



Análise de Malware

Relatório TP2

Integração de Sistemas de Informação

Aluno: Adelino Daniel da Rocha Vilaça - a16939
Docente: Professor Luis Gonzaga Martins Ferreira

Licenciatura em Engenharia de Sistemas Informáticos (PL)

3º ano

Barcelos | Dezembro, 2024

Lista de Abreviaturas e Siglas

Índice de Figuras

Figura 1 - Visualização do Modelo de Entidade Relação do Projeto	7
Figura 2 - Visualização do Novo Modelo de Entidade Relação do Projeto.....	8
Figura 3 - Modelo da tabela File e os respectivos atributos.....	9
Figura 4 Modelo da tabela FileAnalysis e os respectivos atributos.....	9
Figura 5 - Modelo da tabela User e os respectivos atributos.....	10
Figura 6 - Associação dos Modelos aos Objetos através do DbContext.....	10
Figura 7 - Resposta do serviço de Upload do Ficheiro utilizado Swagger como plataforma de testes da API	18
Figura 8 - Resposta do serviço de Leitura do ID do ficheiro utilizado Swagger como plataforma de testes da API.....	19
Figura 9 - Resposta do serviço de Análise de uma Assinatura utilizado Swagger como plataforma de testes da API.....	20
Figura 10 - Imagem relativa ao Script em Python para análise dos dados gerados pela resposta da API e formatação para JSON "User-Friendly"	21
Figura 11 - Imagem relativa ao sample e assinatura (sha256) utilizados para teste dos serviços da API	22

Índice

1.	Enquadramento / Tema	6
2.	Problema	6
3.	Modelo de Entidade Relação (ER / Data Model)	7
4.	Novo Modelo de ER (Simplificado para demonstração do funcionamento esperado)	8
5.	Modelagem	9
6.	Arquitetura	11
6.1.	Serviço de Submissão de Ficheiros para Análise	11
6.2.	Serviço de Consulta de Resultados após Análise de Ficheiros	11
6.3.	Serviço de Pesquisa de Arquivos ou URLs por Hash (Validar)	12
7.	Código	13
7.1.	Serviço de Upload do Ficheiro	13
7.2.	Serviço de Análise por ID	13
7.3.	Serviço de Análise por Hash	14
8.	Endpoints	14
8.1.	Serviço de Upload do Ficheiro	14
8.2.	Serviço de Análise por ID	15
8.3.	Serviço de Análise por Hash	15
9.	App Cliente (Página Web – ASP .NET MVC)	16
9.1.	Serviço de Upload do Ficheiro (Cliente)	16
9.2.	Serviço de Análise por ID (Cliente)	16
9.3.	Serviço de Análise por Hash (Cliente)	17
10.	Testes API (Swagger)	18
10.1.	/api/File/scan	18
10.2.	/api/File/report/{analysisId}	19
10.3.	/api/File/scanHash/{fileHash}	20
11.	Script para Formatação dos dados em JSON	21

12. Hash de Malware Utilizada	22
13. Documentação API VirusTotal	23
14. Conclusão e Trabalhos Futuros (Concluir no Relatório Final)	24
15. Bibliografia	24

1. Enquadramento / Tema

Este trabalho foi realizado no âmbito da Unidade Curricular de Integração de Sistemas de Informação, do curso de Engenharia de Sistemas Informáticos (Pós-Laboral). O objetivo deste segundo trabalho prático é o desenvolvimento de competências utilizando arquiteturas REST/SOAP, relativamente a este trabalho será REST, baseando-se no desenvolvimento de processos de interoperabilidade entre sistemas, assentes em serviços web.

O tema do projeto foca-se na Análise de Malware em ficheiros e URLs, utilizando a API do VirusTotal para detectar assinaturas de Malware. Este sistema permite que utilizadores façam upload de ficheiros e URLs para análise, com resultados detalhados sobre a presença de ameaças. A aplicação tem grande utilidade no contexto da Cibersegurança, auxiliando na identificação de ficheiros e websites maliciosos antes que possam comprometer sistemas ou dados sensíveis. Além disso, a análise de Malware automatizada contribui para a melhoria da segurança digital, oferecendo uma ferramenta prática e eficaz para a prevenção de ataques, podendo ser usada por qualquer pessoa mesmo sem conhecimentos técnicos na área.

Ao longo do projeto provavelmente irão ser feitas algumas alterações a nível estrutural para acrescentar/reduzir serviços para que não ponham em causa a finalização deste segundo trabalho e por consequentemente a cadeira de ISI.

2. Problema

O principal objetivo deste trabalho é demonstrar e resolver problemas relacionados com a análise de ficheiros e URLs para a deteção de Malware. O sistema visa permitir o envio de ficheiros e URLs por parte dos utilizadores, sendo analisados com a API do [VirusTotal](#), que identifica assinaturas de Malware presentes. O desafio principal envolve a integração eficiente da API externa, a organização e análise dos resultados obtidos, bem como a gestão de dados sensíveis, garantindo a segurança e eficácia no processo de deteção.

3. Modelo de Entidade Relação (ER / Data Model)

O sistema permite analisar ficheiros e URLs submetidos por utilizadores para deteção de Malware, utilizando a API do VirusTotal. Cada utilizador pode enviar múltiplos ficheiros ou URLs (dentro do limite diário da API ou de acordo com o tipo de Role que o mesmo tem atribuído), que são analisados, com os resultados registados nas tabelas FileAnalysis e UrlAnalysis.

As "signatures" de Malware detetadas são associadas às análises através de tabelas intermediárias (FileAnalysisMalware e UrlAnalysisMalware). Assim o modelo de dados garantirá uma estrutura normalizada e eficiente para investigar as análises e as assinaturas de Malware relacionadas.

A tabela Role é utilizada para definir os diferentes papéis dos utilizadores, como Admin e User, permitindo que o sistema controle as permissões de acesso e as limitações de utilização da API.

Além disso, a tabela UserSession é utilizada para gerir as sessões dos utilizadores. Cada vez que um utilizador se autentica no sistema, uma nova entrada é criada na tabela UserSession, por sua vez registando o novo token de autenticação.

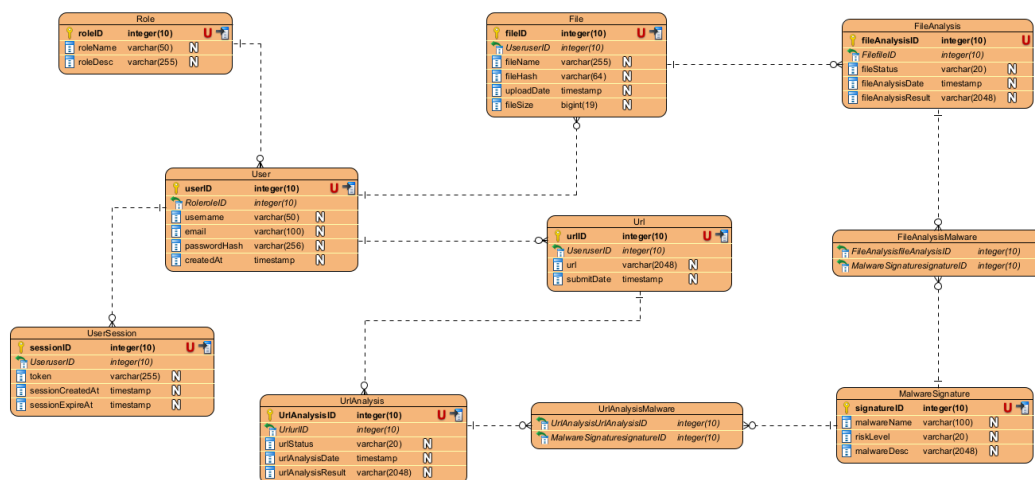


Figura 1 - Visualização do Modelo de Entidade Relação do Projeto

4. Novo Modelo de ER (Simplificado para demonstração do funcionamento esperado)

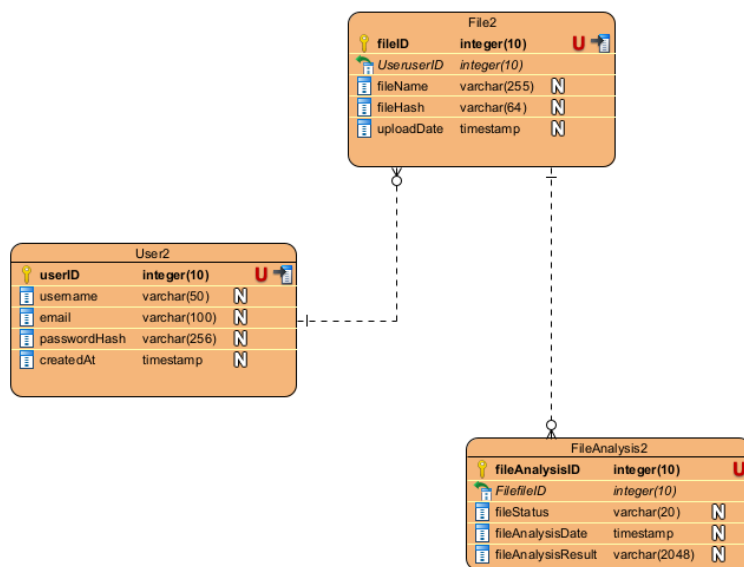


Figura 2 - Visualização do Novo Modelo de Entidade Relação do Projeto

5. Modelagem

```
namespace AnaliseMalwareServices.Models
{
    4 references
    public class File
    {
        [Key]
        0 references
        public int FileID { get; set; }
        [Required]
        0 references
        public int UserID { get; set; }
        0 references
        public string? FileName { get; set; }
        0 references
        public string? FileHash { get; set; }
        0 references
        public DateTime? UploadDate { get; set; }

        // Navigation property
        [ForeignKey("UserID")]
        0 references
        public User User { get; set; }
    }
}
```

Figura 3 - Modelo da tabela File e os respectivos atributos

```
public class FileAnalysis
{
    [Key]

    public int FileAnalysisID { get; set; }
    [Required]

    public int FileID { get; set; }
    0 references
    public string FileStatus { get; set; }
    0 references
    public DateTime? FileAnalysisDate { get; set; }
    0 references
    public string FileAnalysisResult { get; set; }

    // Navigation property
    [ForeignKey("FileID")]
    0 references
    public File File { get; set; }
}
```

Figura 4 Modelo da tabela FileAnalysis e os respectivos atributos

```
3 references
public class User
{
    [Key]
    0 references
    public int UserID { get; set; }
    0 references
    public string Username { get; set; }
    0 references
    public string Email { get; set; }
    0 references
    public string? PasswordHash { get; set; }
    0 references
    public DateTime? CreatedAt { get; set; }
}
```

Figura 5 - Modelo da tabela User e os respectivos atributos

```
public class ApiContext : DbContext
{
    public DbSet<File> Files { get; set; }
    public DbSet<FileAnalysis> FilesAnalysis { get; set; }
    public DbSet<User> Users { get; set; }

    public ApiContext(DbContextOptions<ApiContext> options)
        : base(options)
    {
    }
}
```

Figura 6 - Associação dos Modelos aos Objetos através do DbContext

6. Arquitetura

6.1. Serviço de Submissão de Ficheiros para Análise

Objetivo: Permitir que os utilizadores façam upload de ficheiros para análise de Malware.

Descrição: O utilizador envia um ficheiro através do serviço, que será então enviado à API da VirusTotal para análise, retornando os resultados ao mesmo.

Possível Endpoint:

POST /File/scan

Input: Ficheiro a ser analisado (binário / base64).

Output: ID do ficheiro gerado pela VirusTotal para consultas futuras e a confirmação de que o ficheiro foi enviado para análise (Status).

API: Através do endpoint *file/scan* ou *file/upload* da API do VirusTotal, enviamos o ficheiro para ser analisado e para obter uma hash para monitorizar o status da mesma.

6.2. Serviço de Consulta de Resultados após Análise de Ficheiros

Objetivo: Permitir que os utilizadores consultem os resultados da análise de um ficheiro previamente submetido.

Descrição: O utilizador pode consultar os resultados da análise de um ficheiro com base no ID que foi retornado após o envio para a API do VirusTotal.

Endpoint sugerido:

GET /File/report/{analysisId}

Input: ID do ficheiro (após upload).

Ouput: Normalização do JSON com Malware detectado (por exemplo, o nome do Malware, a Classificação da Ameaça, o Número de Antivírus que possivelmente detetaram o ficheiro malicioso, etc).

API: Através do endpoint *file/report* da API do VirusTotal, consultamos os resultados da análise de um ficheiro específico, incluindo detalhes sobre as assinaturas de Malware que foram identificadas (JSON)

6.3. Serviço de Pesquisa de Arquivos ou URLs por Hash (Validar)

Objetivo: Permitir que o utilizador faça uma pesquisa direta pela hash (MD5, SHA1, ou SHA256) de um ficheiro para verificar rapidamente se ele já foi analisado pelo VirusTotal.

Descrição: O serviço permite que o utilizador forneça uma hash e verifique se essa assinatura já foi analisada.

Endpoint sugerido:

GET /File/scanHash/{fileHash}

Entrada: Hash do ficheiro (MD5, SHA1 ou SHA256).

Saída: Status da análise (se foi analisado anteriormente e, em caso afirmativo, apresenta os resultados).

API: Através do *endpoint file/report* ou *url/report* para consultar rapidamente o status de um ficheiro, caso ele já tenha sido analisado previamente.

7. Código

7.1. Serviço de Upload do Ficheiro

```
// Upload a file to VirusTotal for scanning
1 reference
public async Task<string> UploadFileAsync(byte[] fileData, string fileName)
{
    var tempFilePath = Path.Combine(Path.GetTempPath(), fileName);
    await File.WriteAllBytesAsync(tempFilePath, fileData);

    var request = new RestRequest("files", Method.Post);
    request.AddHeader("x-apikey", _apiKey);
    request.AddFile("file", tempFilePath);

    var response = await _client.ExecuteAsync(request);

    File.Delete(tempFilePath);
    return response.Content;
}
```

Figura 7 - Serviço REST para Upload do Ficheiro

7.2. Serviço de Análise por ID

```
// Method for getting full report using analysis ID
1 reference
public async Task<string> GetFullReportAsync(string analysisId)
{
    var request = new RestRequest($"analyses/{analysisId}", Method.Get);
    request.AddHeader("x-apikey", _apiKey);

    var response = await _client.ExecuteAsync(request);
    return response.Content;
}
```

Figura 8 - Serviço REST para Análise por ID

7.3. Serviço de Análise por Hash

```
// Method for checking file hash
1 reference
public async Task<string> CheckFileHashAsync(string fileHash)
{
    var request = new RestRequest($"files/{fileHash}", Method.Get);
    request.AddHeader("x-apikey", _apiKey);

    var response = await _client.ExecuteAsync(request);

    if (response.IsSuccessfull)
    {
        return response.Content; // Returning the detailed scan result (JSON)
    }
    else
    {
        return $"Error: {response.StatusCode} - {response.Content}";
    }
}
```

Figura 9 - Serviço REST para Análise por Assinatura

8. Endpoints

8.1. Serviço de Upload do Ficheiro

```
// File Upload to VirusTotal API
[HttpPost("scan")]
0 references
public async Task<IActionResult> ScanFile(IFormFile file)
{
    if (file == null || file.Length == 0)
        return BadRequest("No file uploaded.");

    using (var memoryStream = new MemoryStream())
    {
        await file.CopyToAsync(memoryStream);
        var fileData = memoryStream.ToArray();

        // Call the VirusTotal API to upload the file for scanning
        var scanResult = await _virusTotalApiClient.UploadFileAsync(fileData, file.FileName);

        // Return the response from VirusTotal
        return Ok(scanResult);
    }
}
```

Figura 10 - Endpoint para chamada do serviço REST de Upload do Ficheiro

8.2. Serviço de Análise por ID

```
// Endpoint to get the full analysis report by analysis ID
[HttpGet("report/{analysisId}")]
0 references
public async Task<IActionResult> GetReport(string analysisId)
{
    try
    {
        // Get the full report using the analysis ID
        var report = await _virusTotalApiClient.GetFullReportAsync(analysisId);

        // Return the full report
        return Ok(report);
    }
    catch (Exception ex)
    {
        return BadRequest(new { message = ex.Message });
    }
}
```

Figura 11 - Endpoint para chamada do serviço REST de Análise por ID

8.3. Serviço de Análise por Hash

```
// Endpoint for scanning a file hash (e.g., MD5, SHA256)
[HttpGet("scanHash/{fileHash}")]
0 references
public async Task<IActionResult> ScanFileHash(string fileHash)
{
    if (string.IsNullOrEmpty(fileHash))
    {
        return BadRequest("File hash is required.");
    }

    // Use the VirusTotalApiClient to get the scan report for the file hash
    var scanResult = await _virusTotalApiClient.CheckFileHashAsync(fileHash);

    // If scan result contains an error, return it as a bad request
    if (scanResult.Contains("Error"))
    {
        return BadRequest(scanResult);
    }

    // Return the successful result (the scan report in JSON format)
    return Ok(scanResult);
}
```

Figura 12 - Endpoint para chamada do serviço REST para Análise por Assinatura

9. App Cliente (Página Web – ASP .NET MVC)

9.1. Serviço de Upload do Ficheiro (Cliente)

AnaliseMalwareApp

Upload File

Get Report

Scan by Hash

Upload File for Malware Analysis

Choose a file to upload:

Explorar...

Nenhum ficheiro selecionado.

Upload

Server Response:

```
{
  "data": {
    "type": "analysis",
    "id": "OWZkMDRhNWw4Y2E4N2YzZTY5NzIxNDUxYjZlNjgxMTI6MTczNTU4NjI4NQ==",
    "links": {
      "self": "https://www.virustotal.com/api/v3/analyses/OWZkMDRhNWw4Y2E4N2YzZTY5NzIxNDUxYjZlNjgxMTI6MTczNTU4NjI4NQ=="
    }
  }
}
```

© 2024 AnaliseMalwareApp - ISI - a16939

Figura 13 - Visualização da chamada do Serviço de Upload do Ficheiro para o Cliente

9.2. Serviço de Análise por ID (Cliente)

AnaliseMalwareApp

Upload File

Get Report

Scan by Hash

Get Report

Analysis ID:

Get Report

Server Response:

```
{
  "data": {
    "type": "analysis",
    "id": "OWZkMDRhNWw4Y2E4N2YzZTY5NzIxNDUxYjZlNjgxMTI6MTczNTU4NjI4NQ==",
    "links": {
      "self": "https://www.virustotal.com/api/v3/analyses/OWZkMDRhNWw4Y2E4N2YzZTY5NzIxNDUxYjZlNjgxMTI6MTczNTU4NjI4NQ==",
      "item": "https://www.virustotal.com/api/v3/files/bb44434158648a50964fe5342e6d3111e35d3d90198162cb7ce92badce32fa"
    },
    "attributes": {
      "results": [
        {
          "stats": {
            "malicious": 0,
            "suspicious": 0,
            "undetected": 0,
            "harmless": 0
          },
          "timeout": 0,
          "confirmed_timeout": 0,
          "failure": 0,
          "type": "unsupported"
        }
      ],
      "status": "queued",
      "date": "1735586215",
      "meta": {
        "file_info": {
          "sha256": "bb44434158648a50964fe5342e6d3111e35d3d90198162cb7ce92badce32fa",
          "md5": "9fd04a5b8ca87f3e69721451b6e68112",
          "sha1": "e77fabd496aad7f86ed6936fe2700271f9792b63",
          "size": 95476
        }
      }
    }
  }
}
```

© 2024 AnaliseMalwareApp - ISI - a16939

Figura 14 - Visualização da chamada do Serviço de Análise por ID para o Cliente

9.3. Serviço de Análise por Hash (Cliente)

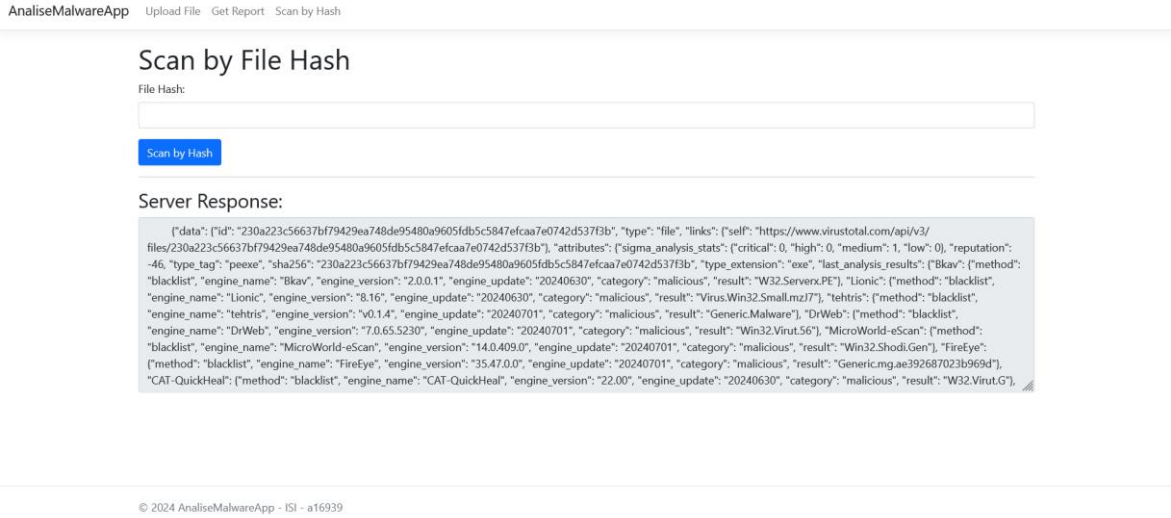


Figura 15 - Visualização da chamada do Serviço de Análise por Assinatura para o Cliente

10. Testes API (Swagger)

10.1. /api/File/scan

The screenshot displays the Swagger UI for the 'AnaliseMalware' API. The selected definition is 'AnaliseMalwareServices v1'. The endpoint being tested is 'POST /api/File/scan'. The request body is set to 'multipart/form-data' and contains a file named 'T188-JYMB1QU574.pdf'. The 'Execute' button has been clicked, resulting in a '200 OK' response. The response body is a JSON object with the following structure:

```
{
  "data": {
    "type": "analysis",
    "id": "0624N0R0M6i4Y2t4N2YzZlY5NzIsNDh0YjJlNjgwMTI0RTczNTU4MzQwMA=="
  },
  "links": {
    "self": "https://www.virustotal.com/api/v3/analyses/0624N0R0M6i4Y2t4N2YzZlY5NzIsNDh0YjJlNjgwMTI0RTczNTU4MzQwMA=="
  }
}
```

The response headers are:

```
content-type: text/plain; charset=utf-8
date: Mon, 30 Dec 2024 18:35:35 GMT
server: Kestrel
x-firefox-spy: h2
```

The 'Responses' section shows a table with the following data:

Code	Description	Links
200	OK	No links

Below the response details, there are two other API endpoints listed:

- GET /api/File/report/{analysisId}
- GET /api/File/scanHash/{filehash}

Figura 16 - Resposta do serviço de Upload do Ficheiro utilizado Swagger como plataforma de testes da API

10.2. /api/File/report/{analysisId}

The screenshot shows the Swagger API interface for the endpoint `/api/File/report/{analysisId}`. The interface includes a Swagger logo, a 'Select a definition' dropdown, and an 'Authorize' button. The main area displays the endpoint details, parameters, and the response body.

Endpoint: `GET /api/File/report/{analysisId}`

Parameters:

Name	Description
<code>analysisId</code>	required
<code>string (path)</code>	<code>zhXNDUxYzINjgMTi6MTGZCNTU4Mzc0MA==</code>

Responses:

200

Response body:

```
{
  "data": {
    "id": "0M2K0R0N0M14Y2E4NZYzZlY3M2I3N0XVjZlNjgMTi6MTGZCNTU4Mzc0MA==",
    "type": "analysis",
    "links": {
      "self": "https://www.virustotal.com/api/v3/analyses/0M2K0R0N0M14Y2E4NZYzZlY3M2I3N0XVjZlNjgMTi6MTGZCNTU4Mzc0MA=="
    },
    "attributes": {
      "date": "2024-12-30T18:36:20Z",
      "status": "completed",
      "stats": {
        "malicious": 0,
        "suspicious": 0,
        "undetected": 11,
        "harmless": 0,
        "failure": 0,
        "confirmed_timeout": 0
      },
      "type": "unreported",
      "result": {
        "Bkav": {
          "method": "blacklist",
          "engine_name": "Bkav",
          "engine_version": "2.0.0.1",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Ikarus": {
          "method": "blacklist",
          "engine_name": "Ikarus",
          "engine_version": "2.0.0.1",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Microsoid-escan": {
          "method": "blacklist",
          "engine_name": "Microsoid-escan",
          "engine_version": "34.0.409.0",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "ClamAV": {
          "method": "blacklist",
          "engine_name": "ClamAV",
          "engine_version": "1.4.11.0",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "CTX": {
          "method": "blacklist",
          "engine_name": "CTX",
          "engine_version": "2024.8.29.1",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "CAT-QuickHeal": {
          "method": "blacklist",
          "engine_name": "CAT-QuickHeal",
          "engine_version": "22.00",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Avast": {
          "method": "blacklist",
          "engine_name": "Avast",
          "engine_version": "2021.2.0.4005",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "McAfee": {
          "method": "blacklist",
          "engine_name": "McAfee",
          "engine_version": "6.0.6.603",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Malwarebytes": {
          "method": "blacklist",
          "engine_name": "Malwarebytes",
          "engine_version": "4.5.5.54",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Zillya": {
          "method": "blacklist",
          "engine_name": "Zillya",
          "engine_version": "2.0.0.5003",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Sangfor": {
          "method": "blacklist",
          "engine_name": "Sangfor",
          "engine_version": "2.25.10.0",
          "engine_update": "20241227",
          "category": "undetected",
          "result": null
        },
        "K7AntiVirus": {
          "method": "blacklist",
          "engine_name": "K7AntiVirus",
          "engine_version": "12.200.54332",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "ESET-NOD32": {
          "method": "blacklist",
          "engine_name": "ESET-NOD32",
          "engine_version": "15.1.0.0",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "TrendMicro-HouseCall": {
          "method": "blacklist",
          "engine_name": "TrendMicro-HouseCall",
          "engine_version": "23.0.8494.0",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Avast": {
          "method": "blacklist",
          "engine_name": "Avast",
          "engine_version": "4.0.1.4",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Cyren": {
          "method": "blacklist",
          "engine_name": "Cyren",
          "engine_version": "22.0.1.28",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Kaspersky": {
          "method": "blacklist",
          "engine_name": "Kaspersky",
          "engine_version": "22.0.1.28",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "BitDefender": {
          "method": "blacklist",
          "engine_name": "BitDefender",
          "engine_version": "7.2",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "NANO-Antivirus": {
          "method": "blacklist",
          "engine_name": "NANO-Antivirus",
          "engine_version": "1.0.140.25790",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "SUPERAntiSpyware": {
          "method": "blacklist",
          "engine_name": "SUPERAntiSpyware",
          "engine_version": "5.6.0.1012",
          "engine_update": "20241227",
          "category": "undetected",
          "result": null
        },
        "Inocent": {
          "method": "blacklist",
          "engine_name": "Inocent",
          "engine_version": "1.0.0.1",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Emsisoft": {
          "method": "blacklist",
          "engine_name": "Emsisoft",
          "engine_version": "2024.1.0.51952",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        },
        "Avira": {
          "method": "blacklist",
          "engine_name": "Avira",
          "engine_version": "16.0.0.0",
          "engine_update": "20241230",
          "category": "undetected",
          "result": null
        }
      }
    }
  }
}
```

Response headers:

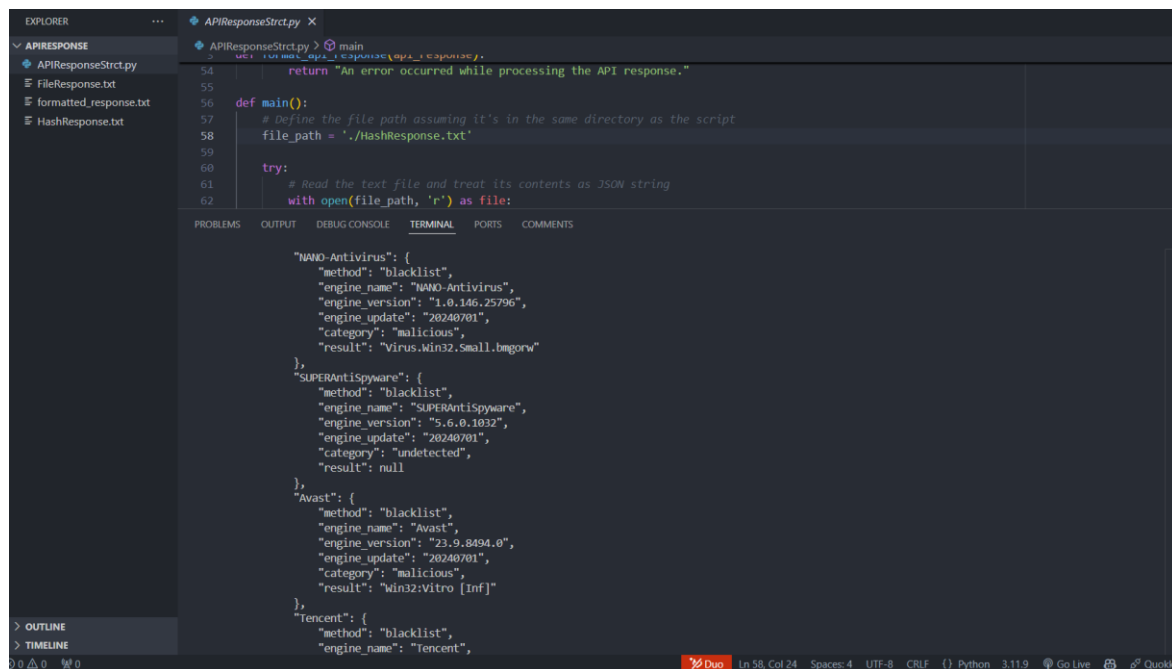
```
content-type: text/plain; charset=utf-8
date: Mon, 30 Dec 2024 18:36:20 GMT
server: Kestrel
x-frame-options: deny
```

Responses:

Code	Description	Links
200	OK	No links

Figura 17 - Resposta do serviço de Leitura do ID do ficheiro utilizado Swagger como plataforma de testes da API

11. Script para Formatação dos dados em JSON



```
EXPLORER
  APIRESPONSE
    APIResponseStrct.py
    FileResponse.txt
    formatted_response.txt
    HashResponse.txt

APIResponseStrct.py
54     return "An error occurred while processing the API response."
55
56 def main():
57     # Define the file path assuming it's in the same directory as the script
58     file_path = './HashResponse.txt'
59
60     try:
61         # Read the text file and treat its contents as JSON string
62         with open(file_path, 'r') as file:

```

```
"NANO-Antivirus": {
  "method": "blacklist",
  "engine_name": "NANO-Antivirus",
  "engine_version": "1.0.146.25796",
  "engine_update": "20240701",
  "category": "malicious",
  "result": "Virus.Win32.Small.bmgowr"
},
"SuperAntispyware": {
  "method": "blacklist",
  "engine_name": "SUPERAntispyware",
  "engine_version": "5.6.0.1032",
  "engine_update": "20240701",
  "category": "undetected",
  "result": null
},
"Avast": {
  "method": "blacklist",
  "engine_name": "Avast",
  "engine_version": "23.9.8494.0",
  "engine_update": "20240701",
  "category": "malicious",
  "result": "Win32:Vitro [Inf]"
},
"Tencent": {
  "method": "blacklist",
  "engine_name": "Tencent",

```

Figura 19 - Imagem relativa ao Script em Python para análise dos dados gerados pela resposta da API e formatação para JSON "User-Friendly"

12. Hash de Malware Utilizada

TEAM CYMRU

Search

Signup

Docs

Malware Hash Registry (MHR)

This web form provides a manual interface for checking hashes against our malware data. Type in one or more hashes into the box below, then press "submit" to see if we recognize the hash as malicious.

1d31bd48b2e864c773ca6a3b9fd0019416809066
22232b5821a1ea9afa2c89bcd87392755e6d643b
30906e3f8bae78f852eb441965a957e68c6c4957
64fd93ef54bbab55b956de71089ee3c4aae852de
36127f11432f4e6cc0df0080da271a1b6060d553
2e9f41ca2846683158cd2e108fe405079910bdd7
436879fe88a928f483c6066434fb7c3c40ce9da2
46c6a243281c2590a0e1499412ba4d3eab38e91f
60765071b09254a3e53c945350edc965be2ff3fe
68ff97056ee6c0b74f9c73717c3ed114de271663

Max Hash limit: 1000

Submit

Reset

Supported Hashes

- MD5
- SHA1
- SHA256

Format

- Hashes can be newline and/or comma-separated
- White space is ignored
- There is a limit of 1000 hashes per-submission

APIs

- DNS
- WHOIS
- REST (requires [signup!](#))

Sample Input

1d31bd48b2e864c773ca6a3b9fd0019416809066
22232b5821a1ea9afa2c89bcd87392755e6d643b
30906e3f8bae78f852eb441965a957e68c6c4957
64fd93ef54bbab55b956de71089ee3c4aae852de
36127f11432f4e6cc0df0080da271a1b6060d553
2e9f41ca2846683158cd2e108fe405079910bdd7
436879fe88a928f483c6066434fb7c3c40ce9da2
46c6a243281c2590a0e1499412ba4d3eab38e91f
60765071b09254a3e53c945350edc965be2ff3fe
68ff97056ee6c0b74f9c73717c3ed114de271663

Results

Hash:

The queried hash, plus its conversion to the other supported hash types, if available.

AV Hit Rate:

The percent of anti-virus (AV) engines we tested that detected this hash as malicious.

Last Seen:

The last time we ran this hash against our AVs and determined this hash was malicious.

Error:

If there was an error searching for this hash's data, it will be displayed here.

Hash	AV Hit Rate	Last Seen	Error
MD5: - SHA1: 46c6a243281c2590a0e1499412ba4d3eab38e91f SHA256: -	-	-	-
MD5: 53f871aeca2eabc299452bd7872f4f SHA1: 1d31bd48b2e864c773ca6a3b9fd0019416809066 SHA256: 8354e3050cd540d0731cd0d5538a0bf1f9bb503bf2240a32cdc09c46aee1770c	86%	2017-01-12T16:23:41Z	-
MD5: 9759c08f2bfcdae5263cde858a8045e94 SHA1: 64fd93ef54bbab55b956de71089ee3c4aae852de SHA256: 2eb18f18837c775b22ed87d195994d1564ef3ca2f50c0e97077d8794249843ea	85%	2016-03-20T14:17:09Z	-
MD5: ae392687023b969d8bd91f4869132c80 SHA1: 2e9f41ca2846683158cd2e108fe405079910bdd7 SHA256: 280c9228d56537bf9f9429ea743de95480a9605fdb5c5847efcaa7e0742d537f3b	97%	2016-05-11T00:38:12Z	-
MD5: 02f07327c6e0123c18fd7764e12e015a SHA1: 436879fe88a928f483c6066434fb7c3c40ce9da2 SHA256: -	67%	2013-10-14T02:35:10Z	-
MD5: c6a2d7f94d93390078a35eabd00afe45 SHA1: 68ff97056ee6c0b74f9c73717c3ed114de271663 SHA256: c083bb3e6c705d9cdcd6bd7b14c1a9982faa703e5edcbebed041b105b68e9af	95%	2017-01-19T01:34:43Z	-
MD5: dc291b7fb395c04aa0ded9c6ed4ea61f SHA1: 22232b5821a1ea9afa2c89bcd87392755e6d643b SHA256: 45e163f5e2a6f4d0b9b82653c698952b9e38b776192852445c0f26f02768b8ea	34%	2015-05-09T11:06:34Z	-
MD5: 1431d9869ecd6c014728a054a86b360c SHA1: 30906e3f8bae78f852eb441965a957e68c6c4957 SHA256: -	83%	2013-10-14T01:34:48Z	-
MD5: 13433c643ae47750d4ba2a3793874f7a SHA1: 36127f11432f4e6cc0df0080da271a1b6060d553 SHA256: -	53%	2022-01-01T11:53:02Z	-
MD5: - SHA1: 60765071b09254a3e53c945350edc965be2ff3fe SHA256: -	-	-	-

WARNING: Sources that are seen making batch requests to the MHR server with large numbers of individual queries instead of using the available bulk interfaces may be rate limited. Sources issuing an abusively large number of queries may be null routed.

Copyright © 2024 Team Cymru. All Rights Reserved.

Figura 20 - Imagem relativa ao sample e assinatura (sha256) utilizados para teste dos serviços da API

13. Documentação API VirusTotal

- [Upload a file for scanning](#): analysis your file with 70+ antivirus products, 10+ dynamic analysis sandboxes and a myriad of other security tools to produce a threat score and relevant context to understand it;
- [Get a file report by hash](#): given a {md5, sha1, sha256} hash, retrieves the pertinent analysis report including threat reputation and context produced by 70+ antivirus products, 10+ dynamic analysis sandboxes and a myriad of other security tools and datasets;
- [Scan URL](#): analysis your URL with 70+ antivirus products/blocklists and a myriad of other security tools to produce a threat score and relevant context to understand it;
- [Get a URL analysis report](#): given a URL, retrieves the pertinent analysis report including threat reputation and context produced by 70+ antivirus products/blocklists and a myriad of other security tools and datasets;
- [Get a domain report](#): given a domain, retrieves the pertinent analysis report including threat reputation and context produced by 70+ antivirus products/blocklists and a myriad of other security tools and datasets;
- [Get a domain report](#): given a domain, retrieves the pertinent analysis report including threat reputation and context produced by 70+ antivirus products/blocklists and a myriad of other security tools and datasets;
- [Get an IP address report](#): given an IP address, retrieves the pertinent analysis report including threat reputation and context produced by 70+ antivirus products/blocklists and a myriad of other security tools and datasets;

14. Links para as Apps

14.1. Backend Serviços (Swagger)

analysemalwareservices-bxhrh2agbsaaf9gt.spaincentral-01.azurewebsites.net/

14.2. Página Web para Cliente usar os Serviços

<https://analysemalwareclientapp-g9bqegayd6cebae9.spaincentral-01.azurewebsites.net/>

15. Conclusão e Trabalhos Futuros

Apesar do desenvolvimento contínuo deste 2º trabalho prático, para a cadeira de Integração de Sistemas de Informação, não correr como esperado, penso que tirei bastantes mais valias, analisando de forma geral o projeto em si. Foram implementados serviços Rest e endpoints, apesar de não existir conexão à Base de Dados colocada na Cloud (apesar de estar configurada), devido a um erro de Token entre a migração da BD, através dos Modelos e a BD em si, tentarei futuramente descobrir o problema para que não me volte a prejudicar um projeto. O backend dos serviços tal como a parte gráfica para o Cliente (Página Web em ASP .NET) estão na Azure (Cloud), podendo serem visualizáveis a qualquer altura e em qualquer dispositivo. Gostaria também de ter implementado serviços em SOAP com conexão ao 1º trabalho prático em KNIME para uma total integração de diversos sistemas. Também gostaria de ter desenvolvido mais serviços REST como para análise de URLs e IP's. Por fim sinto que teria mais para dar neste contexto de integração, mas terei mais chances para me redimir nos próximos projetos. Obrigado

16. Bibliografia

Não existem origens no documento atual.