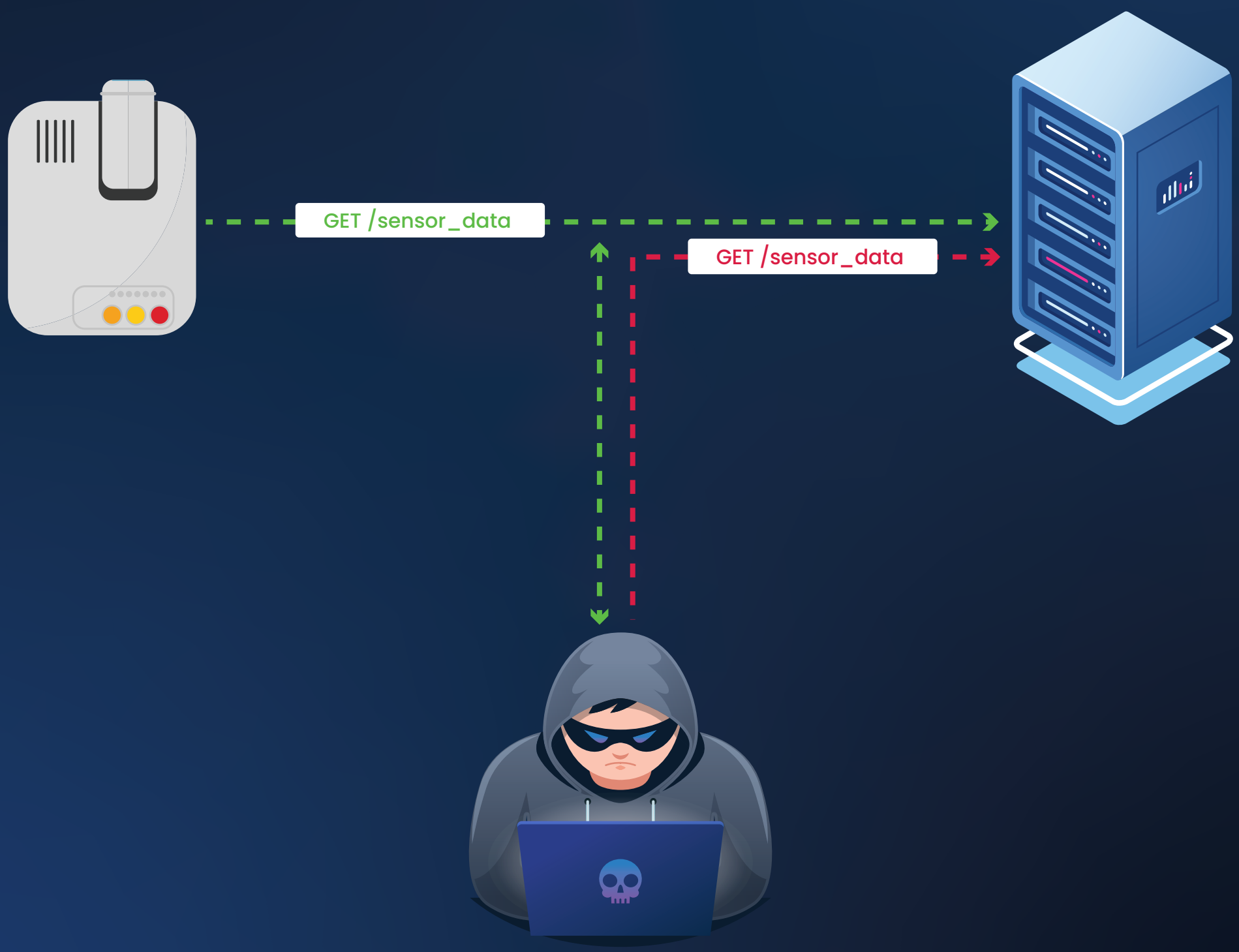
SECURITYBOAT
Frontline Of Your Business

COAP RELAY ATTACK

HANDBOOK



www.securityboat.net

SECURITYBOAT
Frontline Of Your Business

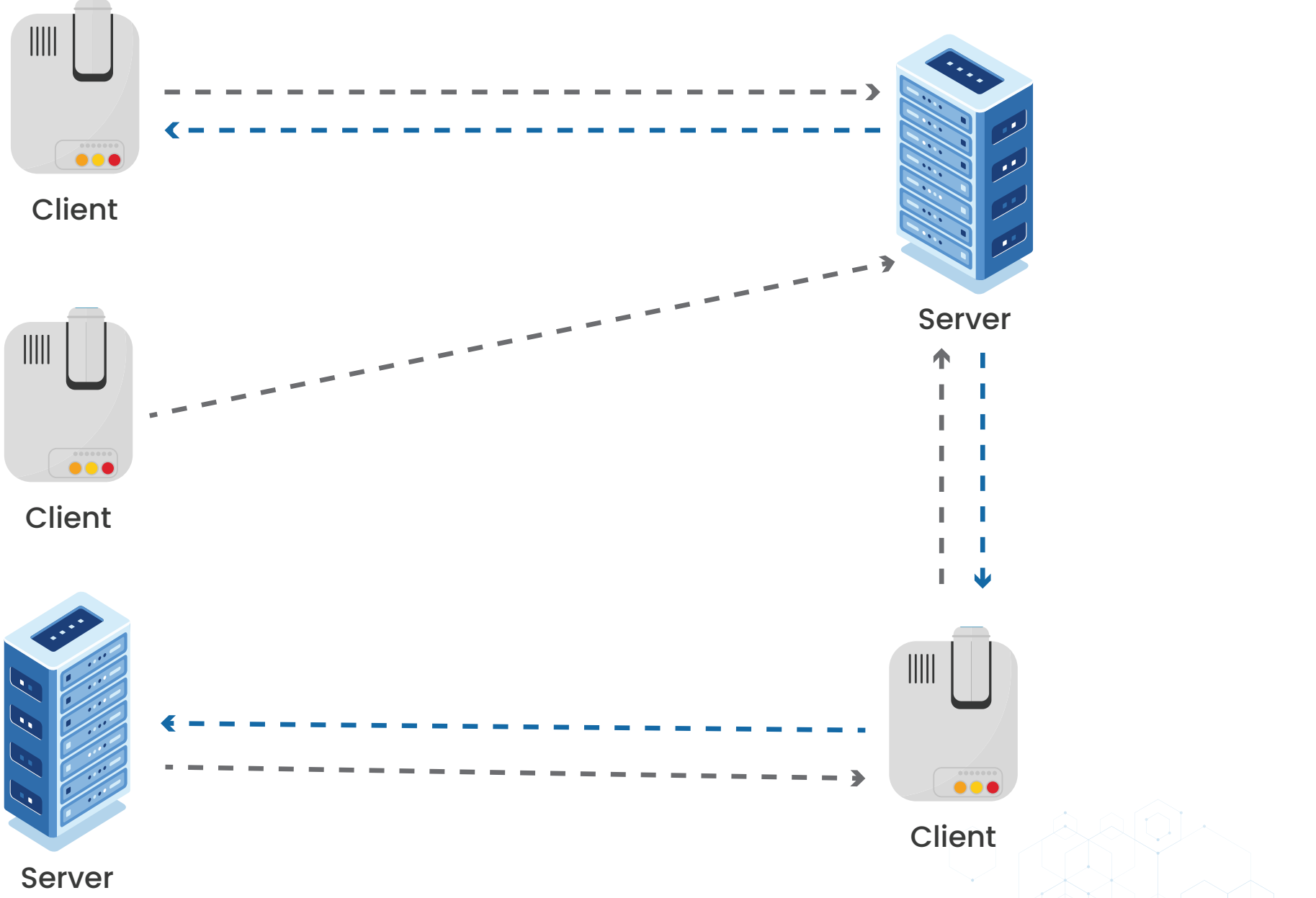
Table of Contents

1. What is CoAP?

CoAP, or Constrained Application Protocol, is a specialized communication protocol designed for IoT devices. It's like a simplified version of the web's HTTP protocol but tailored for devices that have limited processing power and operate on low-bandwidth networks. This makes it ideal for IoT scenarios where efficiency and reliability are key.

Unlike traditional web protocols, CoAP uses a lightweight approach over UDP instead of TCP, which helps conserve resources while maintaining reliable communication. It allows IoT devices to easily request and exchange data with other devices or servers using simple commands similar to those used in web browsing.

CoAP's design also supports the concept of observing resources, which means devices can receive automatic updates when data they are interested in changes. This makes it highly suitable for applications where real-time data updates are critical, such as in smart homes, industrial automation, and environmental monitoring systems. Overall, CoAP plays a crucial role in enabling seamless communication and interaction between IoT devices in a wide range of applications.



2. What is CoAP Relay Attacks?

A CoAP replay attack occurs when someone intercepts and later resends messages exchanged between IoT devices or between a device and a server. The attacker captures these messages as they travel through the network and duplicates them later. This enables the attacker to mimic a legitimate participant in the communication process, potentially gaining unauthorized access or disrupting normal device operations.

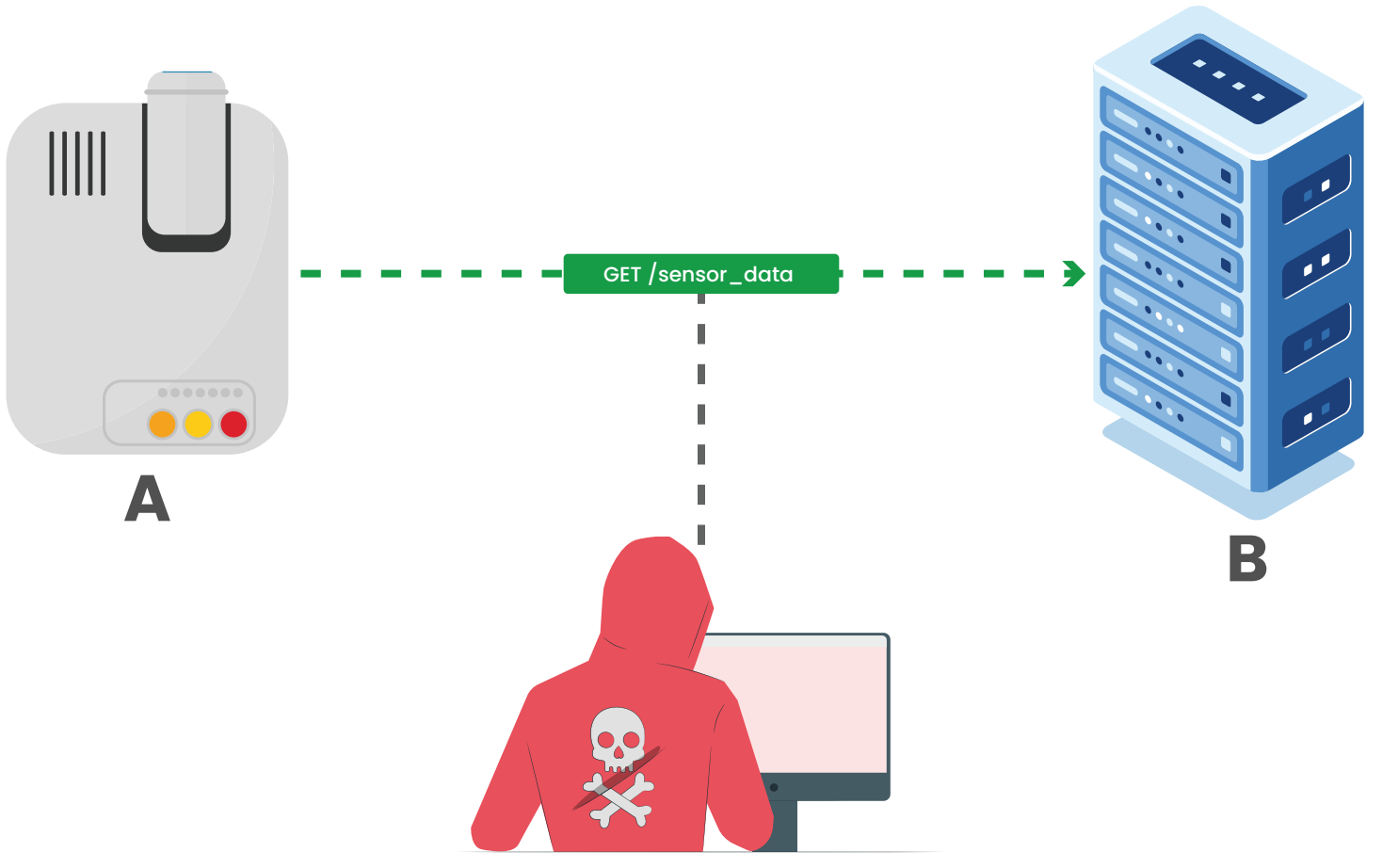
In straightforward terms, it's like copying a conversation and playing it back later to trick someone into thinking the original message is being sent again. Without specific protections in place, CoAP messages can be vulnerable to this type of interception and reuse, posing risks to the security and reliability of IoT networks where such messages are essential for controlling devices and exchanging data.

2.1 Stages of CoAP Replay Attack

Here is how the workflow of replay attack looks

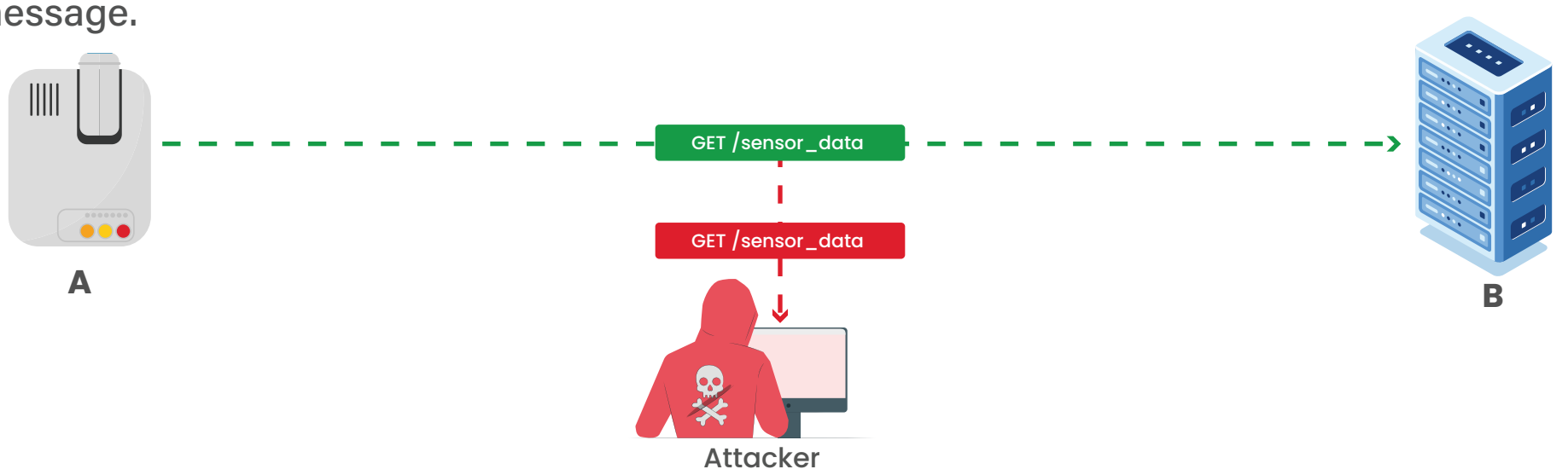
2.1.1 Interception

The attacker intercepts a CoAP message, such as a GET request for sensor data, sent from Device A to Server B over the network. This interception occurs by monitoring the network traffic between the devices.



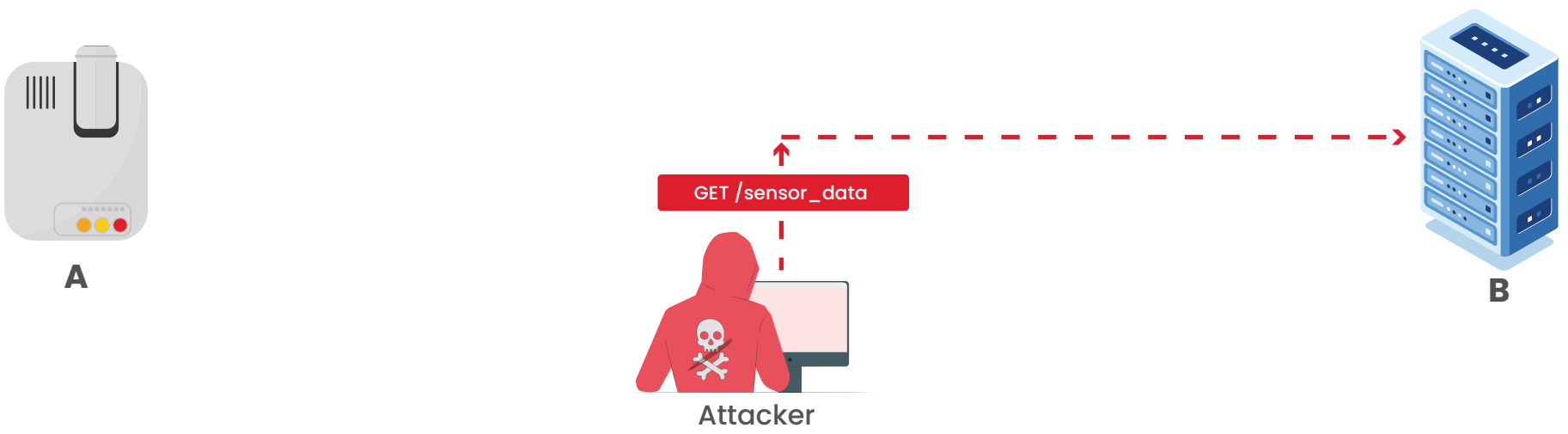
2.1.2 Capture

The intercepted CoAP message, containing the specific request or command, is captured and stored by the attacker. This involves saving the raw data packets that make up the CoAP message.



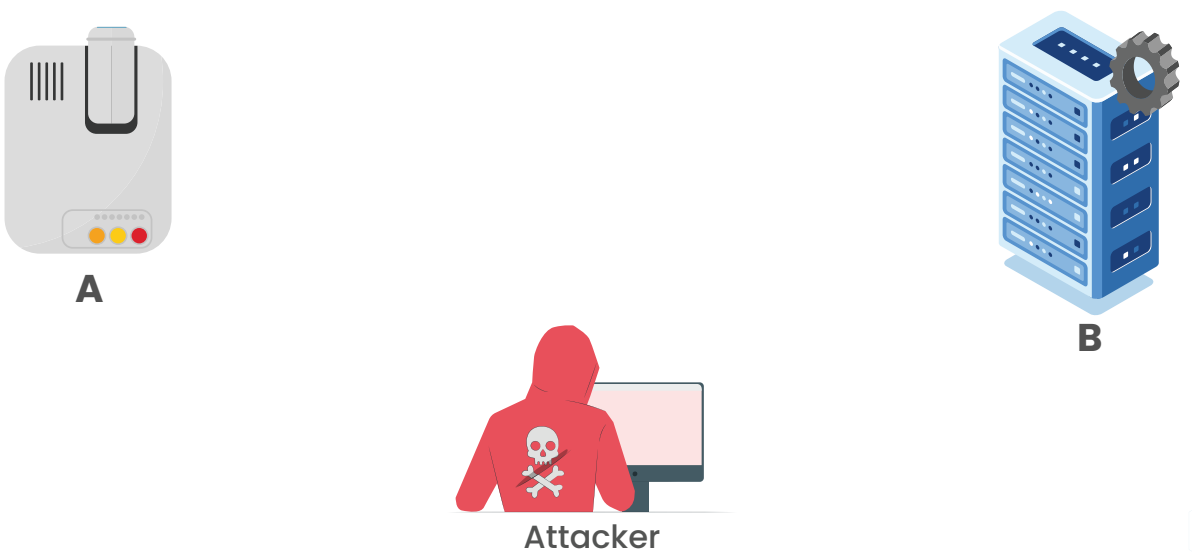
2.1.3 Replay

Later, the attacker retransmits the captured CoAP message back onto the network. This replayed message is sent to Server B, replicating the original request from Device A.



2.3.4 Effect

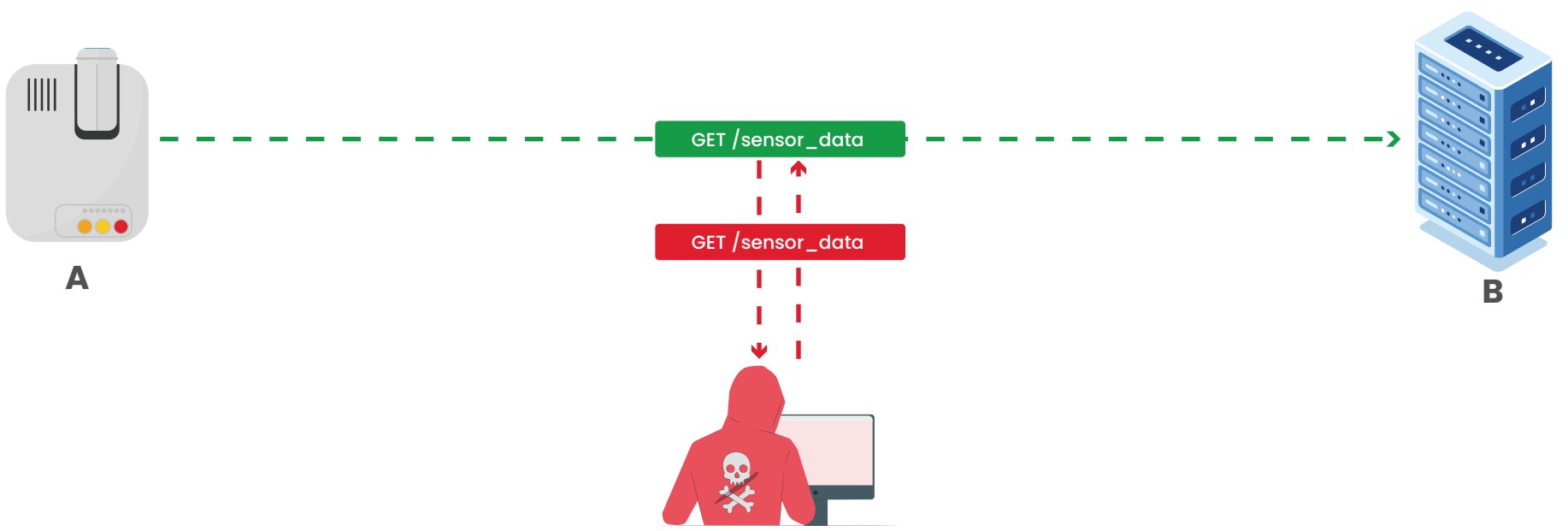
Server B, upon receiving the replayed CoAP message, interprets it as a legitimate request from Device A to fetch sensor data. It processes the request and sends back the requested data as a response.



3. Types of Replay Attack

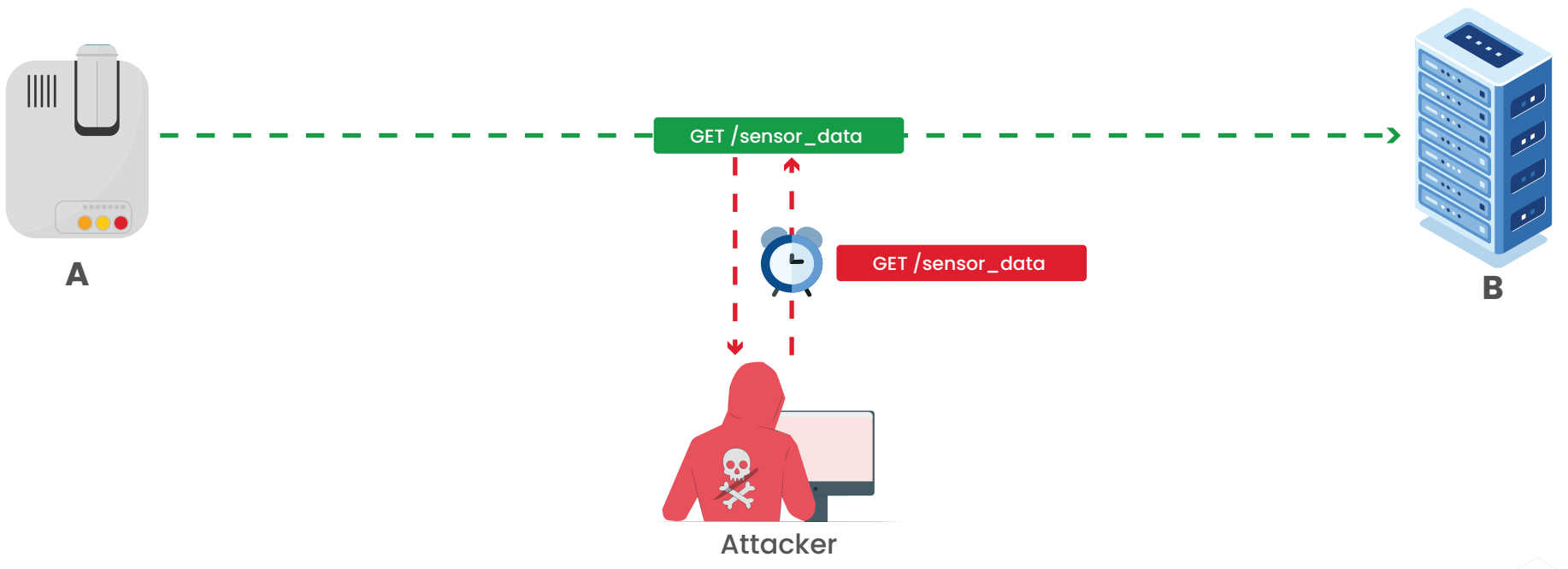
3.1 Basic Replay Attack

In a basic replay attack, the attacker intercepts CoAP messages exchanged between an IoT device (Device A) and a server (Server B). The attacker simply saves the intercepted message and retransmits it later, without making any modifications. This deception can lead to unintended actions or unauthorized access within the IoT network.



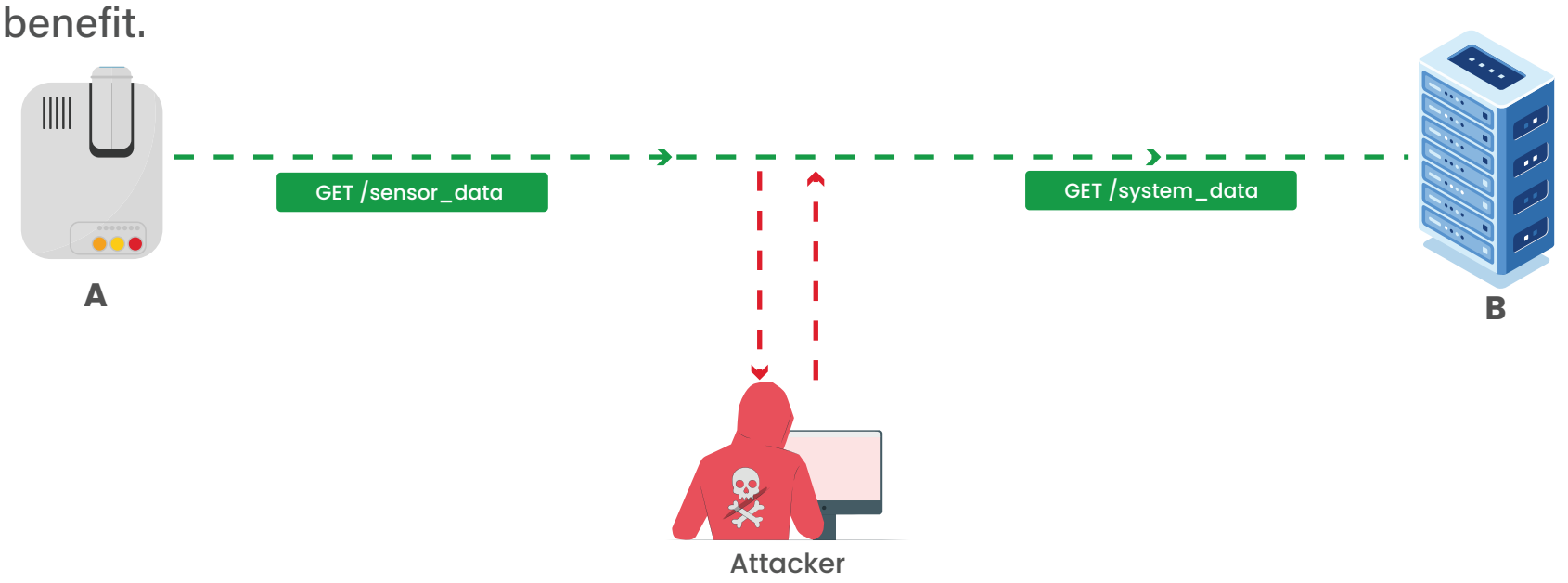
3.2 Delayed Replay Attack

Unlike a basic replay attack, a delayed one involves the attacker holding onto intercepted CoAP messages and strategically retransmitting them later. This delay might be timed to exploit network vulnerabilities or operational lapses, increasing the likelihood of successful unauthorized access or disruption.



3.3 Adaptive Replay Attack

An adaptive replay attack is more sophisticated, where the attacker not only intercepts but also modifies the intercepted CoAP messages before retransmission. This modification could involve altering timestamps, payload contents, or other parameters within the message. By adapting the message contents, the attacker aims to manipulate device behaviors or deceive server responses for their benefit.



4. Tools

4.1 Scapy

Scapy is a powerful interactive packet manipulation program and library for Python. It allows for the creation, manipulation, decoding, and sending of network packets. With Scapy, attackers can craft custom CoAP messages, intercept existing ones, and replay them on the network.

https://github.com/secdev/scapy

4.2 Copper (Cu)

Copper is a CoAP user-agent client for Firefox browsers, available as a browser extension. It allows users to interact with CoAP servers and inspect CoAP messages exchanged over the network. While primarily used for legitimate testing purposes, it can be adapted to capture and replay CoAP messages for malicious intents.

https://github.com/mkovatsc/Copper

4.3 CoAPthon

CoAPthon is a Python library and framework for CoAP (Constrained Application Protocol). It includes tools for both client and server-side CoAP implementations. It can be adapted for testing purposes, including replay attacks, by manipulating CoAP messages programmatically.

https://github.com/Tanganelli/CoAPthon

5. Impacts

5.1 Device Manipulation

Malicious actors can manipulate device behavior by replaying control commands. This could lead to turning devices on or off, changing settings, or triggering specific actions, disrupting normal operations and potentially causing harm.

5.2 Denial of Service (DoS)

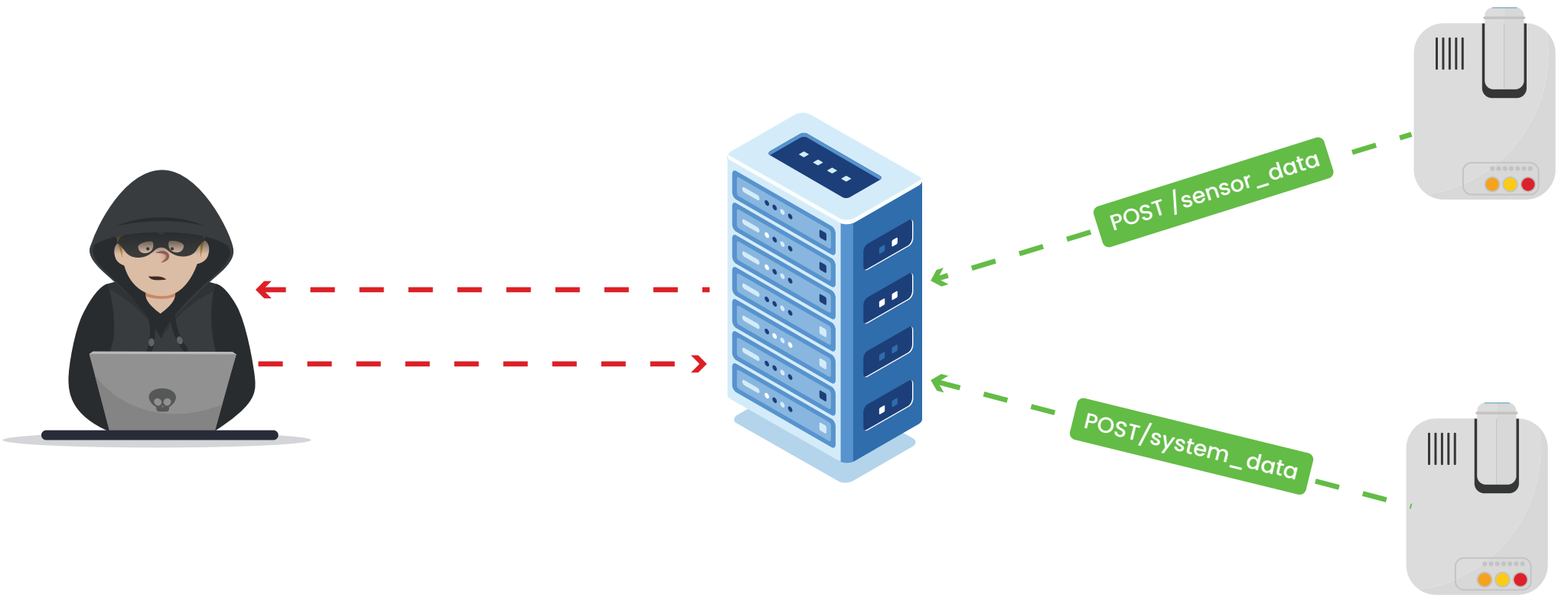
By repeatedly replaying certain CoAP messages, attackers can overwhelm the server or device with excessive requests, leading to resource exhaustion and rendering the service unavailable to legitimate users.

5.3 Unauthorized Access

Attackers can gain unauthorized access to IoT devices or systems by replaying previously intercepted messages, allowing them to execute commands or retrieve sensitive information without proper authentication.

5.4 Confidentiality Breach

Sensitive information transmitted over CoAP, such as sensor data or control commands, can be intercepted and replayed, leading to potential exposure of confidential information.



6. Prevention

6.1 Message Authentication Codes (MACs)

Use MACs to ensure the integrity and authenticity of CoAP messages. By appending a cryptographic signature to each message, any tampering or replay attempts can be detected and rejected by the recipient.

6.2 Timestamps

Incorporate timestamps and unique identifiers in CoAP messages. This ensures that each message is fresh and can only be processed within a specific timeframe, preventing attackers from successfully replaying old messages.

6.3 Session Tokens

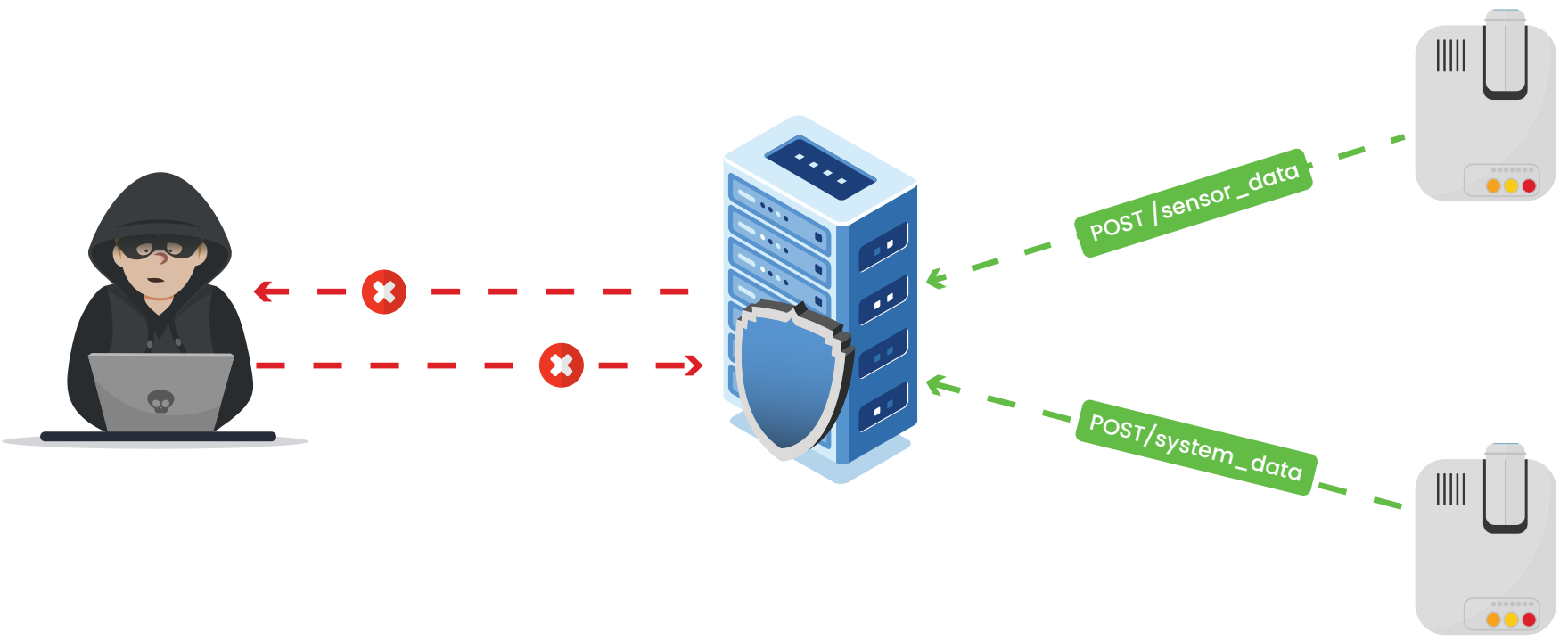
Use session tokens to manage communication sessions between devices. Each session can have a unique token that expires after a certain period, making it difficult for attackers to reuse intercepted messages.

6.4 Replay Detection Mechanisms

Implement replay detection mechanisms on the server side. Maintain a log of recent message IDs and reject any messages that contain duplicates within a specified timeframe.

6.5 Short Message Lifetimes

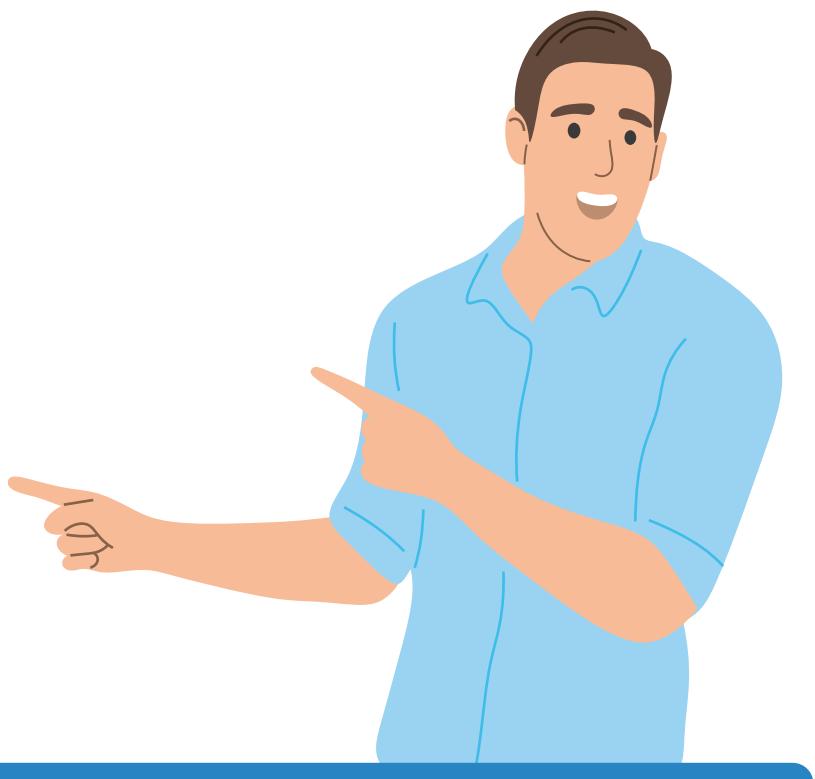
Configure CoAP messages to have short lifetimes, reducing the window of opportunity for an attacker to replay a message. This can be achieved by setting the Max-Age option appropriately.



SECURITYBOAT

Frontline Of Your Business

COAP RELAY ATTACK HANDBOOK



Scan QR Code to Download Handbook

www.securityboat.net