VEDERE LABS

R4IoT:

Next-Generation Ransomware

A view into what can happen when ransomware meets IoT and OT

Table of Contents

1. Executive Summary

In this report, Vedere Labs demonstrates R4IoT: a proof of concept for **next-generation ransomware** that exploits IoT devices for initial access, targets IT devices to deploy ransomware and cryptominers, and leverages poor OT security practices to cause physical disruption to business operations.

- The need for a study like R4IoT emerged from the observation of an increase of the number and diversity of IoT, IoMT and OT devices connected to standard corporate IT networks. Such devices increase the risk posture in nearly every business that has to now deal with the growth of IoT in corporate networks, IT/OT convergence and the rise of supply chain vulnerabilities.
- R4IoT is the results of Vedere Labs' continuous analysis of how ransomware gangs have been evolving in past years. Besides adding new layers of extortion, such as data exfiltration and denials of service, major gangs such as **Conti** and **ALPHV** have been focusing on exploiting network infrastructure devices and increasing the sophistication of their ransomware payloads.
- The intent of a study like R4IoT is to prepare businesses and cybersecurity at large to deal with an inevitable increase in sophistication and scope of traditional ransomware by:
  - providing a step-by-step demonstration of how IoT and OT exploits can be combined with a "traditional" ransomware campaign, and
  - providing a playbook for mitigating this emerging type of attack by relying on complete visibility and enhanced control of all the assets in a network.
- A video showing R4IoT in action can be found **here**.

2. Introduction

In 2021, the cybersecurity community saw many instances of devastating cyberattacks that led organizations to lose huge amounts of money or to temporarily halt their operations. Among them:

- In February, **Oldsmar water treatment plant employees noticed that sodium hydroxide levels were rapidly rising** on their computer screens. Someone accessed the treatment system using the remote connectivity tool TeamViewer, but employees thwarted the attacker from moving laterally into other IT infrastructure.
- In May, Colonial Pipeline was hit by a ransomware attack that caused a gas crisis. The attackers, known as Darkside, gained access through a **VPN that did not require multifactor authentication**. Although Darkside took control of Colonial Pipeline's IT systems, once Colonial Pipeline knew its IT operations were affected, the company chose to proactively take its OT systems offline to prevent the attack from spreading.
- Also in May, JBS Foods was attacked by another ransomware gang, REvil, and forced to shut down its facilities in several countries before **paying $11 million to recover access** to its systems.
- In July, **Iran Railways had to shut down its train operations due to a hacking group** infiltrating an IT system and spreading malware. Iran has not been forthcoming about the details of this attack, leading security researchers to form their own hypotheses.
- Also in July, malicious actors combined a supply chain attack vector with a ransomware payload in the Kaseya VSA incident. REvil, the same group that previously attacked JBS, was able to use the Kaseya remote management tool (VSA) to infect managed service providers and their customers with ransomware. In total, more than **1500 organizations** were hit simultaneously.

While the Oldsmar and Iran Railways incidents show what individuals or small groups of attackers can achieve against critical infrastructure operators, the Colonial Pipeline, JBS Foods and Kaseya incidents are part of a growing and alarming trend: large ransomware gangs, often operating a Ransomware-as-a-Service (RaaS) model, crippling the operations of several types of organizations, often at the same time.

Ransomware was without a doubt the biggest threat of 2021 for most organizations. This was already a known problem in previous years, but attackers have been evolving quickly and have moved from purely encrypting data until circa 2019 to exfiltrating data before encryption in 2020 to large extortion campaigns with several phases in 2021. The trend continued in early-2022 with the emergence of new and very sophisticated ransomware families such as **ALPHV** and more attacks by RaaS groups such as **Conti,** which have even taken a political position after the **Russian invasion of Ukraine**.

This evolution in attacker methods means that ransomware gangs can now cripple the operations of virtually any organization. For that reason, the response to ransomware has been gaining momentum. In January 2021, Emotet, a cybercrime group that develops a malware loader frequently used by ransomware gangs, was disrupted in a **global action coordinated by Europol**, while another global action arrested members of **REvil in January 2022**. In October 2021, United States President Joe Biden issued a **public statement** on cybersecurity and **convened a meeting of 30 countries** to increase their efforts to combat cybercrime and ransomware specifically.

Successful response to ransomware depends not only on legal and political action but also on equipping organizations to be able to defend themselves. In this report, we demonstrate two things: first, that the evolution of the ransomware threat landscape is far from over because attackers still have a large attack surface to explore, and second, that there are ways to mitigate both the likelihood and the impact of attacks on organizations, thus decreasing the overall risk to which these organizations are exposed.

We explore the current state of ransomware attacks (Section 4) and business networks (Section 6) to discuss how ransomware could evolve in the coming years because of two ongoing trends: (i) the proliferation of IoT devices in enterprise organizations, and (ii) the convergence of IT and OT networks. We created a proof-of-concept ransomware (Section 7) that exploits the first trend by using exposed vulnerable devices, such as an IP camera or a Network Attached Storage (NAS) as initial access point, and the second trend to hold OT devices hostage, thus adding another layer of extortion to an attack campaign. Finally, we discuss how cybersecurity controls aligned to mature frameworks can be used to detect and stop this attack or, even better, prevent it from happening in the first place (Section 8).

3. Why R4IoT, Why Now?

R4IoT novelty resides in the following key contributions.

- This is the **first and only known work to combine the worlds of IT, OT and IoT ransomware and to have a full proof of concept from initial access via IoT to lateral movement in the IT network and then impact in the OT network.** Beyond just encryption, our proof of concept on IT equipment includes deployment of a cryptominer and data exfiltration (also known as double extortion).
- The impact we demonstrate on OT **is general purpose:** it is not limited to standard operating systems (e.g., Linux) or device types (e.g., building automation), does not require persistence or firmware modification on the targeted devices and **works at large-scale on a wide variety of devices impacted by TCP/IP stack vulnerabilities.**
- We discuss future scenarios where the OT impact could be launched remotely (as in the current case of Ransom Denial of Service targeting exposed IT systems).
- We implemented detection and response actions for the attack that serve as a playbook for organizations looking to defend against both current and future threats.

Although R4IoT is unique in its kind, in the past five years, other researchers have discussed around the possibility of ransomware extending to IoT and OT and they have produced small-scale demonstrations of how such interplay between ransomware and IoT or OT devices could work. We list such previous works below.

- In 2016, Andrew Tierney at PenTestPartners **demonstrated a proof of concept** to lock a user out of a thermostat until a ransom was paid. This PoC worked by changing the firmware of the device so the user could not access its settings, and the attacker could set the temperature to any desired value.
- In 2017, Stephen Cobb at ESET coined the term "**jackware**" for ransomware that affects IoT devices through hijacking. That paper discussed some possible scenarios for jackware, mostly focusing on the automotive industry. In 2019, the same researcher coined the term "**siegeware**" for ransomware that affects building automation devices. Those works were theoretical analyses extrapolating from real-world incidents without actual implementation, but both terms have gained some popularity (e.g., **AT&T**, **Gartner**, **Sophos**).
- In 2020, Brierley et al. published **PaperW8**, a proof-of-concept ransomware that works on multiple Linux-based IoT. The goal of their PoC is to infect devices, display ransom notes on those devices and threaten to permanently brick them. The same team published in 2021 another PoC that focused on **data-stealing ransomware**, where the data stolen comes from IoT devices, such as audio, video and sensor feeds.
- In 2021, David Nicol analyzed the trend of ransomware attacks affecting IT systems of **energy delivery organizations** and discussed characteristics of OT systems that would make them susceptible to ransomware attacks, such as embedded web servers and rogue devices.

4. The State of Ransomware

4.1 Threat Actors' Motivation

Threat Actors are after money. It is safe to say that ransomware is now a billion-dollar industry, with the market leaders taking in tens of millions of dollars per year.

According to the **Verizon Data Breach Investigations Report (DBIR) 2021**, more than 80% of cyber incidents have a financial motivation and are perpetrated by organized criminals. Ransomware is currently how cyber criminals get their money.

Ransomware is very lucrative, and some of the biggest ransom payouts happened in 2021. For instance, Colonial Pipeline and Brenntag reportedly **paid $4.4 million each to DarkSide**, whereas JBS paid **$11 million to REvil**. That amount does not account for lost revenue, the price of investigation and response, customer notification, fines and any other costs incurred beyond the ransom payment.

Although it's difficult to know exactly how much ransom was paid in total, the **US Financial Crimes Enforcement Network** investigated 635 suspicious activity reports related to ransomware just in the first half of 2021. Those reports had a total value of $590 million, which was more than the $416 million investigated in all of 2020.

Another **data source**, which relies on tracking blockchain transactions related to wallets known to belong to ransomware gangs, reports more than $44 million paid in 2021, with Conti receiving the biggest total payout at $16 million, REvil coming in second at $12 million and DarkSide coming in third at $9 million.

4.2 It's Not About Encryption, It's About Extortion

There is still a big misconception that ransomware means *malware for data encryption*. It started like that, but ransomware is about getting a *ransom* – extorting victims via cyberattacks. The goal of ransomware attacks is to force organizations to face a dilemma: pay the ransom and hope that attackers restore access to systems and go away, or don't pay and try to mitigate the effects of the attack with internal resources. There are many ways to force this dilemma currently. Besides encrypting data, ransomware gangs routinely take other actions to gain leverage and force their victims to pay, such as:

- Exfiltrating massive amounts of sensitive data and threatening to release it publicly. This is currently done by almost every ransomware and has become known as **"double extortion."**
- Unleashing distributed denial of service (DDoS) attacks against their victims during the ransom negotiation period. This method ("**triple extortion**" or "**ransom denial of service**") has been **gaining popularity**, and companies that routinely monitor DDoS attacks reported **record levels of attacks in 2021**.
- Publicly shaming or harassing their victims by contacting customers, partners and media outlets to announce the hack and make the negotiation public ("**quadruple extortion**").

According to **Sophos**, in 2021 there was a decrease in successful data encryption from 73% to 54% of attacks; nevertheless, there was an increase from 3% to 7% in the number of incidents where data was not encrypted but the victim still had to pay a ransom because of other extortion techniques.

4.3 Anatomy of Attacks

There are more than **1,000 different identified ransomware** variants, with the FBI having **stated in June** that they were tracking more than 100 active groups, each responsible for at least a dozen attacks.

Each ransomware group behaves slightly differently, using diverse tools, infrastructure and extortion methods. However, the tactics and techniques used during attacks are very similar. Figure 1 presents a high-level anatomy of a ransomware attack divided into three steps.

Figure 1 – High-level Anatomy of a Ransomware Attack

Initial Access: Threat actors gain unauthorized access to systems either by exploiting local or remote **software vulnerabilities** (e.g., **buffer overflows** or **command injection**) or by leveraging credential-based attacks (e.g., **brute forcing**, **password spraying**, **credential stuffing**).

Vulnerabilities in perimeter devices/services, such as VPN and cloud-based applications, have become **particularly popular** for initial access. Local vulnerabilities are usually exploited by phishing users into running malicious code, which is still the **most common form of compromise**.

- **Lateral Movement:** Once inside a compromised network, ransomware threat actors have three types of tools at their disposal: common exploit/pentesting frameworks (such as **CobaltStrike** and **Mimikatz**), bespoke hacking tools (which are increasingly **less popular**) and internal Windows tools (such as RDP, **WMIC**, **net**, **ping** and **PowerShell**). The use of internal tools is known as "Living-Off-The-Land" and is currently the **most common** (because they are usually already available and harder to detect as malicious). **RDP, for instance, was used in 90% of attacks in 2021**, in 28% of attacks it was used both internally and externally (i.e., for initial access), and in 41% it was used only internally (i.e., for lateral movement). These tools are used to scan the network (net, ping), obtain credentials (Mimikatz), disable security tools such as antivirus and firewalls, move from one machine to another (RDP, WMIC) and connect to a C2 server (CobaltStrike) to receive instructions.
- **Impact:** Once several machines have been infected, the attackers can exfiltrate collected data to the C2 or other servers and encrypt the files directly on local machines or over the network (using SMB shares). The attackers then leave a text file notifying victims of the attack and giving instructions for ransom payment. The amount paid by an organization to recover their data is usually lower than the initially demanded payment, which happens after a **negotiation period that can take dozens of turns**.

Those steps are often not all performed by the same group. Two very common trends today are ransomware as a service (RaaS) and initial access brokers (IABs). In the RaaS model, one group develops the ransomware encryptor and then distributes it to affiliates, who use it after they have gained access to an organization and who then share the received payments with the original developers. IABs are groups that sell initial access to networks, typically in the form of **valid credentials** (obtained via phishing or data leaks) or compromised machines via malware, such as **Hancitor**, **IcedID**, **Qbot** and **Trickbot**. Yet other parts of the criminal underground may enter the picture, such as **bulletproof hosting services**, which provide hosting for malware distribution, as well as command and control servers.

The steps taken by attackers can be more granularly categorized into common Tactics, Techniques, and Procedures (TTPs), for which there is a common framework called **MITRE ATT&CK**. When looking at five of the most common ransomware groups of 2021 (**Conti**, **DarkSide**, **Egregor**, **Maze** and **Ryuk**), the following **TTPs were the most popular**.

TACTIC	TECHNIQUE
Initial Access	T1078 Valid Accounts
Execution	T1059.001 - PowerShell
Command and Control	T1071 Application Layer Protocol T1573 Encrypted Channel (HTTPS)
Discovery	T1082 System Information Discovery T1057 Process Discovery
Privilege Escalation	T1053.005 Scheduled Task/Job
Collection	T1074.001 Data Staged: Local Data Staging T1560 Archive Collected Data
Exfiltration	T1041 Exfiltration Over C2 Channel (HTTPS)
Impact	T1486 Data Encrypted for Impact

In March 2022, Vedere Labs released a **threat briefing that analyzed leaked chats and documents of Conti**. In these chats and documents, the group explains some of its TTPs in more details, such as how VPN and RDP are recommended as ideal backdoors, and how Active Directory Domain Controllers are primary targets for persistence. **One of the discussion points immediately stood out to us: how IoT devices are a major initial attack surface**. They specifically mention how specialized hardware such as printers, routers and PLCs are often left unpatched and are not treated by defenders as a major risk. **They also discuss in their chats how to acquire devices to test specific exploits**.

5. What Future Attacks Could Be Like

Based on some trends described in Section 4, such as new extortion techniques and evolving complexity of attack campaigns, as well as other parts of the threat landscape that we will describe below, we discuss what the future of ransomware could look like from two points of view: initial access and impact.

We deliberately left out a discussion on lateral movement because this is a "solved" problem from the point of view of attackers with the use of commoditized exploit/pentest tools and "living off the land" as discussed in the previous section.

5.1 IoT and OT to Gain Initial Access

Ransomware groups could soon directly be using IoT and OT devices as entry points, or initial access brokers could be ready to acquire exploits and sell access to millions of those devices to other actors. This is because of the following reasons.

1. **Phishing is very effective but still depends on human interaction**. Vulnerabilities on IT perimeter devices and applications are being routinely exploited automatically, but they tend to be patched fast because of the immediate risk they expose. On the other hand, a growing number of IoT and OT devices connected to enterprise networks and actively **exploited** could provide valuable entry points for attackers because they are harder to patch and manage. IoT devices are currently compromised primarily to become part of large botnets that execute DDoS attacks, which started with **Mirai back in 2016** and has evolved toward modern malware such as **Mozi** and **Gafgyt**. These malware use either default and weak credentials or unpatched vulnerabilities to gain remote control of devices such as IP cameras, Network Video Recorders (NVRs) and routers. Modern examples such as **BotenaGo** pack more than 30 exploits for several types of devices. **Botnet operators could leverage the initial access provided by IoT devices to either deploy ransomware themselves or sell the access to ransomware affiliates**. There are already **examples** of IoT botnets used in ransom DDoS attacks and containing messages from known ransomware gangs.
2. **Exploits for IoT devices are frequently negotiated in darknet markets**, and other threat actors are starting to notice the potential of these devices. For instance, **Lemon Duck** is a Monero cryptomining botnet that uses IoT devices as entry points to infect computers. The Conti ransomware group targets devices, such as **routers, cameras and NAS with exposed web interfaces**, to move internally in affected organizations, variants of the Trickbot malware use **routers as a proxy to contact Command & Control** servers.Finally, the **Cyclops Blink** malware (linked to the state-sponsored Sandworm group) exploits routers for initial access.
3. **Some major breaches are already believed to be tied to exposed IoT or OT devices.** During the cyberattacks against the **Israeli water sector in 2020**, the attackers supposedly got access to PLCs via routers that exposed them to the internet. A similar case are internet-facing RTUs or gateways and converters. It's more likely that these devices are exposed to the internet than PLCs directly, in the case that asset owners don't use a private WAN for their geographically distributed infrastructure. These devices are increasingly Linux-based and, in many cases, are riddled with **known vulnerabilities or default credentials** that would allow for initial attacker access.

5.2 Impact Beyond Encryption

All the forms of extortion mentioned in Section 4.2 work very well for attackers, but as defenders increase their capabilities (from incident response to backups and even cyber insurance), attackers must come up with new types of impact to continue to get their payouts.

Ransomware was initially about denying access to files via encryption, but other forms of denial of service could become part of attack campaigns, such as **Telephony Denial of Service (TDoS)**, where attackers flood VoIP systems to deny communication, and siegeware, where attackers take building automation devices hostage (which happened in **real incidents** in 2021).

IoT devices could also be leveraged in other ways. For instance, hacktivists recently spammed several internet-connected **receipt printers** with "antiwork" messages. Sending ransom notes via the same printers and preventing them from being used for business operations would be an effective way to leverage those devices as part of a ransomware.

Cryptomining networks that hijack many computers to mine for cryptocurrencies is a rising trend and **less noticeable and risky for attackers than ransomware**; there have been many arrests related to ransomware but far fewer because of cryptomining. But combining both in the same campaign, such that critical devices are impacted by ransomware and less critical devices run unnoticed, cryptominers would give attackers another assurance that they will get a return on their investment. Otherwise, ransom groups could use cryptominers as a decoy while implanting encryptors, **similar to what other Advanced Persistent Threats (APTs) have done**.

Another trend is the rise of attacks targeting operational technology, **particularly internet-exposed devices**, and leading to loss of availability. Recent examples include threat actors **targeting UPS devices** via weak credentials and **EV charging stations**. Impacting OT field devices could add another layer to extortion campaigns focusing on critical infrastructure targets.

One thing that ties together both the initial access and impact possibilities brought by embedded IoT and OT devices is the increasing number of supply chain vulnerabilities affecting millions of these devices at the same time. Examples include **Project Memoria** affecting TCP/IP stacks, **BadAlloc** affecting RTOSes, **Access:7** affecting a popular IoT management platform and vulnerabilities in the **busybox** application used by many Linux devices. Exploiting supply chain vulnerabilities could allow attackers to greatly amplify the effect of attacks that were previously specific to some types of devices.

TECHNICAL NOTE: OT-SPECIFIC IMPACT CONSIDERATIONS

The predictions about initial access and impact above apply to organizations in any industry since the growth in the use of IoT and OT is not restricted to a specific sector. However, we would also like to add some considerations about future initial access and impact for OT environments.

It is simple to lock out and extort victims for Purdue Level 2 and above because those are regular Windows/Linux machines, but doing the same for PLCs is more complex. There has been prior academic work targeting specific PLCs by changing their configurations, however the implementation differs a lot between models/vendors and requires attackers to know what specific devices their victims run. Ransomware as a service needs to exploit economies of scale with minimal need for finetuning by affiliates. To build a threat that extorts OT environments at scale, attackers need to figure out a way to be able to 'lock' many different environments.

One option is to use network-level denials of service like we explore on the rest of this report (see Section 7.2.3). Another option is to focus on homogenous, high-impact environments like distributed control systems (DCS). Here the attacker has a guarantee that all the controllers in a victim are of a particular vendor, so they only need to develop a limited amount of access methods and controller payloads to have big impact in many environments. They could rely on firmware or logic downloads on the controller to drop a payload that disables engineering interfaces (so no further updates are possible) and starts a countdown on a logic bomb. This could be very simple like just strobe toggling all the inputs/outputs when it goes off (which requires no process comprehension).

Notifying the extorted victim that it has some time before the logic bombs go off in all its controllers puts pressure on paying the ransom. This is scalable since the attacker must only figure out ways to get code execution on the controllers for each major DCS once and then port the payload for each of them.

It's a one-time attacker investment for few major parties (e.g., ABB 800xA, Siemens PCS7, Emerson DeltaV and GE Mark VIe) that are used all over the world. Many of their controllers run on well-known RTOSes like QNX (Emerson, GE) and VxWorks (800xA) or have otherwise well-understood internals (PCS7). That way attackers don't have to port their malware to thousands of PLCs but can take a 'big game hunting' approach where they list companies known to use specific DCSs and target them directly with the guarantee that all controllers their controllers will be affected.

6. Reality Check – The Data Behind the Scenes

To show that the predictions in Section 5 are realistic, we analyzed data from Forescout Device Cloud, one of the world's largest repositories of connected enterprise device data —including IT, OT and IoT device data — with a number of devices that grows daily. The anonymous data comes from Forescout customer deployments and, at the time of this report's publication, contains information about 18 million devices from more than 1,400 global customers. Real-time visualizations of the data presented in this section are available online as part of [Vedere Labs' Global Cyber Intelligence Dashboard](#).

Figure 2 and Figure 3 show a breakdown of our data that allows us to make some interesting observations.

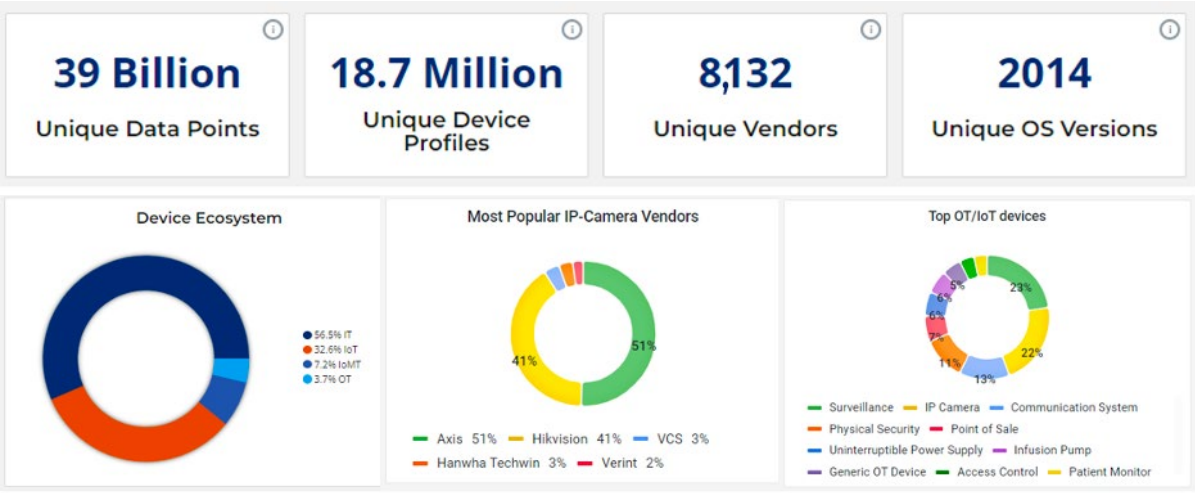

Figure 2 - Vedere Labs Global Cyber Intelligence Dashboard

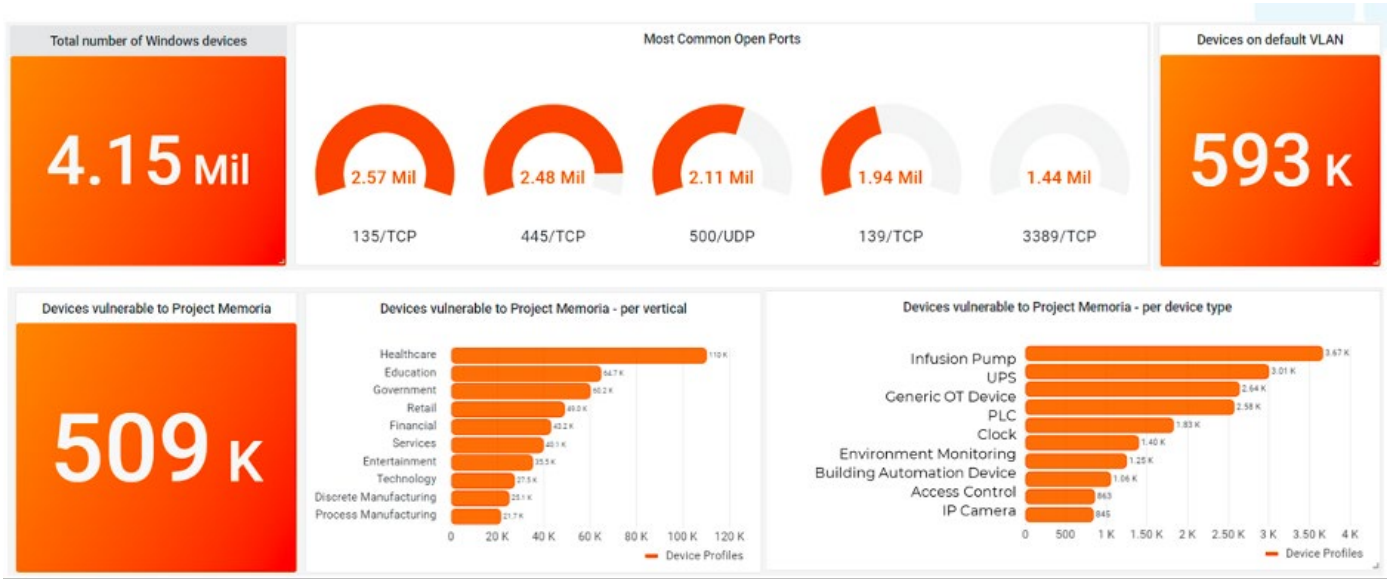

Figure 3 - Weakest Points on IT and IoT/IoMT/OT Devices

1. IoT, IoMT and OT devices combined represent 44% of the total devices in enterprise networks. **This means that ransomware threat actors focusing only on IT equipment are missing almost half of the available attack surface on organizations.**
2. Surveillance equipment, such as IP cameras and NVRs, represent 40% of these devices. **This means that attackers focusing on IP cameras are sure to find popular targets.**
3. Two vendors – Axis and Hikvision – account for 77% of the IP cameras in these networks. Axis cameras alone account for 39% of the ones observed. Models from **both vendors** have multiple known code execution vulnerabilities. **This means that weaponizing IP camera exploits as a reusable point of entry to many organizations (exactly what initial access brokers do) is feasible.**
4. Based on the data in Figure 3, of the 4.15 million devices running Windows OS in our dataset, more than 60% have an open WMI port (135/TCP), while roughly 35% have a RDP port (3389/TCP) open. **This means that "living off the land" using common Windows tools is feasible in enterprise organizations.**
5. There are more than half a million devices running TCP/IP stacks vulnerable to Project Memoria, spread out across organizations in almost every industry vertical. **This means that exploiting these devices with similar and simple denial of service attacks grants to attackers the possibility of disrupting many types of organizations.**
6. Healthcare is the most affected vertical, with more than 100 thousand devices impacted by Project Memoria. Among the most common OT/IoT devices are PLCs, building automation controllers and infusion pumps. **As we have described in previous research, healthcare organizations are great targets for attackers, partly because of the diversity of their device ecosystems.**
7. Our data shows **more than half a million devices using the default VLAN1**, meaning that segmentation is frequently not implemented. Network segmentation is a fundamental measure to limit the attack surface in any network. Segmentation is often achieved by a combination of different techniques at Layer 2 and Layer 3, including deploying VLANs, subnetting, ACLs and firewalling. There are several **important reasons** why user devices should not be left on the default VLAN – VLAN1 contains control plane traffic which a malicious device can tamper with to cause disastrous consequences, such as deletion of a VLAN database, performing VLAN hopping attacks and changing the root bridge, among others. While examining the VLANs with most IoT/OT devices, we noticed several VLANs containing a mix of IT and IoT/OT (i.e., IP cameras, building automation equipment and point-of-care diagnostic systems sit together with Windows workstations). Secure network segmentation should consider the context and purpose of devices rather than segmenting based on location, floor or department. **Mixing IP cameras and diagnostic systems – or other business-critical devices – in the same VLAN means that there is a path for an attack to spread from an insecure camera to a critical device.**

7. R4IoT: Creating a Ransomware in a Lab

To demonstrate the points discussed so far, we implemented R4IoT in our Vedere Labs locations (Figure 4). R4IoT is a proof-of-concept malware that combines an IoT entry point and typical ransomware lateral movement plus encryption on an IT network with an extended impact on both IT and OT. In the next subsections, we describe the technical details behind R4IoT. A summary of the attack can also be found in this **video**.

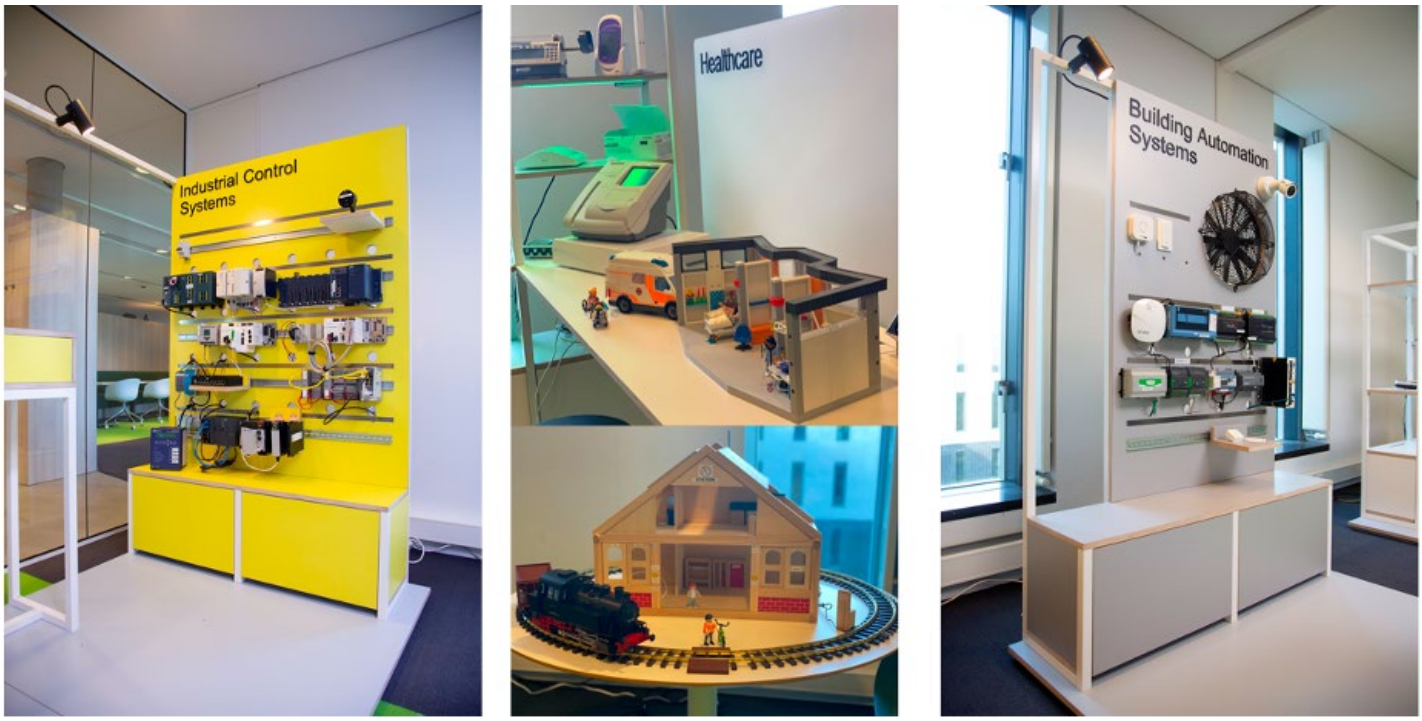

Figure 4 - Vedere Labs Facilities

7.1 Lab Setup

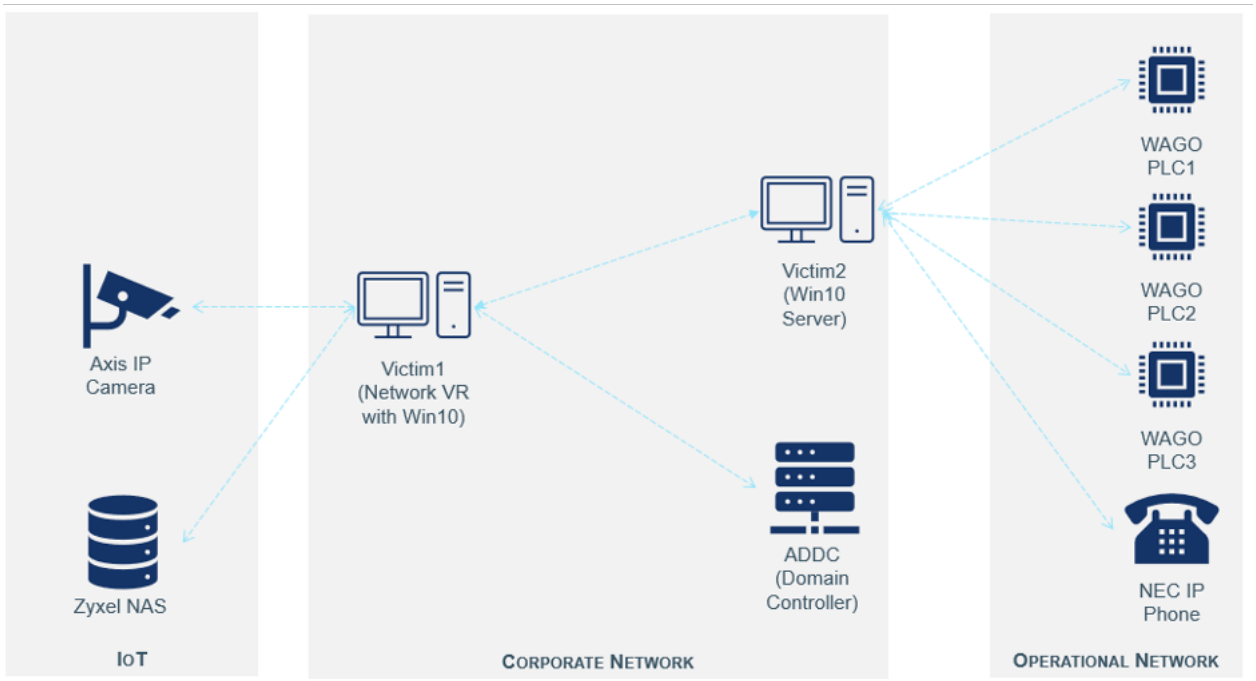

Figure 5 – Lab Network

Figure 5 shows the devices and networks in our lab, which is a simplified model of an enterprise organization with the following subnets:

- *192.168.85.0/24* – **External Network** (not shown in the Figure) – a local network that simulates the external network. We have chosen to use this network instead of the real Internet for security considerations.
- *192.168.4.0/24* – **Corporate Network** with Windows workstations. This is an internal network that is connected to other internal networks (see below). This network has limited connections to the **External Network** (managed by the Windows firewall): no devices from the "internet" can reach machines in the **Corporate Network**. Hosts in this network rely on Windows remote administration capabilities (such as WMI).
- *192.168.2.0/24* – **IoT network** with IoT devices connected to the **Corporate Network**. One of the devices (**Axis** M2025-LE camera) is misconfigured in such a way that it can be accessed from the **External Network**. This is a realistic scenario, as we see many IP cameras exposed directly over the internet (e.g., **Shodan queries** or incidents such as the **Hikvision hack**).
- *192.168.1.0/24* and *192.168.3.0/24* – **Operational Network** that holds several IoT and OT devices. These devices can be accessed only from the **Corporate Network**.

Our hypothetical organization consists of the following devices and machines:

1. **Axis** M2025-LE camera, vulnerable to **CVE-2018-10660**, **CVE-2018-10661**, **CVE-2020-10662** and **Zyxel** NAS 326 vulnerable to **CVE-2020-9054**. These are the only devices directly exposed to inbound connections from the **External Network**.
2. **ADDC** Windows server – Windows Active Directory Domain Controller (ADDC) machine deployed in the **Corporate Network**. This machine is not exposed to inbound traffic from the **External Network** and is vulnerable to **CVE-2020-1472 (Zerologon)**.
3. **Victim1** and **Victim2** are Windows 10 machines that are part of the domain controlled by the **ADDC (Corporate Network)**. **Victim1** is used by the security personnel to access the video feed provided by the **Axis**. Finally, this machine has an RDP port enabled with weak credentials. Inbound traffic from the **External Network** is not allowed for **Victim1** and **Victim2**.

4. **Attacker's machine** (Figure 6) – a machine that the attacker uses for Initial Access and initial Lateral Movement. Initially, this machine can only access **Axis**, as **Victim1**, **Victim2** and **ADDC** rely on Windows Firewall to restrict connections to the **External Network**.
5. **C&C Server** (Figure 6) – another attacker-controlled machine in the **External Network**. This machine is used as a Command & Control server for R4IoT executables deployed at **Victim1** and **Victim2**.
6. **WAGO PLC1, WAGO PLC2, WAGO PLC3** and **NEC IP Phone** – several OT/IoT devices within the **Operational Network(s)**. These devices are affected by the **NUCLEUS:13 vulnerabilities** (found within **Project Memoria**).

7.2 Attack Details

Figure 6 illustrates the various steps the attacker takes to execute R4IoT, which are detailed in Sections 7.2.1 to 7.2.3.

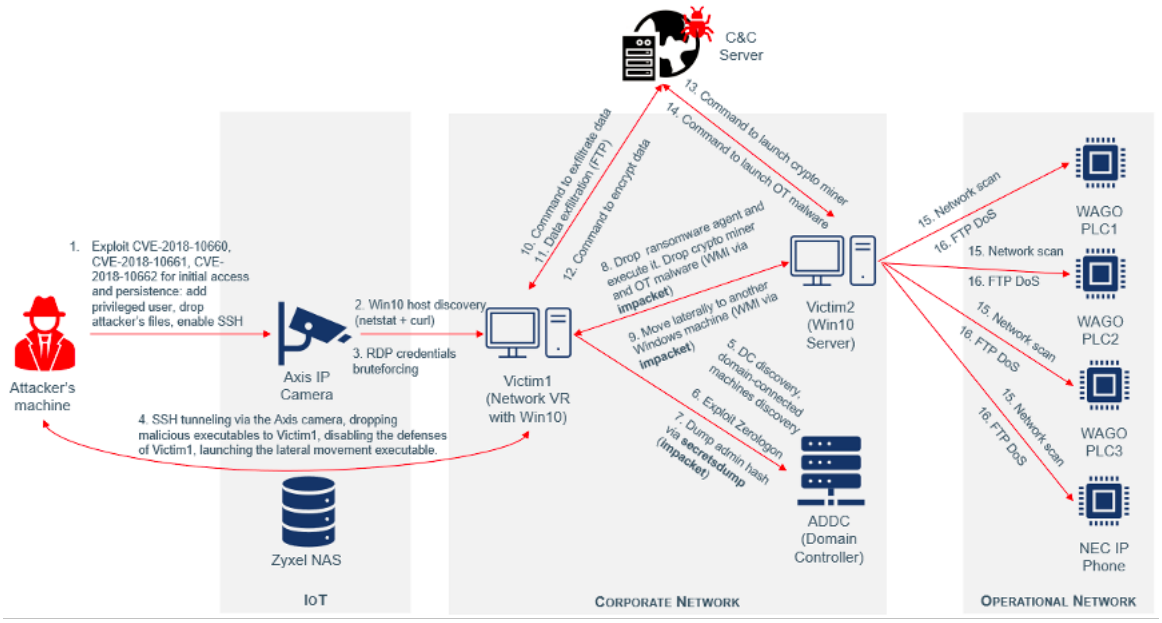

Figure 6 – Attack Overview

7.2.1 Initial Access

Attacker uses the **Axis** M2025-LE camera as the entry point into the **Corporate Network**. Initially, the access to the web interface of the camera (which also contains administrative settings) is password protected. In the past, we have seen wide usage of known **default credentials** for gaining access to internet-facing web cameras. In our scenario, we assume that the default password has been changed and the new password is unknown to **Attacker**.

However, the **Axis** camera in our lab is affected by critical **vulnerabilities**. **Attacker** achieves remote command execution and takes over the camera by exploiting the following vulnerabilities:

- **CVE-2018-10661**: Authorization bypass vulnerability. Anyone can send unauthenticated HTTP requests that reach **.srv** files of the Apache Tomcat webserver running on the camera. Such requests are, in turn, forwarded to the **/bin/ssid** process that runs with **root** privileges.

- CVE-2018-10662: Attackers can access the interface that allows unrestricted **dbus** messages. This interface is reachable from **/bin/ssid**'s **.srv** files.
- CVE-2018-10660: Shell command injection vulnerability into one of the service interfaces of **dbus**.

Attacker performs the following actions, which are fully automated, on the **Axis** camera:

- Originally, the root '/' directory is mounted in the read-only mode (RO). This limits the amount of non-volatile disk space available to **Attacker** to only a few megabytes. Therefore, '/' is re-mounted in the read-and-write (RW) mode, allowing uploads of large files and keeping them on the disk.
- Start a local web server (**Attacker**'s machine) to upload files to the camera. These files include the busybox utility, and **Attacker**-developed scripts and binaries.
- By default, SSH connections to **Axis** are disabled. Therefore, **Attacker** enables **sshd** and creates a new user with root privileges (so that if something goes wrong, **Attacker** may still retain control over the camera).
- Find active network connections from hosts of the **Corporate Network** to **Axis** (using ***netstat***). **Attacker** assumes that there will be a Windows machine connected to the camera to monitor the video feed. **Netstat** is a host-based utility that shows active connections to the host where it runs without firing "noisy" network scans.
- If a connected Windows machine is found, scan it for the Windows RDP service via a single HTTP request with curl to port 3389. If the port is open, it is assumed that the RDP service is available.
- Obtain valid RDP credentials using a dictionary attack against accounts with high privileges (a custom tool developed by **Attacker** is used) [1].
- If successful, create an SSH tunnel between the attacker machine and the RDP machine (Victim1, as per Figure 6), making the camera act as a proxy server.
- Mount a folder from the attacker's machine to the RDP machine (Victim1) for dropping the R4IoT executables and auxiliary files.

Once having RDP access to **Victim1**, **Attacker** manually disables the Windows Firewall and any antivirus or defense software. Then, they copy the R4IoT executables and auxiliary files to **Victim1** and run the lateral movement executable. Figure 7 shows a screenshot from the **Attacker**'s machine: the Initial Access executable has been successfully executed and the **Attacker** can start deploying R4IoT.

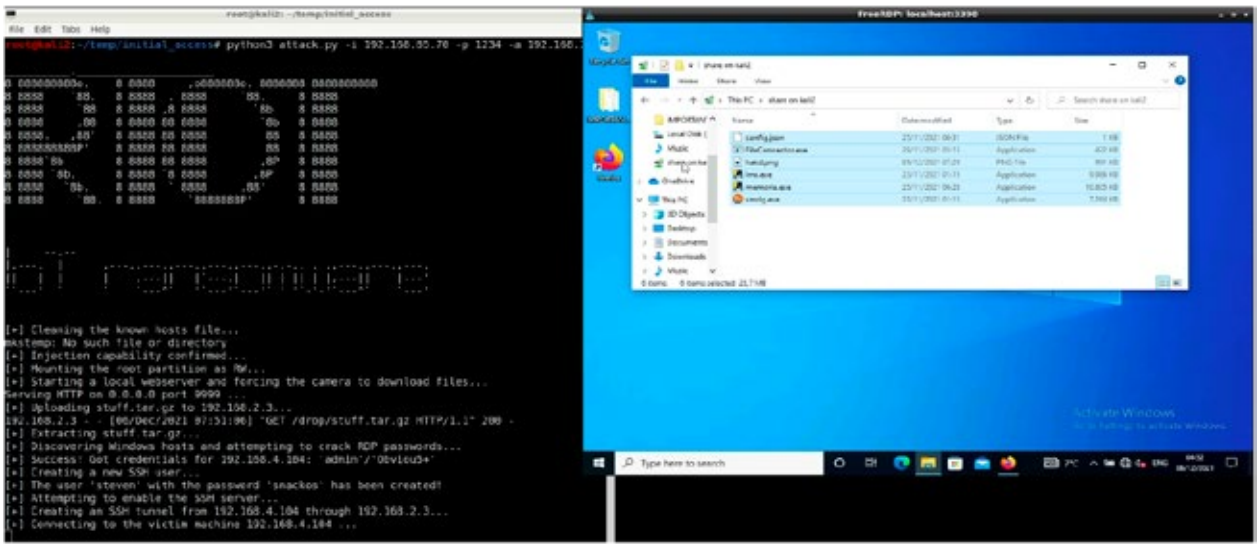

Figure 7 – Initial Access from Axis to Victim1

[1] We want to land on a user account that allows us to disable endpoint security tools on the Victim1 machine and to initiate connections towards a domain controller. We do not try to detect user privileges automatically because the **Attacker** will interact the Victim1 machine and deduce that.

To further illustrate how IoT devices can be used for initial access in ransomware operations, we have explored an alternative scenario, in which, instead of the **Axis** camera, the attacker finds another IoT device exposed to a public network.

We used a **Zyxel** network-attached storage (NAS) device (Zyxel NAS326) that runs embedded Linux on an ARM processor. The device is affected by **CVE-2020-9054**, a pre-authentication command injection vulnerability, which may allow a remote, unauthenticated attacker to execute arbitrary code on the device.

```
/usr/local/apache/web_framework/bin $ ls -lA
-rwsr-xr-x    1 root     root          3696 Sep  4  2019 executer_su
-rwxr-xr-x    1 root     root           542 Sep  4  2019 run_model
/usr/local/apache/web_framework/bin $
```

```
1  signed int __fastcall main_0(signed int a1, _DWORD *a2)
2  {
3    signed int v2; // r3
4    FILE *v3; // r4
5    int v4; // r0
6    _DWORD *v6; // [sp+0h] [bp-1Ch]
7    signed int v7; // [sp+4h] [bp-18h]
8    char **argv; // [sp+8h] [bp-14h]
9    signed int i; // [sp+Ch] [bp-10h]
10
11   v7 = a1;
12   v6 = a2;
13   if ( a1 > 1 )
14   {
15     close(2);
16     dup(1);
17     argv = (char **)malloc(4 * v7);
18     if ( argv )
19     {
20       for ( i = 1; i < v7; ++i )
21         argv[i + 0x3FFFFFFF] = strdup((const char *)v6[i]);
22       argv[i + 0x3FFFFFFF] = 0;
23       setuid(0);
24       v3 = (FILE *)stderr;
25       v4 = execv((const char *)v6[1], argv);
26       fprintf(v3, "%d\n", v4);
27       v2 = 0;
28     }
29     else
30     {
31       fwrite("insufficient memory.\n", 1u, 0x15u, (FILE *)stderr);
32       v2 = -1;
33     }
34   }
35   else
36   {
37     fprintf((FILE *)stderr, "usage: %s command [arg1 arg2 arg3 ...]\n", *a2);
38     v2 = -1;
39   }
40   return v2;
41 }
```

Figure 8 – Pseudocode of the "executer_su" Main Function

We have discovered that remote commands launched via CVE-2020-9054 are executed with the same privileges as the webserver's user "nobody". It is a special Linux user that does not own any files and has no special privileges. However, the vulnerability description mentions the "setuid utility", which can be used for privilege escalation. By analyzing the firmware files, we found the "executor_su" binary that is common to various IoT devices and is commonly used for privilege escalation. (We have seen some evidence suggesting that Mukashi, the newer version of the **Mirai malware**, used the same technique for weaponizing this vulnerability.)

Figure 8 illustrates that the "executer_su" binary is owned by the root user. It also shows that the binary is just a wrapper around the "**execv**()" call that executes any command, and it also contains a call "**setuid**(0)", which sets the effective user ID of the calling process to the owner of the binary (root).

Considering the above, we have modified the original exploit for CVE-2020-9054 to execute remote commands through the "executer_su" binary, but the rest of the attack is performed exactly as with the **Axis** camera.

7.2.2 Lateral Movement

The R4IoT lateral movement executable will identify **Domain Controllers** (DCs) in the network and attack them with an exploit for *CVE-2020-1472 (Zerologon)*. After attacking a vulnerable DC, it will dump the **LSA hashes** from the compromised DC and the account names of machines subscribed to it. Next, it will resolve these names to IP addresses, search for the Administrator account's password hash and use it to disable Windows Firewall and Windows Defender in every domain-subscribed host through **WMI**.

7.2.2.1 Lateral movement via WMI

Understanding WMI is crucial to get insights about the inner workings of R4IoT. WMI stands for "Windows Management Instrumentation" and is used as the infrastructure to manage data and operations on Windows-based operating systems. It is heavily used for administrative tasks and is designed for local and remote management. It exposes manageable entities through Common Information Model **(CIM) classes** and their providers. Windows exposes a set of core CIM classes that can be used out of the box to manage the system. (PowerShell is one scripting environment where they can be used.) However, threat actors are also known to **use it** heavily for infiltration into Windows networks and systems. R4IoT is no different in this aspect, as it relies on the same techniques.

7.2.2.2 Discovery of Domain Controllers, Zerologon, pass the hash

Once the R4IoT lateral movement executable is executed on **Victim1**, it grabs all the instances of the class "*Win32_NTDomain*", which represents a Windows domain, and extracts the following fields:

- ***DomainControllerName***: Computer name for the discovered domain controller (example: "*WIN-8DS4VJS9R7A*")
- ***DomainControllerAddress***: IP address of the discovered Domain Controller (example: "*192.168.4.102*")
- **DomainName**: Name of the domain (example: "*VICTIMSNET*")
- **DnsForestName**: Name of the root of the DNS tree (example: "*victimsnet.hack*")

Since **Victim1** is part of a domain, this machine will have at least two instances of the "*Win32_NTDomain*" class, and one of them will contain this set of fields.

R4IoT is designed to attack more than one Domain Controller. In our lab environment, it will attack the only DC we have with *CVE-2020-1472 (Zerologon)*. After that attack, the target DC will have a null password associated with the DC machine account "*WIN-8DS4VJS9R7A*", allowing the Attacker to login into it with a null password and dump the LSA secrets.

We relied on the "*secretsdump.py*" **script** of **impacket** to dump LSA hashes that eventually contain the **NTLM hash** of the domain administrator, as well as hashes for the machine accounts. A typical output of this script is shown on Figure 9.

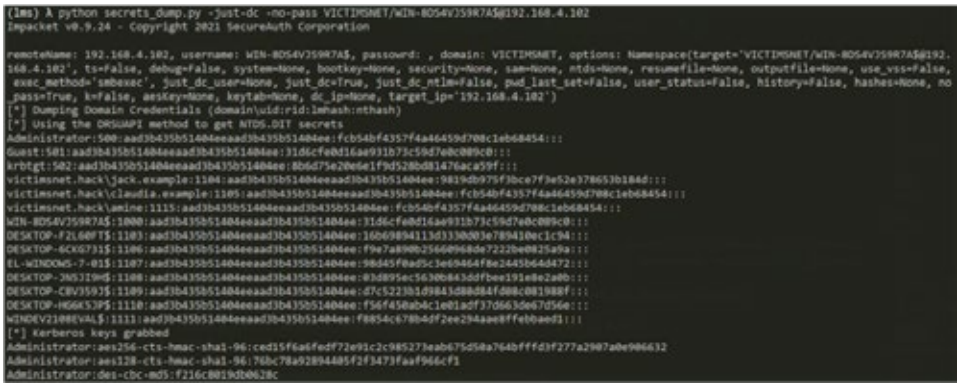

Figure 9 – Output Example of "secretsdump.py"

The NTLM hash format is composed of the following: (1) an account name string; (2) a relative ID number; and (3) a concatenation of the NT and LM hashes. It follows the format '***ACCOUNT_NAME:RELATIVE_ID:NT_HASH:LM_HASH:::***'. In our case, for example, the domain administrator's account has the following hash:

'Administrator:500:aad3b435b51404eeaad3b435b51404ee:fcb54bf4357f4a46459d708c1eb68454:::'

In addition to dumping NTLM hashes, "secretsdump.py" extracts Kerberos keys from the DC. Such keys are handy because they contain the machine names, which are part of the domain forest governed by the compromised DC. This allows use of the DC server to retrieve the IP addresses of these machines with DNS queries.

7.2.2.3 More on lateral movement, dropping R4IoT executables

The R4IoT lateral movement executable maps the IP addresses of machines to their machine names within the compromised domain. It uses the NTLM hash of the administrator's account and the WMI functionality implemented within **impacket** to connect to each of these machines. Once connected, the executable disables Windows firewall and Windows Defender using the "Set-MpPreference" and "Set-NetFirewallProfile" commands. Finally, it drops other R4IoT executables and auxiliary files with the SMB request "SMB_COM_WRITE_ANDX" and executes the **C&C Agent** executable through the WMI CIM class instances "Win32_Process" and "Win32_ProcessStartup".

7.2.3 Impact

Apart from the lateral movement executable, R4IoT includes the following components:

- **C&C Agent** executable that reports back to **C&C Server** and runs local commands based on instructions received from **C&C Server**. This executable is automatically started on every Windows machine that the lateral movement executable can reach.
- **Cryptominer** executable – a client for mining a cryptocurrency. Its purpose is to hijack the computational resources of the victim machine and use them in favor of **Attacker**.
- **Memoria** executable that will launch DoS attacks against critical IoT/OT assets.

7.2.3.1 C&C Server/Agent

We rely on a modified version of the Racketeer toolkit [2] to provide **C&C Server/Agent** functionalities. Upon receiving a command from **C&C Server**, **C&C Agent** can encrypt/decrypt files on the infected machine, exfiltrate files and launch arbitrary executables with administrative privileges.

Figure 10 – Racketeer C&C Server

Figure 10 shows that after the lateral movement executable is done, there are two victim machines that report back to **C&C Server**. For example, by using the "heartbeat" command, **Attacker** can retrieve the name of the compromised machine, the process ID of **C&C Agent** running on that machine and the name of the user with whose privileges **C&C Agent** was started.

First, **Attacker** may choose to exfiltrate data from a victim machine. For example, Figure 10 shows that launching the "attack exfil" command against one of the **C&C Agents** will enumerate text files on the victim and send them to **C&C Server**. There might be sensitive information [3] of interest to **Attacker**.

[2] We would like to thank Dimitry Snezhkov for providing this useful toolkit to the community. To learn more, please watch this DEFCON presentation at https://www.youtube.com/watch?v=VJ8aqReB118 or visit the Github page of the tool at https://github.com/dsnezhkov/racketeer.

[3] For our exercise, we only retrieve text files and send them to an FTP server hosted on **C&C Server**. This functionality can be further extended to perform targeted searches for data of interest, as well as to decrease the detectability of data exfiltration.

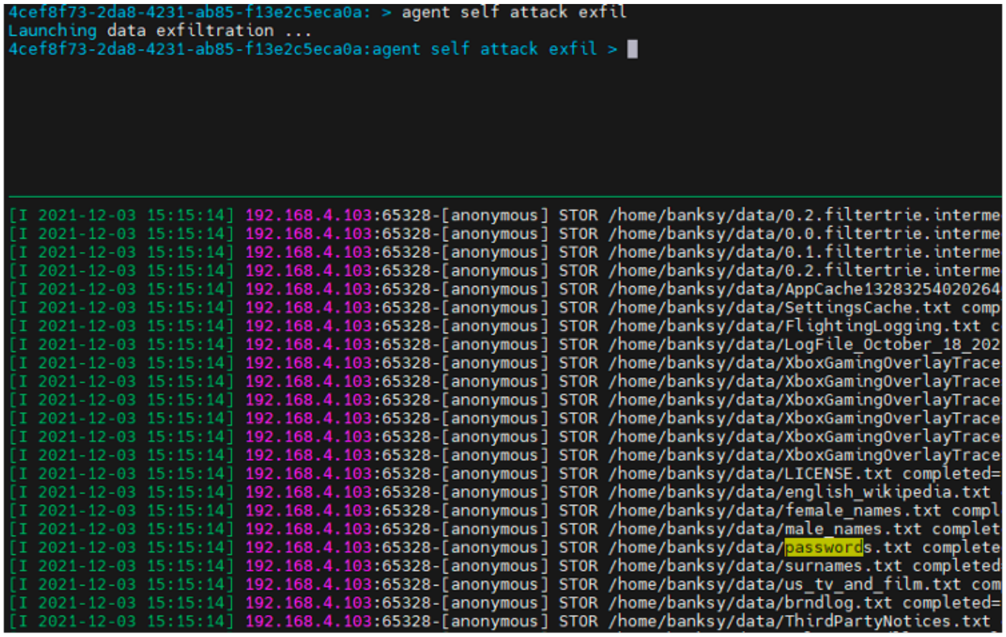

Figure 11 – Data Exfiltration

7.2.3.2 Encryption

After the sensitive information has been retrieved, **Attacker** will proceed with encrypting sensitive files and posting a ransom notice. Figure 12 shows some of the sensitive files being encrypted, and Figure 13 shows the ransom notice posted. Upon receiving a corresponding command, **C&C Agent** will create hundreds of text files on a desktop of every user of the machine. These text files will contain the same ransom notice with demands of payment.

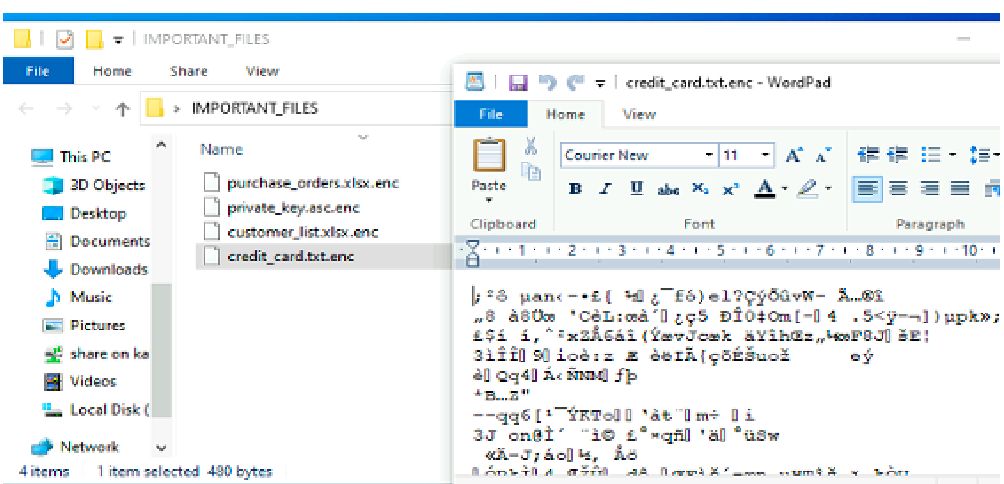

Figure 12 - Encrypting Sensitive Files

Figure 13 - Ransom Notice

7.2.3.3 Cryptocurrency mining

Upon receiving a command, the **C&C Agent** will launch an executable for mining the **Monero** cryptocurrency. We use a pre-configured off-the-shelf client called **XMRig** that, when started, will attempt to connect to a mining pool and perform mining operations. For example, Figure 14 shows a small portion of the traffic generated by **XMRig**.

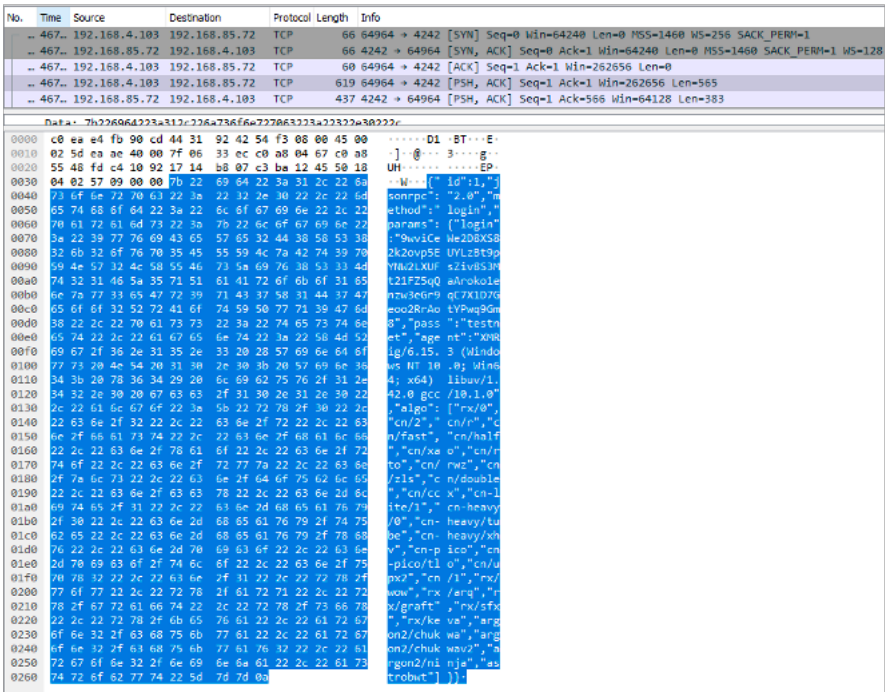

Figure 14 - Monero Traffic

7.2.3.4 IoT/OT impact

The **Memoria** executable can be invoked from **C&C Server** via **C&C Agent**. The executable will launch a custom network scanner [4] to identify critical IoT/OT assets in the network that may contain critical vulnerabilities [5]. After such assets are located, **Memoria** will launch a Denial-of-Service attack against these assets (an exploit for CVE-2021-31886). After the attack, the vulnerable devices will go offline. In addition, any physical process controlled by some of the affected devices (WAGO PLCs) will be interrupted. Figure 15 shows the physical effect of the attack against one of such devices in our lab, visible to **Attacker**: the WAGO PLC on the left crashes so that the HVAC system on the right stops functioning immediately, so the fan stops and the lights go off.

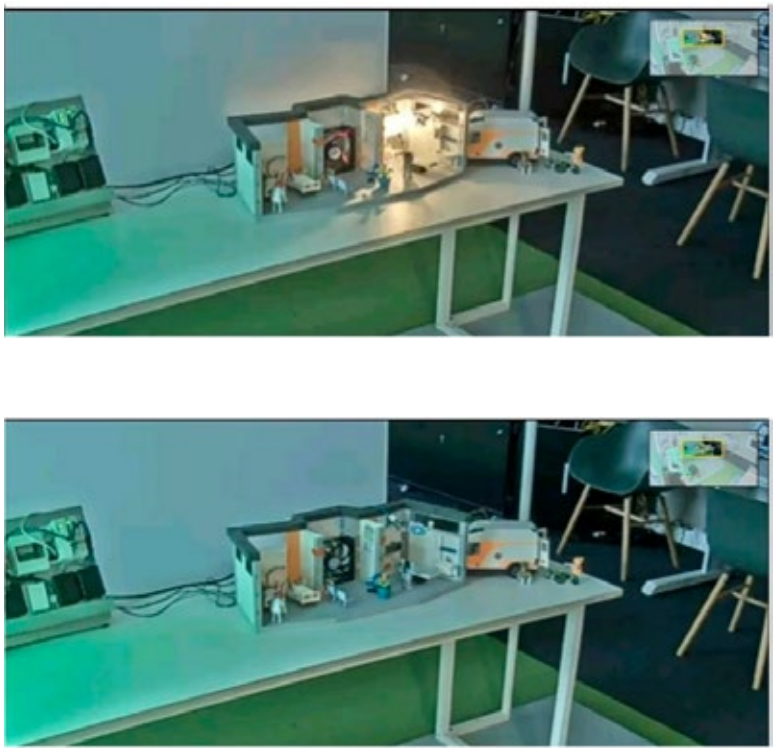

Figure 15 – The Physical Effect of an Exploit of CVE-2021-31886

[4] The network scanner is based on: **https://github.com/Forescout/project-memoria-detector**

[5] Our lab has several devices affected by Nucleus:13 (see **https://www.forescout.com/research-labs/nucleus-13/**).

7.3 A Summary of R4IoT TTPs

Table 1 shows a summary of the attacker tactics, techniques and procedures that are part of R4IoT.

STEP	TACTIC	TECHNIQUE	PROCEDURE
1	Initial Access	Exploit public-facing application	Exploit CVE-2018-10660, CVE-2018-10661, CVE-2018-10662 for the IP camera Exploit CVE-2020-9054 for the NAS
2	Persistence	Create Account	Useradd
3	Discovery	Remote system discovery	Netstat + Curl
4	Credential Access	Brute Force: Pasword Guessing	Bespoke cracker
5	Lateral Movement	Remote Services: Remote Desktop Protocol	RDP with valid account via freerdp and SSH tunneling
6	Defense Evasion	Impair Defenses: Disable or Modify Tools	Manually disable Windows Defender
7	Defense Evasion	Impair Defenses: Disable or Modify System Firewall	Manually disable Windows Firewall
8	Command and Control	Ingress Tool Transfer	Windows share (SMB through RDP)
9	Discovery	Remote system discovery	Win32_NTDomain
10	Lateral Movement	Exploitation of remote services	ZeroLogon (CVE-2020-1472)
11	Credential Access	OS Credential Dumping: NTDS	Secretsdump.py
12	Command and Control	Ingress Tool Transfer	SMB_COM_WRITE_ANDX (SMB)
13	Execution	WMI	Win32_Process
14	Command and Control	Application Layer Protocol	HTTPS
15	Collection	Data from Local System	File system read
16	Exfiltration	Exfiltration over C2 Channel	FTP
17	Impact	Data Encrypted for Impact	Racketeer
18	Impact	Resource Hijacking	XMRig
19	Discovery	Network Service Scanning	Project-memoria-detector
20	Impact	Endpoint Denial of Service	Exploit CVE-2021-31886

Table 1 – A Summary of R4IoT TTPs

8 Stopping the Threat: a Playbook for Risk Management

As mentioned in the Introduction, successful risk management for ransomware (both for current and future threats) is paramount. In this Section, we examine how the NIST Cybersecurity Framework and a Zero Trust Architecture help to protect organizations against ransomware, using R4IoT as an example.

8.1 Risk Management with the NIST Cybersecurity Framework

The **NIST Cybersecurity Framework** serves as the basis for risk management in several organizations, especially in critical infrastructure sectors. The framework has five key functions – Identify, Protect, Detect, Respond and Recover – which encompass the whole lifecycle of security management.

There are three important observations from our study of the ransomware threat landscape that make mitigation of this threat possible across the NIST functions.

- **Identification and Protection** are possible because there are hundreds of very similar attacks happening simultaneously currently. For instance, Conti was one of the most successful ransomware gangs in 2021 with more than **400 successful attacks on U.S. and international organizations**. That means it is possible to identify devices and vulnerabilities being **actively exploited** so their protection can be prioritized.
- **Detection** is possible because most tools and techniques these actors use are well-known. We already presented the top TTPs in Section 4.3.
- **Response and Recovery** are possible because attacks are not immediate and fully automated. The **average dwell time** of ransomware attackers was five days in 2021. For instance, there are **several detailed reports** of Conti incidents available online that detail and timestamp the steps taken by attackers over these days.

Table 2 uses the R4IoT TTPs (described in Section 7.3) and the NIST Cybersecurity Framework key functions to present mitigation steps for complex ransomware threats. We focus on network-based mitigation, so we have removed steps that depend exclusively on endpoint behavior (2, 9, 11, 15, 17) or that represent legitimate network behavior (8, 12).

#	TACTIC AND TECHNIQUE	PROCEDURE	IDENTIFY	PROTECT	DETECT	RESPOND
1	Initial Access – Exploit public-facing application	CVE-2018-10660 CVE-2018-10661 CVE-2018-10662 CVE-2020-9054	Identify vulnerable devices Monitor inbound and outbound traffic from/ to vulnerable devices	Patch vulnerable devices Segment the network to prevent external access	Detect command injections via protocols such as HTTP Detect breaches of segmentation policies	Temporarily quarantine device in VLAN or disconnect it from the network
3	Discovery – Remote system discovery	Netstat + Curl	Monitor inbound and outbound traffic from/ to vulnerable devices	Allow only the minimum necessary traffic (e.g., no HTTP to RDP ports)	Detect deviations of network communications baseline	Temporarily quarantine device in VLAN or disconnect it from the network
4	Credential Access – Brute Force: Password Guessing	Bespoke Cracker	Identify hosts with weak credentials	Implement policies for password strength and expiration Implement Multi-Factor Authentication Segment the network to prevent communication between IoT and IT devices	Detect deviations of network communications baseline Detect RDP brute forcing Detect breaches of segmentation policies	Temporarily quarantine device in VLAN or disconnect it from the network

Table 2 – Mitigations for R4IoT TTPs

#	TACTIC AND TECHNIQUE	PROCEDURE	IDENTIFY	PROTECT	DETECT	RESPOND
5	Lateral Movement – Remote Services: Remote Desktop Protocol	RDP with Valid Account via Freerdp and SSH Tunneling	Identify potential targets (hosts with RDP enabled) with service and asset inventory	Restrict RDP connections only from trusted sources either via targeted rules or segmentation policies	Detect deviations of network communications baseline Detect breaches of segmentation policies	Temporarily quarantine device in VLAN or disconnect it from the network
6	Defense Evasion – Impair Defenses: Disable or Modify Tools	Manually Disable Windows Defender	Identify security tools running on hosts	Enforce compliance policy: AV should be always turned on and updated	Detect change of AV state to disabled	Enable AV
7	Defense Evasion – Impair Defenses: Disable or Modify System Firewall	Manually Disable Windows Firewall	Identify security tools running on hosts	Enforce compliance policy: firewall should be always turned on	Detect change of firewall state to disabled	Enable firewall
10	Lateral Movement – Exploitation of Remote Services	ZeroLogon (CVE-2020-1472)	Identify vulnerable servers	Patch vulnerable servers Enforce update policy	Detect Zerologon exploitation attempts	Temporarily quarantine server in VLAN or disconnect it from the network
13	Execution – WMI	Win32_Process	Identify potential targets (hosts with WMI enabled) with service and asset inventory	Restrict WMI connections only from trusted sources either via targeted rules or segmentation policies	Detect cleartext WMI and blacklisted PowerShell commands	Temporarily quarantine device in VLAN or disconnect it from the network
14	Command and Control – Application Layer Protocol	HTTPS	Monitor HTTPS connections	Keep an up-to-date list of known C&C hosts and malicious JA3 hashes	Detect blacklisted C&C hosts and HTTPS connections matching malicious JA3	Temporarily quarantine device in VLAN or disconnect it from the network
16	Exfiltration – Exfiltration over C2 Channel	FTP	Monitor FTP sessions	Keep an up-to-date list of known C&C hosts Disable FTP traffic when not needed	Detect blacklisted C&C hosts Detect FTP traffic	Temporarily quarantine device in VLAN or disconnect it from the network
18	Impact – Resource Hijacking	XMRig	Integrate with EDR solution to identify running processes	Keep an up-to-date list of known malicious processes	Detect known malicious processes running on endpoint Detect network traffic related to cryptocurrency mining	Kill malicious process
19	Discovery – Network Service Scanning	Project-memoria-detector	Monitor inbound and outbound traffic from/to vulnerable devices	Segment the network to prevent communication between IT and OT/IoT devices	Detect network scanning event Detect breaches of segmentation policies	Temporarily quarantine device in VLAN or disconnect it from the network
20	Impact – Endpoint Denial of Service	CVE-2021-31886	Identify vulnerable devices	Patch devices Segment the network to isolate vulnerable critical devices	Detect exploitation attempts (buffer overflows) Detect breaches of segmentation policies	Temporarily quarantine device in VLAN or disconnect it from the network

Table 2 continued – Mitigations for R4IoT TTPs

8.2 Implementing Policies with a Zero Trust Architecture

One way of efficiently implementing many of the mitigation steps presented in Table 2 is to use a Zero Trust Architecture.

Forrester coined the term **Zero Trust in 2010**, which became a NIST standard in 2020 with **NIST 800-207** "Zero Trust Architecture" (ZTA). Zero trust is the modern replacement for perimeter-based security. In perimeter-based security, the overall idea was that whatever is outside the network is potentially malicious, whatever is inside is probably benign and a demilitarized zone (DMZ) keeps the two worlds apart. The idea behind zero trust is radically opposite: never implicitly trust any device or communication, even if it is part of the internal network. In ZTA, each device, application and user should have their own perimeter so very fine-grained access control policies can be implemented and enforced.

There are three key pillars to implementing Zero Trust.

- **Visibility** is foundational to resource defense since "you can't combat a threat you can't see or understand." Often, device security comes first in practical discussions of technical controls, but visibility must extend beyond devices to network communications. That is where controls may detect anomalous behavior.
- **Compliance** establishes what should or should not be trusted in the network, making it possible to act on devices that do not meet a minimum set of compliance requirements.
- **Segmentation** is a fundamental control that allows enforcing Zero Trust by limiting the allowed network communications of devices. Zero Trust implemented based on network visibility, compliance rules and via appropriate network segmentation policies can help stop the spread of ransomware by limiting attack surfaces. In this way, only devices that need to be internet-facing will be accessible, and any lateral movement in the network becomes more difficult since devices can only communicate to other devices they should.

Below, we show how we implemented a Zero Trust architecture to stop R4IoT in our lab. Our implementation was done by leveraging **Forescout Products**, but the guidelines we discuss below can be generalized.

The strategy was based on two ideas: enforce a restrictive segmentation policy by default with least privilege rules and quarantine non-compliant devices. The general segmentation rule is to prevent any host from any segment to reach any other host from any other segment. To allow the lab network to function, we have added the following exceptions to the Zero Trust policy:

- Only DHCP and DNS traffic can flow between segments
- Only a few trusted hosts can reach IP cameras
- Only a few trusted hosts can reach OT devices
- IT devices should be able to reach the ADDC server
- Only outbound traffic is allowed from IT devices to the external networks

Even if a device is in a trusted segment or group, it will get banned from the network the moment it is not compliant and will not be allowed back until there is a proof of its compliance.

Table 3 illustrates the segments and the segmentation policies we have in place. Slate cells represent allowed communications, whereas orange cells represent unallowed communications. We omitted potential granular rules for other types of devices grouping them as 'Rest' for the sake of simplicity.

Allowed Communication

Unallowed Communication

DESTINATION / SOURCE		EXTERNAL NETWORKS [6]		ENTERPRISE				ICS	MEDICAL	BAS	
		Internet	Office	Rest	AD servers	OT admin-istration	Trusted NVRs			Rest	IP cameras
EXTERNAL NETWORKS	Internet	N/A									
	Office										
ENTERPRISE	Rest										
	AD servers										
	OT admin-istration										
	Trusted NVRs										
ICS											
MEDICAL											
BAS	Rest										
	IP cameras										

Table 3 – Zero Trust Policy to Stop R4IoT

[6] No such segment exists in our setup; we use the name as an abstraction.

The network layout of the segments and groups is as follows:

SEGMENT	TYPE	IP ADDRESSES
Internet	Group	Public IPs
Office	Segment	192.168.85.0/24
Enterprise	Segment	192.168.4.0/24
ICS	Segment	192.168.1.0/24
Medical	Segment	192.168.3.0/24
BAS	Segment	192.168.2.0/24
IP Cameras	Group	Based on device fingerprints
AD Servers	Segment	192.168.4.102/32
OT Administration	Segment	192.168.4.103/32
Trusted NVRs	Segment	192.168.4.104/32

8.3 Further Resources

We focused on mitigation against R4IoT as an example in this section, but more information about risk mitigation for ransomware in general can be found on:

- **The No More Ransom Project** – an initiative by the Dutch police, Europol, Kaspersky and McAfee to help ransomware victims recover encrypted data without paying the ransom.
- **StopRansomware** – a website maintained by CISA with information, tips, FAQs, an assessment for ransomware readiness and a form to report incidents.
- **Rising Ransomware Threat To Operational Technology Assets** – a CISA fact sheet with step-by-step risk mitigation recommendations for operational technology asset owners.
- **NIST IR 8374: Cybersecurity Framework Profile for Ransomware Risk Management** – identifies the security objectives in the cybersecurity framework that help prevent, detect, respond to and recover from ransomware incidents.
- **NIST SP 1800-25: Data Integrity – Identifying and Protecting Assets Against Ransomware and Other Destructive Events**
- **NIST SP 1800-26: Data Integrity – Detecting and Responding to Ransomware and Other Destructive Events**
- **NIST SP 1800-11: Data Integrity – Recovering from Ransomware and Other Destructive Events**

9. Conclusion

Ransomware has been the most prevalent threat of the past few years, and so far, it has mostly leveraged vulnerabilities in traditional IT equipment to cripple organizations. But new connectivity trends have added a number and a diversity of OT and IoT devices that have increased risk in nearly every business.

We have discussed how attacker evolution, the growth of the Internet of Things, the IT/OT convergence and the emergence of widespread supply chain vulnerabilities point to two future trends for ransomware: IoT as an entry point and OT as the target of attacks. We have also demonstrated how we created a malware in our lab that exploits IoT, OT and IT devices for initial access, lateral movement and to achieve final objectives that go beyond the usual encryption and data exfiltration to cause physical disruption on business operations.

The most important messages of this report are that IoT and OT exploits are new tools in the attacker's arsenal but also that to mitigate this type of attack, solutions are required that allow for extensive visibility and enhanced control of all the assets in a network.