info@apriorit.com
www.apriorit.com

Case study

# Developing a Custom ICAP Server for Traffic Filtering and Analysis

Our client is a cybersecurity services provider offering SaaS solutions for detecting and protecting against known vulnerabilities and zero-day exploits. They wanted to enable delivery of their services via the Internet Content Adaptation Protocol (ICAP). The Apriorit team researched possible approaches to creating an ICAP server for sanitizing files that our client can use to extend the possibilities of their proxy servers and build such a solution.

## The client

Our client is an international company providing SaaS solutions for corporate cybersecurity. The goal of their products is to ensure the detection of and protection against cyberthreats, with a special focus on zero-day vulnerabilities. The company offers a wide range of SaaS products for analyzing and securing data sent by or downloaded from emails and web applications.

## The challenge

Our client already had a cloud service with a web API for sanitizing potentially harmful files and detecting direct threats. Now they wanted to enable file sanitizing with an ICAP server for filtering all files sent or downloaded with their proxy servers. As Apriorit has relevant expertise in this area, they entrusted us with this task.

## Our solution

For this project, we formed a team of experienced web developers, a quality assurance specialist, and a project manager. After researching several possible scenarios and eliciting product requirements from the client, we offered to build a custom ICAP server.

## The result

Our team first created an MVP that could analyze uploaded files and decide whether they needed to be sent to the client's server for further processing. Then we developed a unique custom ICAP server with extended functionality that could not only analyze uploaded files but also edit and replace them if a threat was detected. Both solutions were successfully integrated into our client's system.

### Characteristics of the final ICAP server solution

- Written in .NET Framework and C#
- Works with HTTP and HTTPS
- Supports Windows, Linux, and macOS
- Processes both POST and PUT requests
- Can be fully integrated with the existing system

## Choosing the right technical approach

For their new solution, our client wanted to use ICAP — a lightweight HTTP-like protocol used for extending the possibilities of proxy servers. This protocol can analyze and modify almost any type of traffic, including files, browser requests, and server responses.

Initially, our client wanted us to build a solution that met two criteria:

- Written in the .NET Framework and C#
- Supports the Windows operating system

**The first technical approach** we considered was to use an open-source ICAP server as the basis for our solution. We decided on c-icap server, a popular ICAP server implementation.

However, c-icap has two major technical limitations that contradicted our client's requirements:

1. It works only on Linux
2. It only supports C

The first limitation could be dealt with by using Docker containers for porting the developed Linux solution to Windows. Such an approach was even in line with our client's decision to move the architecture of their existing solution to a combination of Linux, Docker, and

Kubernetes. So theoretically, using Docker for porting a Linux solution to Windows would ease the integration process and improve the product's adjustability.

The main downside of this approach was the fact that c-icap doesn't support C# and only supports C. Deeper research also showed that c-icap lacked some crucial functionality. But if we added all the layers needed to cover this functionality gap, the system performance would be significantly degraded.

**A solution based on c-icap**

| | |
|---|---|
| Easy to implement | Doesn't support C# and Windows |
| Fits the client's plans for further changes in system architecture | Slows down system performance |

Therefore, we started looking for an alternative solution. **The second technical approach we considered** was to build a custom C++ ICAP server from scratch. In this way, we could ensure that the final solution met all our client's requirements in full and performed well.

After evaluating several possible options, we decided to write the custom ICAP server in .Net Core. This allowed us to create a unique and fully cross-platform solution that would support not only Windows (as required by the client) but also Linux and macOS. In this way, the server could be easily adjusted to the client's needs and successfully built into their existing system.

**Custom ICAP server**

| | | |
|---|---|---|
| Meets all the client's requirements | Ensures great system performance | Makes the final solution fully cross-platform |

Thanks to our thoughtful project management and thorough assessment of risks related to the use of a third-party solution, the team managed to meet the initial budget and provide the final product on time even after reconsidering the initial technical solution.

## Building a custom ICAP server

Once we settled on the right technical approach, we moved to the practical implementation of the custom ICAP server. We split the development process into two major stages:

**Key stages of the project**

| | |
|---|---|
| 1 | 2 |
| Creating a minimum viable product (MVP) | Developing and integrating a fully functional ICAP server |

At each stage, we worked closely with the client, assisting them in the integration and further adjustment of the developed solution.

### 1. Creating an MVP

Before starting development activities, we elicited project requirements, prepared a detailed development plan, and approved it with the client. First, the client wanted our team to build an MVP with minimal functionality that could be successfully integrated into their existing system. This solution would help them evaluate the potential of the new service and gather feedback from end users.

In particular, the MVP needed to support:

- the HTTP protocol
- POST requests.

Our client planned to implement support for the HTTPS protocol and PUT requests during the next stage of the project.

**MVP requirements**

| | |
|---|---|
| ✓ HTTP | ✗ HTTPS |
| ✓ POST requests | ✗ PUT requests |

To speed up development and meet the strict deadlines set by the client, during this stage, we used c-icap as the foundation for our ICAP server.

At the end of this stage, we provided the client with a ready solution that could work with any proxy and a full set of detailed documentation on it. Apriorit specialists also assisted the client with integrating the ready MVP into their system and adjusting it.

### 2. Developing and integrating a fully functional ICAP server

After the successful delivery of the MVP, our client decided to move to the next stage of the project and start developing a custom ICAP server with full functionality.

At this point, we switched from using a third-party solution as the basis for our project to building a custom ICAP server for sanitizing any file. Just as during the first stage, we started by outlining the requirements for the final solution, discussing and approving them with the client. We also provided the client with detailed estimates and development plans.

**Full ICAP server requirements**

| | |
|---|---|
| ✓ HTTP | ✓ HTTPS |
| ✓ POST requests | ✓ PUT requests |

At this stage of the project, we developed a fully functional ICAP server capable of working with the HTTP and HTTPS protocols as well as POST and PUT requests. The server was integrated into our client's solution, providing them with all the functionality needed for further traffic filtering and analysis.

The general workflow of the final solution consists of four key steps:

1. A third-party proxy redirects user traffic to the ICAP server.
2. On the ICAP server, a special module analyses user-originated files for threats, and suspicious files are sanitized.
3. Safe files are sent back to the user.
4. A PDF report on detected threats is sent to the user instead of sanitized files.

## Challenges and solutions

Throughout this project, the Apriorit team faced several challenges:

- **Detecting user-originated files in traffic.** Our client's ICAP server was supposed to define which files to send for further analysis and which to ignore. The server needed to analyze and sanitize only files that were uploaded or downloaded by a user. Therefore, we needed to ensure that all website files — cookies, scripts, fonts, images, audio and video files — would be skipped.

- **Working with uploaded files.** Files are usually uploaded using multipart POST requests. To detect such files in the traffic, we needed to implement a special parser for POST requests. This parser detects all parts of the file being uploaded, analyzes them, and replaces the entire file if necessary.

- **Changing the names of sanitized files.** When a threat is detected, the ICAP server is supposed to replace the original file with a PDF report on the identified cybersecurity issue. Therefore, alongside changing the contents of the uploaded files, our ICAP server has to alter the code and change the names of these files in browser requests. To enable this feature, we needed to implement an automatic change of requests so they would refer to the name and extension of the PDF report instead of the original file. The main challenge here was that the file name can be presented in the request in several ways, and the server has to parse and work with all of them.

While solving all these challenges, the Apriorit team developed a high-performance cross-platform ICAP server for file analysis and sanitization that provides a great alternative to the c-icap plugin and meets all of our client's requirements.

## The impact

Our client has successfully implemented the developed ICAP server into their existing product and is now distributing this solution to their customers worldwide. Enabling this method of service delivery allowed our client to attract new customers and increase revenue.

Satisfied with the quality of our cooperation, our client also introduced Apriorit to one of their affiliates — a company that selects and adjusts software solutions to the needs of corporate customers. Now, Apriorit is assisting this second company in further customization of the ICAP server solution.

Apriorit developers possess unique expertise in C, C#, and C++ development. Entrust your next project to our team and we'll make it real together.