The Hunt for Major League IoT and ICS Threats
A Deep Dive into IoT Threat Terrain

November 2020

TXOne Networks Global Threat Research Group

Mars Cheng
Patrick Kuo

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Because the Internet of Things (IoT) plays a major role in modern society and business, IoT and ICS threats are consistently in development. Security incidents and threat research have both shown that more than half of internet- active IoT devices have been impacted by attackers' malicious actions.

We developed and deployed several automated threat hunting engines worldwide with the long-term intention of improving detection of and defense against IoT and ICS threats. From September 2019 to October 2020, we received about 20 TB of traffic through our system. Within that traffic, we detected 1.2 billion attacks (originating from 200 countries), classified 70 million distinct suspicious IPs, identified 2 million distinct malicious domains from 15 million suspicious domains, and collected over 2.63 million malicious files including RATs, trojans, worms and ransomware. Among these malicious files, more than 33% were unknown at the time of reception – meaning, VirusTotal does not have a listing for them. We also found that more than 1.49 million devices may have been assimilated into botnets.

In this paper, we will show how we built our large-scale automated threat hunting system, and use 6 hunting examples we analyzed in the past year to give an overview of current trends in threat development.

- Published by TXOne Research
- Written by **Mars Cheng** and **Patrick Kuo** from TXOne's Global Threat Research Group
- With contributions from **TXOne Threat Research**, **TXOne Signature Research** and **Trend Micro Inc.**
- **Acknowledgements**: The authors would like to thank Michael Cheng and his team for their contributions to the previous work of hunting systems. The team members listed in alphabetical order are Babylon Tien, Chizuru Toyama, Eric M Kao, Fisher Wu, James Chang, Joe Chang, Linwei Tsao, Mesh Wu, Samuel Chen, and William K Chang. We would also like to thank Marco Balduzzi and Numaan Huq from Trend Micro Research for conducting peer review of our work.

INTRODUCTION TO AUTOMATED THREAT HUNTING

Internet of Things (IoT) technology is indispensable in today's society. It assists people in their daily lives and adds tremendous convenience. Applications in society and business can be as small as routers, webcams, printers, smart lights, door locks, smart refrigerators, and medical equipment, or as large as smart cities, smart grids, smart ports, industrial manufacturing, and other critical infrastructure systems. However, a large number of vulnerabilities in vital IoT devices have already been exploited, and defenders are constantly on the move to discover and provide patches. Massive known and unknown global IoT attacks wait for no one, so every opportunity to patch holes in an ICS system's defenses must be maximized.

Manual operation-based threat hunting is less able to effectively detect and defend against large-scale IoT threats. The need for an automated system capable of handling the massive amount of incoming data lead us to conduct this research, with our main goal being for the technology to be able to detect and quickly act against IoT attacks – automatically.

In view of the fast-increasing number of IoT threats, traditional cyber defenses are no longer sufficient to deal with threats active in today's world. To help solve this problem, we decided to build and refine an AI-based system for the active hunt of cyber threats. This system, after over a year of development, has become able to hunt and analyze threats from all over the world in real time. The streamlined and automated design allows researchers to focus on the work of analysis and finding ways to apply the invaluable data that can be uncovered.
Let's talk about the benefits of proactive and automated threat hunting. Threat hunting through a fully automated threat hunting system has the following advantages:

1. Automatic detection and real-time blocking of various threats
2. Instantly locate various threat trends
3. Follow-up analysis of a large number of intelligence resources
4. The cost of human maintenance is extremely low

Before we built the IoT and ICS threat hunting system, we conducted an in-depth analysis of various possible implementation strategies and methods. We concluded that our hunting system must have several key features:

1. **Scalability:** For a 24/7 hunting system that hunts threats without interruption, the scalability of the network is of utmost importance. The transmission of network traffic and the output of the threat hunting system must be able to adjust flexibly in real time to handle different situations. The back-end processing also needs to have a dynamic distribution mechanism to ensure that the traffic will get the corresponding server resources for data analysis based on the conditions at different times.
2. **High availability and stability:** Overall service availability and stability are essential. If there is an abnormality in the transmission mechanism, storage area, or other parts of the threat hunting system, it will inevitably affect the overall output.
3. **Easy monitoring and analysis:** There must be a mechanism that can monitor and locate in real time to facilitate rapid response and processing, which must maintain functionality when the various components and transmission mechanisms in the hunting system might be abnormal. The large amount of data in the hunting system must be able to interface with various data analysis mechanisms and have fast calculation and processing functions so that threat analysts can hunt threats quickly and effectively.
4. **Fast adjustment:** In response to ever-changing threats, the deployment area, location, and IP of the hunting engine will also be rapidly adjusted and converted in response to different landscapes.
5. **Data security:** Any data we hunt must be stored safely and properly.

Based on the above requirements, we decided to fully embrace the cloud environment for our platform.

THE ANATOMY OF OUR IOT AND ICS THREAT HUNTING SYSTEM

Our IoT and ICS threat hunting system's architecture and processes, were designed with specific features in mind.

The Threat Hunting System's Architecture as Shown by Data Flow

The full data flow of our hunting process can be seen in Figure 1. This is how we connect data from the Internet to our Hunting System Cloud, which is protected and deployed in Amazon Web Services (AWS). Within our hunting system, we have divided the process into 7 steps.

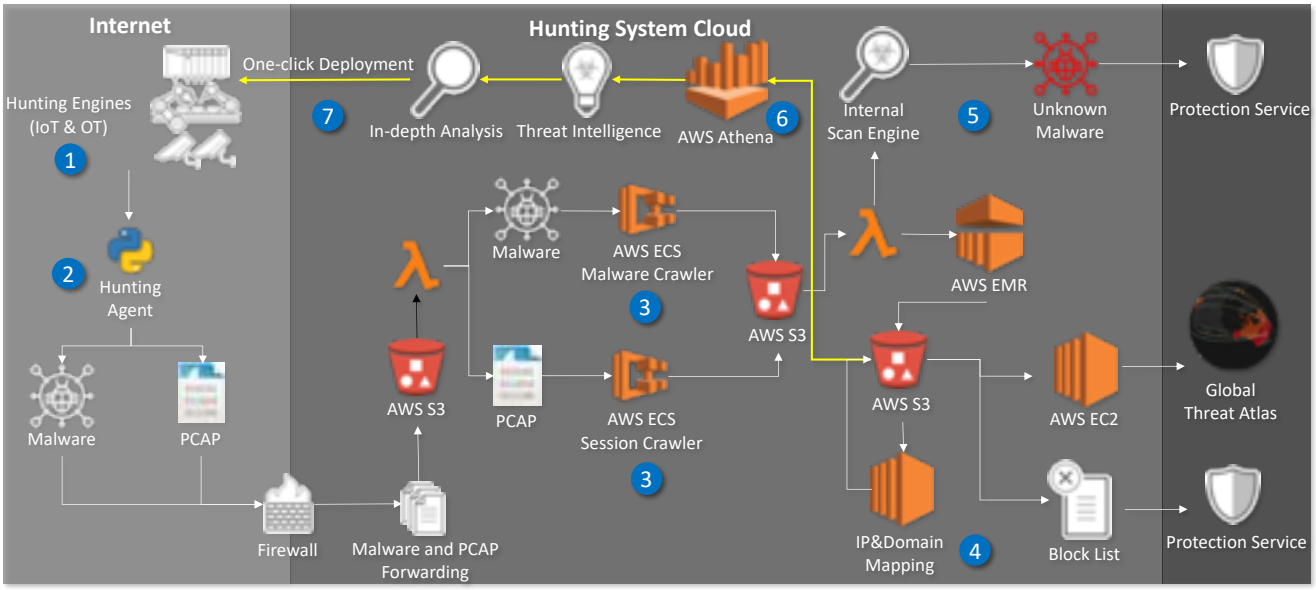

Figure 1. The Architecture of the IoT and ICS Threat Hunting System

Our threat hunting process runs on an hourly cycle, as our automated threat hunting system uses the hour as its base unit. The traffic and information our hunting engines are collected every hour. Step 1 to Step 5 are fully automatic processes, requiring no human intervention. Step 1 to Step 3 gather and process hunted data, and then Step 4 and Step 5 are used to generate indicators of compromise (IoC) for discovered malware. Step 6 and 7 are used to hunt and analyze in-depth threats, which is where our threat analysts take over.

The Threat Hunting Process

Step 1. Data Collection via Interaction

To hunt global threats, we deployed our hunting engines globally. We used cloud services such as Amazon Web Services (AWS) to deploy our hunting engines. We deployed over 350 hunting engines to data centers all over the world using these cloud services. Figure 2 shows, roughly, the geographic distribution of our hunting engines.

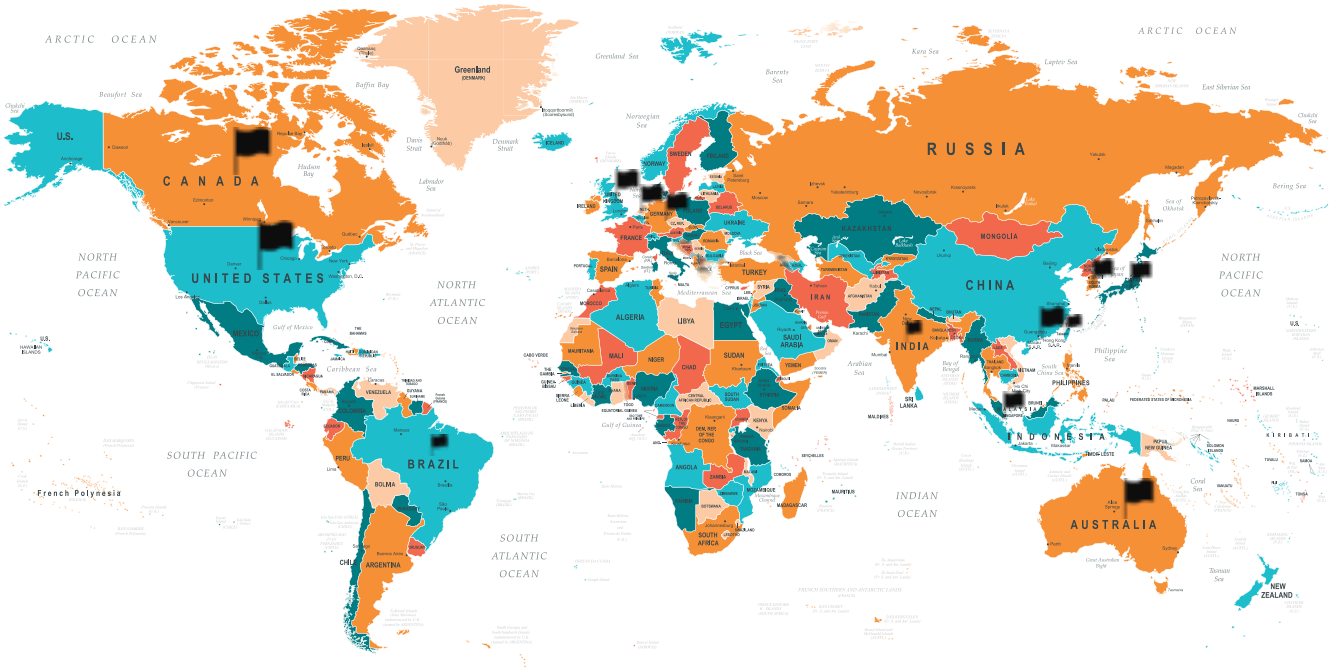

Figure 2. Geographical Distribution of Our Hunting Engines

In **Step 1**, all of our hunting engines interact with attackers, collect traffic, and conduct partial analysis of recorded traffic. We imported and aggregated open-source honeypot modules to create a new and improved base model for our hunting engines. We took care to ensure that each instance of the hunting engine could be stably synchronized with our hunting system cloud. Through this interaction with attackers, we gather attack traffic, malicious samples, and attacker information. Other gathered intelligence, which we usually don't initially analyze, will be passed to the hunting agent and load balancer for later in-depth analysis.

Please see the data flow of our hunting engine as diagrammed in Figure 3.

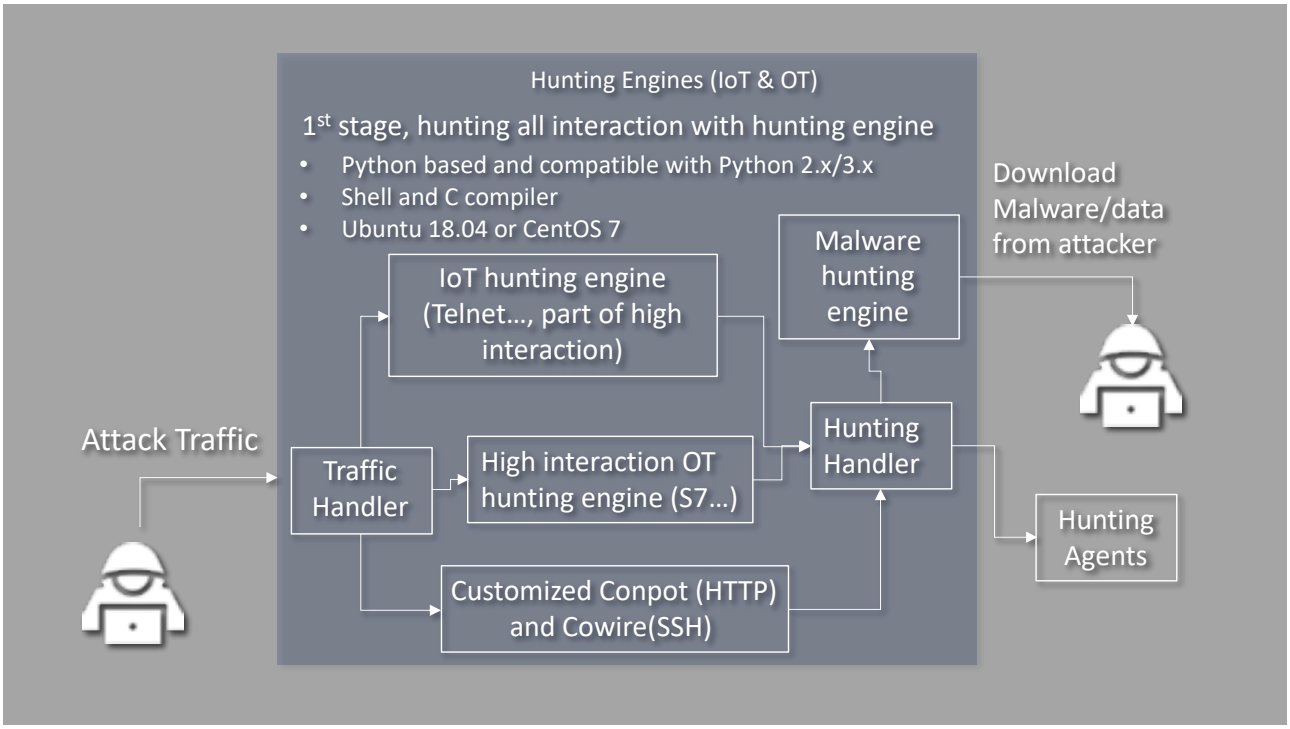

Figure 3. Hunting Engine Process

To ensure the engines can offer high interaction to intruders, we integrated open-source honeypot projects' such as Cowrie [3] and Conpot [4]. This allowed for more attacker commands, interaction improvements with the core and shell modules, and authentic-seeming response messages designed to more effectively deceive attackers. Our engines are able to transmit traffic, record interactions, and pass information to our load balancer for analysis via the use of MTPot [5].

The system can present intruders with a full simulation of PLC operation, as well as many functions of the Siemens S7 protocol. Other OT protocols such as Modbus/TCP and EtherNet/IP will be added in the near future, but in the meantime we can still identify and analyze the traffic on these protocols. We designed the system for continuous monitoring and to be enhanced over time.

As it gathers data for us, the hunting system will change its IP address regularly, making it more difficult for attackers to find. Also, IoT scanning engines such as Shodan are unable to recognize the hunting engine as a honeypot.

When attackers access the hunting system directly, it will immediately collect their IP, port, domain, and other available information. Internally, the hunting system builds a simple table which it uses to map the attacker's information and payload.

When intruders interact with an engine using a protocol which the engine can understand, it will provide them with realistic and believable responses. For example, our S7 hunting engines can reply to attackers' read requests with specific data from fake S7 memory.

When the hunting system receives attack traffic on known protocols such as HTTP, SSH, Telnet, SMB or Siemens S7 (which are commonly used in industrial control systems), it will conduct preliminary identification before acting like a switchboard as it connects the traffic with an engine that will be most responsive to their attack. For example, if the traffic handler recognizes incoming traffic as Telnet traffic, it will direct that traffic to the IoT-based engine that can give appropriate responses to Telnet. This way, the IoT engine provides authentic-seeming interaction behavior to supposed intruders by allowing further interaction with the attacker on the Telnet protocol.

If the attack traffic is related to an ICS protocol, such as Siemens S7, it will be directed to the OT-based high-interaction engine for further interaction. Through our in-built support for high interaction, attackers believe themselves to be communicating with a real PLC. We based our design on ICS-specific protocol operation and the results of analyzed payloads. We used this the basis for a high-interaction application that simulates a PLC as completely as possible. Unknown attack traffic, such as unfamiliar industrial control protocols or incomplete traffic, is directed to the back-end for further processing and analysis.

After interaction, hunting engines retain data on the different kinds of attack traffic received, which is then integrated into the hunting handler. When the hunting handler receives the traffic, it will perform the first filtering. The filtering mechanism is used to analyze whether there are possible download behaviors or links in the traffic. The malware hunting engine will also download and package suspicious files (executables, scripts, etc.) for analysis.

Step 2. The Hunting Agent

In Step 2, the hunting agent aggregates the different kinds of traffic captured by the threat hunting engines, particularly received files. It divides them into malware/suspicious files and

PCAP (traffic) after ensuring their integrity. The logs are then transferred to the hunting system cloud. Our private hunting system cloud protects data transmissions to to AWS S3 using a firewall with a trust list.

Before malware is forwarded to AWS S3, it will be parsed, crawled, and compressed. This is to avoid making our hunting system vulnerable to the files it must handle to do its work . By using malware, PCAP forwarding, and pre-process processing, our hunting system can integrate malware simulations, data visualization, and other cloud services.

Our hunting agent will return hunted data every hour. S3 also uses the hour as the basis for its data segmentation. This is to avoid considerable loading and inconvenience for our researchers when they conduct data engineering and analyze specific threats.

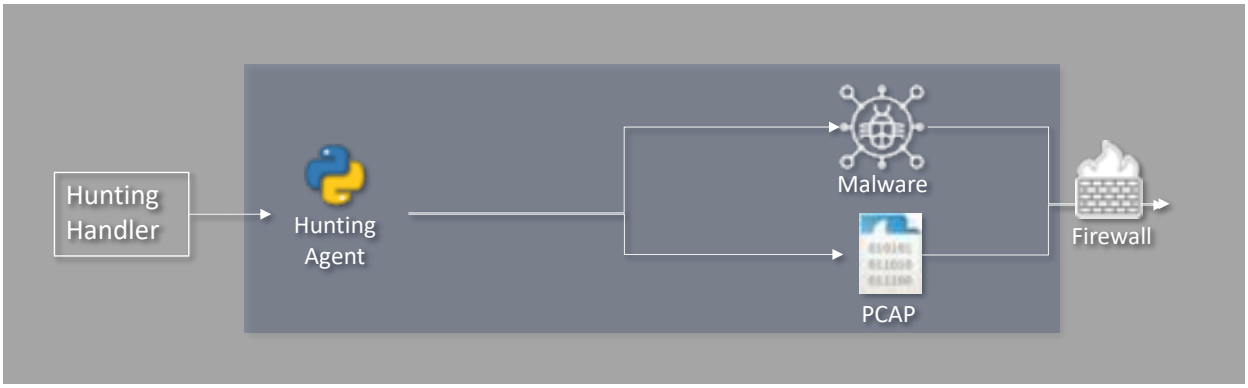

Figure 4. Hunting Agent Process

Step 3. Malware Crawler and Session Crawler

There are two crawlers working as active parts of our threat hunting system: the malware crawler and the session crawler. In Step 3, the crawler parses, crawls, and compresses data before uploading it to AWS S3. As it parses, the crawler gathers URLs and C&C servers. This process happens on an hourly cycle.

The malware crawler will detect the connection status of the C&C server and download samples from it. It also gathers materials through the hunting system before the connection to the C&C server expires. This way, the hunting system will be able to analyze threats with the newest materials and information gathered directly from the C&C server. The malware crawler also allows the hunting system to track the source of the malware samples, adding depth to our research results.

The session crawler is responsible for parsing the content of the PCAP, where it attempts to find specific command injections, URLs, and anything else that might be considered strange. If the command injection includes a retrieval process, the session crawler will refactor the injection command to collect any malicious materials. If it collects URLs from the contents of a PCAP, the hunting system will map them to confirm their purpose. The mapping table lists all malicious URLs from the PCAP sessions by hour. This list helps researchers to understand the relationship between the payload, the URLs, and captured malicious files.

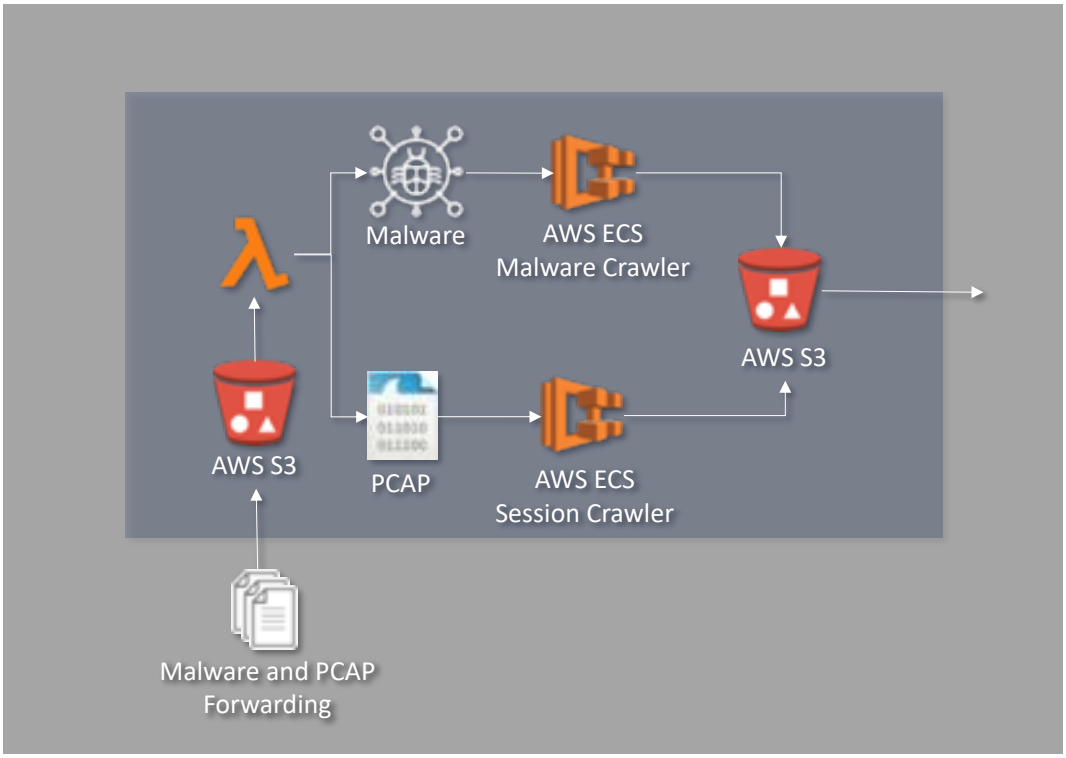

Figure 5. Malware Crawler and Session Crawler

Step 4. Generate IoC to Block List

After the hunting system analyzes the PCAP and malware, it produces lists of malicious IPs and URLs. The hunting system will review these lists through internal services to map the IP and domain name. When it reviews the IP list, the hunting system will query internal services to confirm if each IP belongs to a shared IP or not. If the IP belongs to a public or shared IP, it means the attacker is hiding behind this IP, so blocking it would be a waste. We will also filter out specific public IPs that will not be blocked – the hunting system will filter this IP to avoid the IP being inserted into a block list.

When reviewing the domain list, the hunting system also queries internal services to check the domain name's ownership details. If the domain name isn't defined as a trusted domain, it means this domain is unknown or malicious, so the hunting system will insert it into a block list. These block lists will later be used to provide protection from malicious domains. The mapping process will match block lists and trust lists which include many trusted domains. This is to avoid the protective service blocking trusted IPs or domains. The hunting system has multiple such processes like this one to ensure that the lists it generates are trustworthy and reliable to internal services.

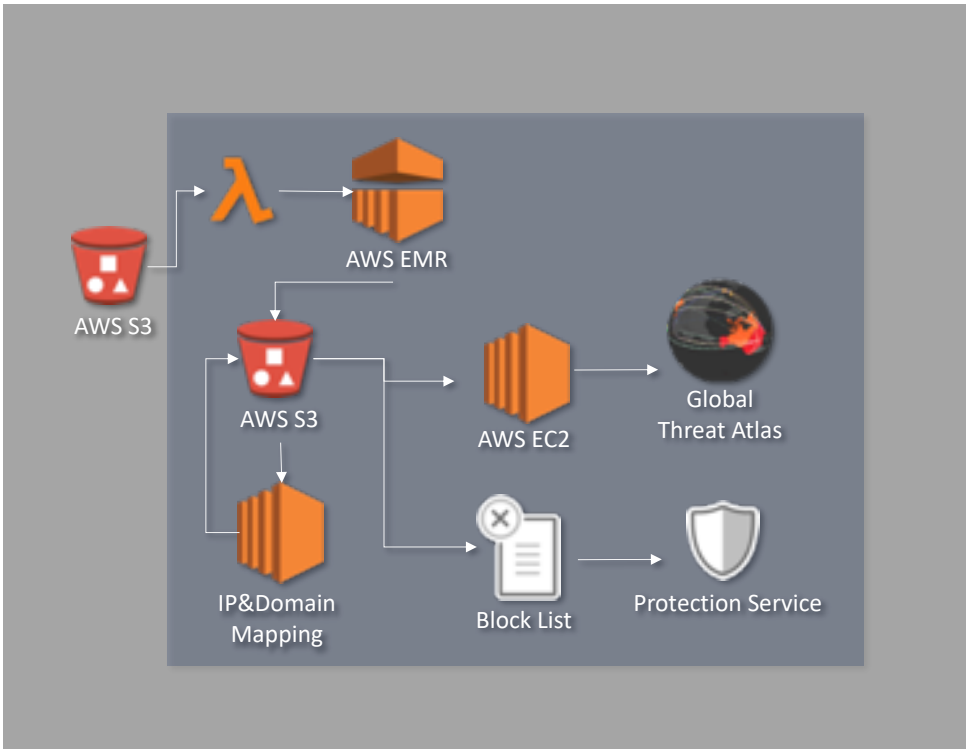

Figure 6. Generating IoC to Block List

Step 5. Malware Analyzer

The hunting system's malware analyzer is built into the container of AWS ECS, where it's triggered every hour as discovered pieces of malware are uploaded. Before the malware is collected from the hunting agents for storage and analysis, the agents will run some pre-processes.

For ease of filing, they change each piece of malware's name to an individual SHA-1. The agent then builds a malware table to record details about the malware's SHA-1, C&C server of origin, and the attacker's IP. The malware will then be compressed with a password and sent to AWS S3, where it will automatically trigger the malware analyzer.

This analyzer uses multiple processes to analyze collected pieces of malware. First, we get the threat name and use it to query VirusTotal. If VirusTotal does not recognize the malware, we know it's one of many totally new pieces of malware our hunting agents discover each hour. Secondly, we use our internal scan engine to re-scan the malware. Based on what the scan engine finds as it analyzes the malware (signatures, etc.), we will enhance our internal scanner's malware detection patterns. The known malware we received is handled by a much shorter process – we just record it in the malware table and then upload the table to AWS S3.

The malware analyzer helps our hunting system classify the threat type of each piece of malware and give a definition to malware that isn't yet classified. The hunting system contributes anywhere from tens to hundreds of pieces of new malware to our protection service daily. After the newest malware is classified, internal researchers will analyze each piece manually. Our malware analyzer has significantly decreased malware investigation and analysis time for our internal researchers.

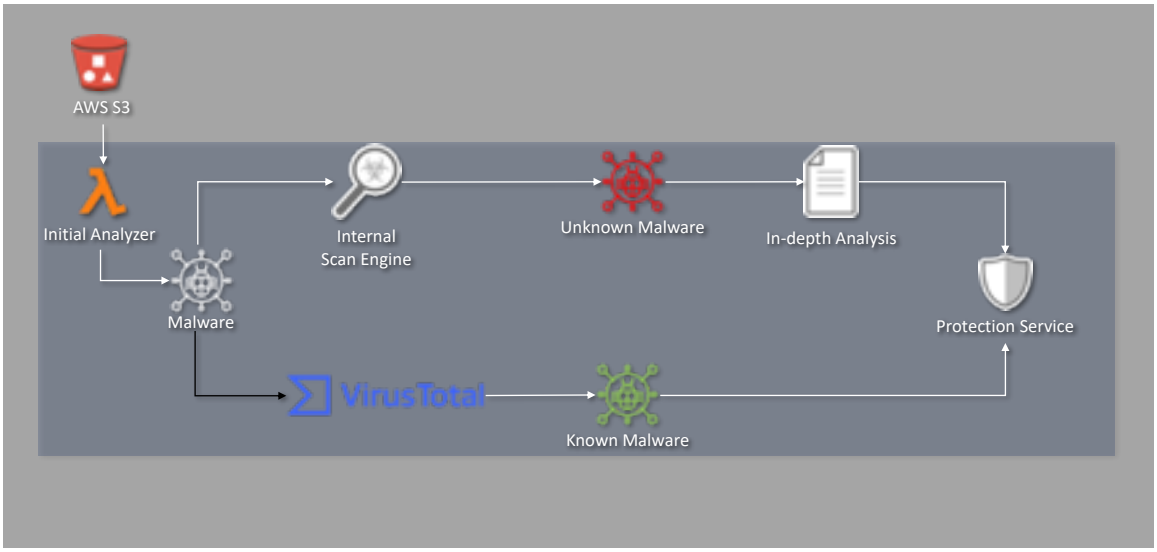

Figure 7. Malware Analyzer

Next, the threat analyst manually hunts down the in-depth threat. However, as mentioned previously, the number of threats from IoT and ICS is too many to be dealt with manually. Therefore, in order to facilitate threat analysts to quickly address targets, we have designed some automated auxiliary mechanisms, including:

1- Global Threat Atlas

The visualized map allows us to quickly converge the current attack trends and the distribution of known threats.

Figure 8. Global Threat Atlas [7]

2- Attack Payload Integration Notification

Email notifications are sent out 3 times per day detailing the last 8 hours of payload (high similarity aggregation determined by customized algorithm), region, count, and signature hits.

3- Customized Hunting

Threat analysts will more deeply automate the content that they want to hunt and analyze the clues they find to facilitate the timeliness of analysis. For example, every day we monitor international information security incidents and the status of vulnerability notifications and releases, as shown in Figure 9 and Figure 10. We will conduct a deeper analysis of these threats and determine whether or not we need to update our hunting engines around the world to ensure that we are capable of detecting these threats.

Figure 9. Automated News Parser

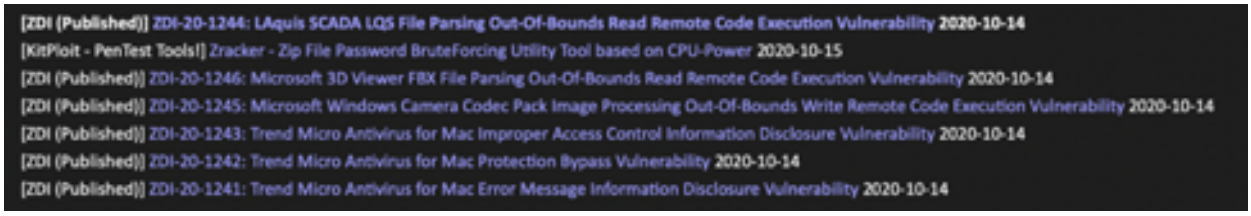

Figure 10. Automated Vulnerability Parser

Step 6. One-Click Deployment/Re-Deployment

This is a function set up to strengthen our automated process. Threats are constantly evolving with each passing day, so that our hunting engines all over the world must highly-tuned detection capabilities that can compete with such adaptivity. In response to constantly updated threats, we often need to deploy new hunting engine images to our hunting system

Features of Our IoT and ICS Threat Hunting System

Hunting Engine

After undergoing several adjustments and enhancements, our hunting engine can identify more than 30 protocols across IoT and ICS, use high interaction-based strategies to trick attackers and land their malware, dynamically unpack and analyze payloads to export data for research, analyze malware information in real time, hunt suspicious files from attack payloads and malware samples, and build a proxy infrastructure to hide behind an extranet.

Dynamic Analysis

Our threat hunting systems will automatically dynamically adjust the analysis process based on hourly traffic size. This mechanism solves two key problems in data processing:

- If the system has too much data to export it might cause delays – we were able to set this up so it wouldn't impact later system operation and data processing.
- We don't need to dedicate a lot of powerful machines to do data processing and this efficiently cuts down costs.

In-Depth Analysis

While normally honeypots are analyzed using log files, we conduct our analysis in a totally different way to perform detailed analysis (IP, domain, payload, malware, etc.) based on PCAP via our load balancers, which are built into the AWS ECS service. This process allows us to efficiently get in-depth analysis on any malicious traffic. At the same time we won't miss any suspicious information. Computing resources aren't a problem, because the load balancer will be scaled automatically based on hourly data input. For in-depth analysis, we also built in a way for our threat hunting system to pre-process all raw data, before joining and mapping it with related fields of all raw data. Through these pre-processes, we can compare each unique piece of raw data with the others.

Payload Classification

When we get an unknown payload, we analyze it. We then identify internal patterns in those malicious payloads and integrate these new patterns into our hunting system. Through the payload classification process, we can review and categorize these patterns, allowing unknown and known payloads to easily be told apart. Payload classification, which executes in the hunting system automatically, helps our hunting system to optimize distinguishability. It means that we can monitor the classified result and update our internal pattern findings in real time.

One-Click Deployment

For the purposes of management, one-click deployment is prioritized for development and further streamlined with each enhancement of our hunting engines.

Construction Costs are Gradually Decreasing

We will automatically freeze data more than 1 year old -- that is, our regular hunting timeline defaults to a maximum of one year. If it is necessary to analyze threat data more than one year old, the data must be thawed (24-48 hours later) before analysis.

NEXT GENERATION IIOT THREAT-HUNTING SYSTEM

What is a Next Generation Threat?

With the maturity of Industry 4.0, many industrial control-related devices support direct networking functions. At the same time, according to the results of our recent observations, probes or attack traffic related to ICS communication protocols are very common. Attackers are gradually turning to IIoT-related fields to target industrial applications and production processes. The number of connected IIoT devices will only continue to climb rapidly for the foreseeable future, and this will be the crux of most hacker attacks. In summary, we believe that attacks and threats against IIoT will become even more severe, especially as they become more specialized to target different industries.

The Next Steps of Next Generation IIoT Threat-Hunting System

To keep our next-generation IIoT threat hunting system up-to-date, and more importantly continue to move it into the future, we're working together to create a high-performance and high-precision hunting system for next generation threats.

Our goals are:

1. Bringing the complete industry 4.0 environment into our hunting system, fully virtualized. This will allow us to present attackers with better deceptions, simulating various critical infrastructure scenarios including smart factories and power plants, or various related pieces of equipment such as Programmable Logic Controllers (PLC), Human Machine Interfaces (HMI), and Field Devices.

2. In the short-term we plan to import real devices int our ICS/SCADA laboratory as parts of our hunting engine, significantly shortening development time. The architecture is shown in Figure 11.

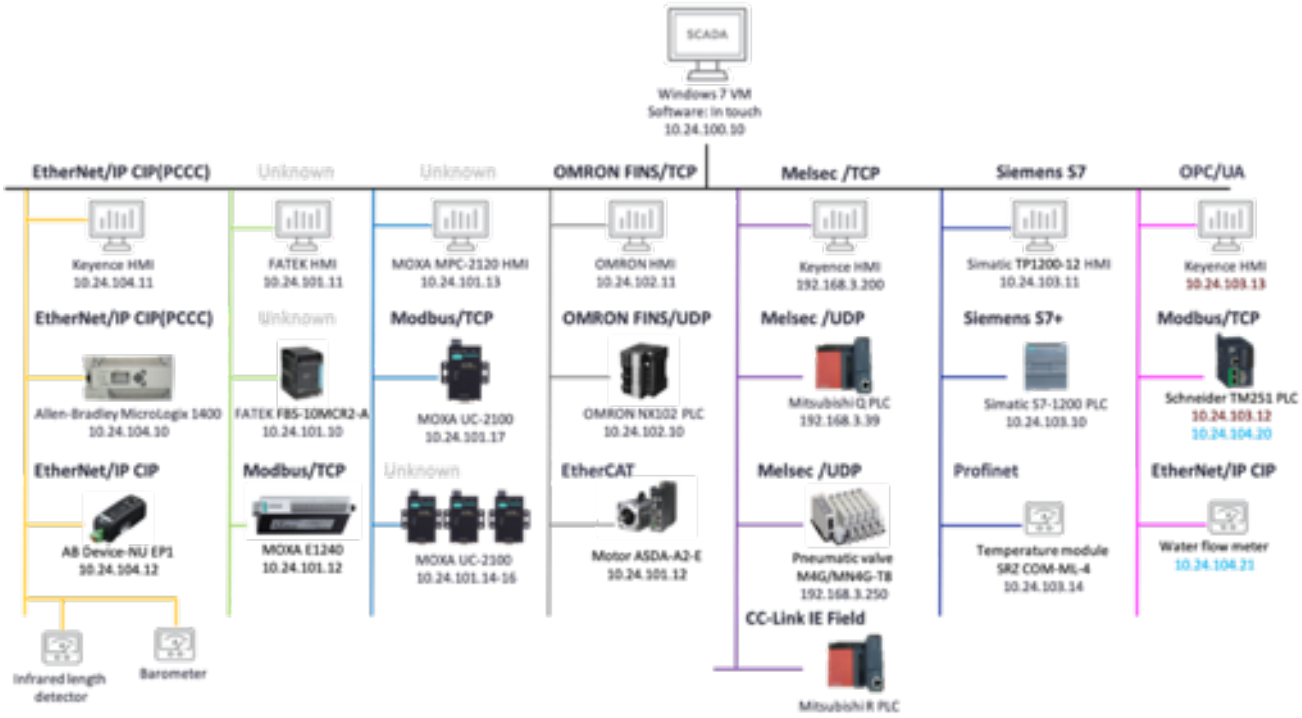

Figure 11. The Short-term Architecture of Next Generation IIoT Threat-Hunting System

3. In the long-term:
   A. The next generation of our hunting system will coexist with the existing IoT hunting system, increasing our ability to cover the scope of IIoT. Our system must make it impossible for an attacker to distinguish that its attack target is forged, and be able to simulate various attack behaviors that attackers might want to perform.
   B. To advance our system's ability to carry out automated analysis, we will conduct an in-depth study of attack traffic and malware, including the various ways machine learning can be applied to traffic and malicious program analysis. Automated sandbox analysis is intended to be included in the near future.
   C. For vulnerability analysis and exploitation, 0-day/1-day vulnerabilities related to IIoT will be gradually introduced into the next generation hunting system in order to build a hunting system that can analyze IIoT threats at a macro level.

CONCLUSION

The creation of an automated system has made marvelous and efficient changes to our procedures for hunting and fighting continuously expanding IoT and ICS threats. The number of these threats will continue to grow in number. We must maximize our capacity to act fast in response to such threats, as unchecked they will only become more and more reckless. The future will certainly hold cyber attacks that leverage human lives and major disasters as part of extortion campaigns.

Our automated threat hunting system can effectively detect new threats in real time and block many malicious attacks. In the near future we will release a second white paper, sharing 6 cases that show just how game-changing these resources can be when leveraged effectively.

REFERENCES

[3] "Cowire." https://github.com/cowrie/cowrie (accessed 11/17, 2020).

[4] "Conpot." https://github.com/mushorg/conpot (accessed 11/17, 2020).

[5] "MTPot." https://github.com/Cymmetria/MTPot (accessed 11/17, 2020).

[7] "Global IoT/ICS Threat Atlas." https://www.tr.txone-networks.com/ (accessed 10/10, 2020).

Created by:

The Global Threat Research Group of TXOne Networks

TXOne Networks is a joint-venture company of Trend Micro and Moxa. TXOne Networks is mainly offering cybersecurity solutions to protect industrial control systems. Trend Micro has more than 30+ years of cybersecurity threats intelligent and MOXA has more than 30+ years of OT network expertise, which makes TXOne Networks have both IT and OT technology to provide the comprehensive adaptive ICS cybersecurity solution. TXOne Networks leverage those advantages to develop the ICS cybersecurity products including endpoint security and network security, both Trend Micro and Moxa are not just providing the technology and knowledge, they are also taking care the go-to market channel for both sales and support service in IT and OT.