypot for collaborative defense. In *Proceedings of the 7th International Conference on Security of Information and Networks*, pages 330:330–330:333. ACM, 2014.

[19] Emmanouil Vasilomanolakis, Shreyas Srinivasa, and Max Mühlhäuser. Did you really hack a nuclear power plant? an industrial control mobile honeypot. In *IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2015. to appear.

[20] Matthias Wählisch, Sebastian Trapp, Christian Keil, Jochen Schönfelder, Jochen Schiller, et al. First insights from a mobile honeypot. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 305–306. ACM, 2012.

[21] John W. Webb and Ronald A. Reis. *Programmable Logic Controllers: Principles and Applications*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition, 1998.