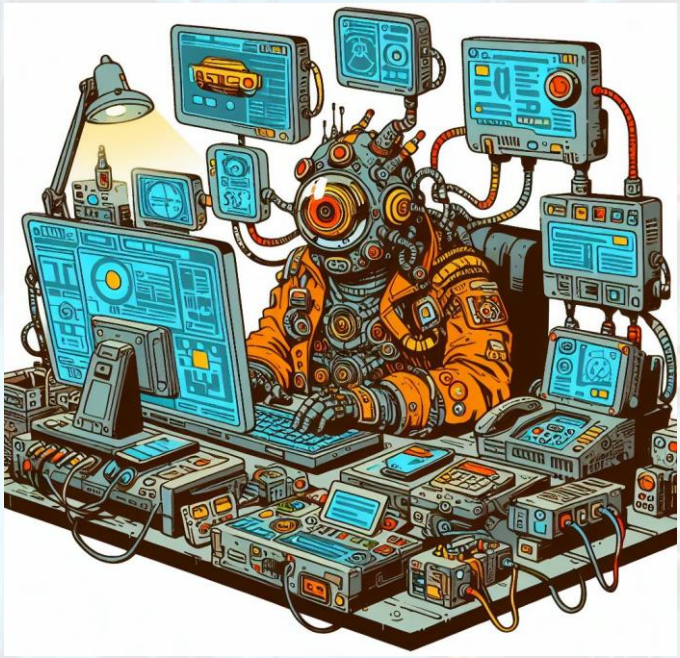

Abstract –This document presents a comprehensive analysis of the Fuxnet malware, attributed to the Blackjack hacking group, which has reportedly targeted infrastructure. The analysis delves into various aspects of the malware, including its technical specifications, impact on systems, defense mechanisms, propagation methods, targets, and the motivations behind its deployment. By examining these facets, the document aims to provide a detailed overview of Fuxnet's capabilities and its implications for cybersecurity.

The document offers a qualitative summary of the Fuxnet malware, based on the information publicly shared by the attackers and analyzed by cybersecurity experts. This analysis is invaluable for security professionals, IT specialists, and stakeholders in various industries, as it not only sheds light on the technical intricacies of a sophisticated cyber threat but also emphasizes the importance of robust cybersecurity measures in safeguarding critical infrastructure against emerging threats. Through this detailed examination, the document contributes to the broader understanding of cyber warfare tactics and enhances the preparedness of organizations to defend against similar attacks in the future.

I. Introduction

The Blackjack hacking group, purportedly linked to Ukrainian intelligence services, has claimed responsibility for a cyberattack that allegedly compromised emergency detection and response capabilities in Moscow and its surrounding areas. This group has been associated with previous cyberattacks targeting internet providers and military infrastructure. Their most recent claim involves an attack on Moscollector, a company responsible for constructing and monitoring underground water, sewage, and communications infrastructure.

The group has disseminated detailed information about this attack on the website ruexfil.com, including the use of Fuxnet malware to disrupt the Moscollector network operations center. They have published screenshots of monitoring systems, servers, and databases they assert have been erased and made inoperative and additionally password dumps.

Regarding the infection methods, the Fuxnet malware appears to have been designed to target sensor-gateways and potentially disable them, as well as to fuzz sensors, which could lead to their malfunction or destruction.

The destruction of these gateways and the fuzzing of sensors could have serious implications for the monitoring and control of various systems, potentially leading to a loss of operational visibility and control for the affected infrastructure.

The key takeaways from the analysis of the Fuxnet malware and including results of Team82 and Claroty, are as follows:

- **Unverified Claims**: Team82 and Claroty have not been able to confirm the claims made by the Blackjack group regarding the impact of their cyberattack on the government's emergency response capabilities or the extent of the damage caused by the Fuxnet malware.
- **Discrepancy in Reported Impact**: The Blackjack group initially claimed to have targeted 2,659 sensor-gateways, with about 1,700 being successfully attacked. However, Team82's analysis of the data leaked by Blackjack suggests that only a little more than 500 sensor gateways were actually impacted by the malware. The claim of having destroyed 87,000 sensors was also clarified by Blackjack, stating that they disabled the sensors by destroying the gateways and using M-Bus fuzzing, rather than physically destroying the sensors.
- **M-Bus Fuzzing**: The Blackjack group utilized a dedicated M-Bus fuzzer within the Fuxnet malware's code to fuzz the sensors. This technique was aimed at disabling the sensors, but the exact number of sensors that were "fried" or permanently damaged as a result of this fuzzing is unknown due to the network being taken down and access to the sensor-gateways being disabled.
- **Lack of Direct Evidence**: Direct evidence to confirm the extent of the damage or the impact on emergency detection and response capabilities is lacking (including targeted Moscollector).
- **Clarification from Blackjack**: Following the publication of Team82's initial analysis, the Blackjack group reached out to provide updates and clarifications, particularly challenging the contention that only around 500 sensor-gateways had been impacted. They emphasized that the JSON files made public were only a sample of the full extent of their activity.

II. Affected Industries and Potential Consequences

A. *Affected Industries:*

- **Utility Services**: The primary target of the Fuxnet malware was the utility sector, specifically the sensor gateways that manage water and sewage systems. This could have implications for the delivery and monitoring of these essential services.
- **Emergency Services**: The group claimed to have gained access to 112 emergency service number, which

could impact the ability to respond to emergencies effectively.

- **Transportation**: The group also claimed to have bricked sensors and controllers in critical infrastructure, including airports and subways, which could disrupt transportation services and safety.
- **Energy**: Gas pipelines were mentioned as another target, indicating a potential risk to energy distribution and monitoring systems.

B. Potential Consequences:

- **Disruption of Services**: The destruction or malfunction of sensor gateways could lead to a disruption of the monitoring and control systems for utilities, potentially causing service outages or failures.
- **Compromised Safety**: In transportation and energy sectors, the loss of sensor functionality could pose safety risks, as these sensors are often critical for detecting hazardous conditions.
- **Economic Impact**: The potential downtime and repair costs associated with replacing or reflashing damaged sensor gateways could have significant economic repercussions for the affected industries.
- **Emergency Response Delays**: If the claims about accessing the 112-emergency service number are accurate, this could lead to delays in emergency response, affecting public safety.
- **Data Exfiltration**: Although not explicitly mentioned in the context of Fuxnet, the malware's ability to compromise network systems could potentially lead to data breaches and the exfiltration of sensitive information.
- **Loss of Public Confidence**: Cyberattacks on critical infrastructure can lead to a loss of public confidence in the affected services and the entities responsible for their security.

III. Moscollector Attack

The attack, which began its initial compromise in June 2023, was methodically orchestrated to undermine the industrial sensors and monitoring infrastructure. Recently, the group made public their activities and the stolen information on the ruexfil website, detailing the extent and impact of their cyber offensive. The compromise of this system could potentially disrupt emergency response capabilities, affecting the safety and security of the populace.

A. Bricking of Critical Infrastructure Sensors and Controllers

Group alleges to have hacked and bricked sensors and controllers within critical infrastructure sectors, including airports, subways, and gas pipelines. This action, if true, could have disabled essential monitoring and control systems, leading to significant disruptions in public services and safety.

B. Network Appliance Disruption

The group asserts that they have disabled network appliances such as routers and firewalls. This would have a cascading effect on the network's integrity, potentially isolating various segments and hindering communication across the infrastructure.

C. Deletion of Servers and Databases

The attackers claim to have deleted servers, workstations, and databases, wiping out approximately 30 TB of data, including backup drives. This kind of data destruction could lead to a loss of historical data, disrupt ongoing operations, and complicate recovery efforts.

D. Invalidation of Moscollector Office Building Access

All keycards to the office building have reportedly been invalidated. This action could prevent employees from accessing their workplace, further hindering any attempts to assess the damage or initiate recovery protocols.

E. Password Dumping

The dumping of passwords from multiple internal services has also been claimed. This could allow unauthorized access to various systems and data, exacerbating the breach's impact and potentially leading to further exploitation.

IV. Attack's Equipment

The attack's focus was on the communication gateways that serve as critical nodes in the data transmission from the sensors to the global monitoring systems. These sensors are integral to various environmental monitoring systems, including those used in fire alarms, gas monitoring, and lighting controls.

The sensors are designed to collect physical data such as temperature and transmit this information through a serial or bus connection, specifically an RS485/Meter-Bus, to a gateway. These gateways act as transmission units, enabling the telemetry data to be sent over the internet to a centralized monitoring system, which provides operators with visibility and control over the systems.

The RS485 communication standard, as mentioned in the attack details, is a widely adopted protocol for industrial control systems due to its reliability and capability for long-distance communication. It allows for multiple devices to communicate over a single bus system, which is essential for the centralized monitoring of various sensors and controllers.

The Meter-Bus (M-Bus) is another communication protocol used for the collection and transmission of consumption data, typically for utilities like electricity, gas, water, or heat. When combined with RS485, it forms a robust network for industrial sensors to communicate and relay information to central systems.

By compromising the gateways, the attackers could potentially disrupt the telemetry and control of the sensors, leading to a loss of operational visibility and potentially causing chaos in the systems that rely on this data.

A. Leaked Information

The information from the JSON files was corroborated by two YouTube videos released by the attackers, showing the

deployment of the Fuxnet malware. The devices listed in the videos matched the gateways from the JSON file, confirming that the TMSB/MPSB gateways were the primary targets of the Fuxnet malware.

The JSON data included device types and names, IP addresses, communication ports, and location data. The types of devices listed in the JSON file were:

- MPSB (sensor gateway): 424 Devices
- TMSB (sensor gateway+modem): 93 Devices
- IBZ (3g router): 93 Devices
- Windows 10 (workstation): 9 Devices
- Windows 7 (workstation): 1 Device
- Windows XP (workstation): 1 Device

This list indicates that the attack was focused on the sensor gateways rather than the end sensors themselves. The gateways serve as the communication hubs for potentially numerous sensors connected via a serial bus such as RS485/Meter-Bus.

The leaked data from the attackers, including screenshots and JSON exports, revealed two specific types of gateways compromised during the attack:

- **MPSB Gateway**: This gateway is engineered for information exchange with external devices through multiple interfaces. It supports Ethernet and serial communication protocols, including CAN, RS-232, and RS-485. The MPSB gateway is a crucial component for integrating various sensor inputs into a cohesive monitoring system.
- **TMSB Gateway**: Similar in function to the MPSB, the TMSB gateway includes a built-in 3/4G modem, which allows it to transmit data directly over the internet to a remote system without the need for additional routing equipment.

The cyberattack targeted a critical part of the sensor ecosystem: the orchestrator/gateway devices, specifically the MPSB and TMSB gateways. These devices are essential for reading and controlling basic input/output sensors and transmitting the data to a global monitoring system for centralized oversight.

The attack exploited the communication pathways between the sensors and the global monitoring system. The typical data transmission scenarios targeted were:

- **For MPSB Gateway**: **Sensor —--- MBus/RS485 → MPSB + IoT Router —---Internet → Monitoring system.** In this scenario, the sensor data is transmitted via MBus/RS485 to the MPSB gateway, which then passes the data through an IoT router to the internet, and finally to the monitoring system.
- **For TMSB Gateway**: **Sensor —--- MBus/RS485 → TMSB (3g/4g modem) —---Internet → Monitoring system.** Here, the sensor data is sent via MBus/RS485 directly to the TMSB gateway, which uses its built-in modem to transmit the data over the internet to the monitoring system.

B. Security Lapses and Attack Methodology

The attackers exploited a significant security lapse: the use of default credentials (Username: sbk, Password: temppwd) to access the gateways via SSH. This vulnerability provided an easy entry point for the attackers to compromise the devices.

The attackers also leaked diagrams and screenshots from the sensor management UI, showcasing the network topology.

In addition to the TMSB module with built-in 3/4G capabilities, the attackers mentioned the use of iRZ RL22w routers. These routers, which use OpenWRT, were likely employed as internet-gateway devices to connect the sensors to the internet via 3G.

The attackers reportedly used the SSH service to connect to these IoT devices and tunnel to internal devices, likely after obtaining root passwords. Shodan and Censys searches revealed that thousands of iRZ routers are exposed on the internet, with around 4,100 devices directly exposing their services and about 500 enabling Telnet.

C. Sensor Management and Commissioning Software:

The software suite is a critical tool used by engineers to manage and configure sensors within an industrial or infrastructure setting. This software connects to devices using a proprietary protocol that runs over TCP port 4321. The interface allows engineers to access and modify the settings of sensors, including their input/output configurations, nodes, and readings. This capability is essential for the proper setup and maintenance of sensor networks, ensuring they operate efficiently and accurately within their designated environments.

Features of software:

- **Device Connection**: Utilizes a proprietary protocol over TCP/4321 to establish a secure connection with sensors.
- **Configuration Capabilities**: Enables the configuration of sensor settings, including adjustments to their operational parameters and the management of data they collect.
- User Interface: The interface provides a straightforward and intuitive means for engineers to interact with connected sensors, facilitating ease of use and efficiency in sensor management tasks.

D. Technical Impact

The sensor monitoring system is another significant component of the infrastructure targeted in the. This system is designed to aggregate and display telemetry and status reports from a network of sensors. It plays a vital role in operational oversight by allowing system operators to receive real-time alerts, log data, and manage sensors remotely.

According to the claims made by group, they successfully compromised this monitoring system. By doing so, they gained access to a comprehensive list of managed sensors and were able to correlate these sensors geographically on a map. This breach not only exposed sensitive operational data but also potentially

allowed the attackers to manipulate sensor outputs and disrupt normal operations. In terms of visualization and control:

- **Geolocation Features**: The monitoring system includes geolocation markings, which help in visualizing the physical locations of sensors across the network. This feature is particularly useful for large-scale operations where sensors are dispersed over extensive areas.
- **Facility-Specific Monitoring**: Screenshots from the system show that it is capable of focusing on specific facilities, such as hospitals, indicating its use in critical infrastructure settings where precise monitoring is necessary for safety and operational integrity.

V. Analyzing the Fuxnet Malware

The malware was designed to target sensor gateways, which are crucial components in the infrastructure of monitoring and control systems. The logical processes identified in the behavior of the Fuxnet malware include several steps aimed at causing irreversible damage to the targeted devices.

- The Fuxnet malware was specifically designed to target and destroy sensor gateways, not the end-sensors.
- The malware's actions included locking devices, destroying filesystems, NAND chips, and UBI volumes, and flooding communication channels.
- The attack was likely facilitated by exploiting default credentials and vulnerabilities in remote-access protocols.
- Despite claims of compromising 87,000 devices, the actual impact appears to be limited to the sensor gateways, with the end-sensors likely remaining intact.

A. Deployment Script

The attack began with the creation of a deployment script. The attackers compiled a comprehensive list of the IP addresses of the sensor gateways they intended to target, along with detailed descriptions of each sensor's physical location. The malware was then distributed to each target, likely using remote-access protocols such as SSH or the proprietary SBK sensor protocol over TCP port 4321.

B. Locking Up Devices and Destroying the Filesystem

Upon execution on the target device, the Fuxnet malware initiated a process to lock out the device. It remounted the filesystem with write access and proceeded to delete critical files and directories. It also shut down remote access services, including SSH, HTTP, telnet, and SNMP, effectively preventing any remote restoration efforts. Additionally, the malware deleted the device's routing table, crippling its communication capabilities.

C. Destroying NAND Chips

The malware's next step was to physically destroy the NAND memory chips within the devices. It performed a bit-flip operation on sections of the SSD NAND chip, repeatedly writing and rewriting memory until the chip was corrupted. NAND memory has a limited number of write cycles, and the malware exploited this limitation to cause the chips to malfunction and become inoperable.

D. Destroying UBI Volume

To prevent the sensor from rebooting, the malware rewrote the UBI volume. It used the IOCTL interface UBI_IOCVOLUP to mislead the kernel into expecting a certain number of bytes to be written, but then wrote fewer bytes, causing the device to hang indefinitely. The malware then overwrote the UBI volume with junk data, destabilizing the filesystem.

E. Denial-Of-Service on Monitoring

The final step in the malware's process was to disrupt the communication between the sensor gateways and the sensors themselves. The malware flooded the RS485/Meter-Bus serial channels with random data, overwhelming the bus and the sensors. This action prevented the sensors and gateways from transmitting and receiving data, rendering the data acquisition process useless.

F. The M-Bus Fuzzing Strategy

This strategy involved the constant sending of M-Bus frames over the serial channel, likely RS485, aiming to overwhelm and potentially damage the sensors connected to this network. The attack involved two main tactics: flooding the M-Bus channel with an excessive number of frames and employing fuzzing techniques to potentially exploit vulnerabilities within the sensors.

G. M-Bus Flooding

The attackers aimed to disable sensor communication by overwhelming the M-Bus channel with a high volume of frames. This tactic was likely intended to either directly damage the sensors through overload or to create conditions conducive to exploiting vulnerabilities. The fuzzing approach was more nuanced and targeted. The group implemented two fuzzing strategies within their malware:

- **Random Fuzzing**: This method involved generating random bytes and sending them over the M-Bus, appending a simple M-Bus CRC to ensure the frames were not dropped by the sensors. The goal was to cover the entire range of possible M-Bus payloads, valid or not, in hopes of triggering sensor malfunctions or vulnerabilities.
- **Structured Fuzzing**: this approach attempted to generate valid M-Bus frames, only randomizing specific fields within the protocol. By adhering more closely to the M-Bus structure, the malware increased the likelihood of the sensor treating the packet as valid and parsing it fully, thereby increasing the chances of triggering a vulnerability.