Faculdade de Informática e Administração Paulista

Leonardo Lago Garroti

FAWKES

SÃO PAULO

2025

Leonardo Lago Garroti

FAWKES

Trabalho desenvolvido como parte da Avaliação de todas as Unidades Curriculares

Orientador: Profº Fábio Pires

SÃO PAULO

2025

FOLHA DE APROVAÇÃO DO PROJETO

Leonardo Lago Garroti

FAWKES

Trabalho de avaliação Challenge, apresentado à Faculdade de Informática e Administração Paulista como requisito parcial para a conclusão das Unidades Curriculares sob a orientação do Professor: Profº Fábio Pires.

Data: ____/____/____ _____

Assinatura do professor orientador

OBSERVAÇÕES: _____

_____

_____

_____

_____

RESUMO

O Fawkes Anti-Ransomware foi desenvolvido com o objetivo de oferecer uma solução prática, inteligente e eficiente para a detecção e mitigação de ataques de ransomware em sistemas Windows. A ferramenta combina múltiplas camadas de defesa — honeypots, verificação de assinaturas digitais e análise comportamental — para identificar ameaças em tempo real e reagir automaticamente antes que os arquivos sejam comprometidos.

A iniciativa de criar o Fawkes surgiu da crescente necessidade de soluções acessíveis e eficazes contra ransomwares, que continuam sendo uma das maiores ameaças à segurança da informação. O projeto reflete a aplicação prática dos conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas, unindo fundamentos de programação, sistemas operacionais e segurança cibernética em uma ferramenta funcional e de fácil utilização.

Além de sua base técnica sólida, o Fawkes se destaca por sua interface gráfica intuitiva, desenvolvida em PySide6, que permite ao usuário monitorar processos, visualizar logs e gerenciar arquivos em quarentena. O empacotamento do sistema como executável independente (.exe) garante portabilidade e praticidade, tornando-o aplicável tanto em contextos acadêmicos quanto corporativos.

O presente relatório detalha a arquitetura, funcionamento e evolução do projeto, destacando seus diferenciais e a relevância de seu papel na defesa proativa contra ameaças digitais.

Palavras-chave: ransomware, honeypot, antivírus, segurança da informação, Python, detecção comportamental.

AGRADECIMENTOS

Aos professores e colegas que ajudaram diretamente para a nossa formação, mostrando o mundo da pesquisa e da educação. Ao professor Fábio, pelo feedback, suporte, conhecimento e paciência, auxiliando-nos no caminho daquilo que não pode ser mensurado, tomado ou roubado, apenas conquistado: O conhecimento. Por fim, a todos que de alguma forma contribuíram nesta jornada acadêmica.

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicações
AV	Antivírus
EXE	Arquivo Executável
GUI	Interface Gráfica do Usuário
PID	Identificador de Processo
VM	Máquina Virtual
WMI	Interface de Gerenciamento do Windows
YARA	Ferramenta de análise e detecção baseada em regras

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

FOLHA DE APROVAÇÃO DO PROJETO 3

1 INTRODUÇÃO 8

1.1 JUSTIFICATIVA E MOTIVAÇÃO 8

1.2 OBJETIVOS 9

1.2.1 OBJETIVO GERAL. 9

DESENVOLVER E VALIDAR UM AGENTE ANTI-RANSOMWARE CAPAZ DE DETECTAR ATIVIDADES MALICIOSAS EM SISTEMAS WINDOWS E RESPONDER AUTOMATICAMENTE PARA MITIGAR IMPACTOS, PRESERVANDO EVIDÊNCIAS PARA AUDITORIA. 9

1.2.2 OBJETIVOS ESPECÍFICOS 9

1.3 MATERIAIS E MÉTODOS 10

1.4 PROBLEMA 11

2 EVOLUÇÃO DO PROJETO 12

2.1 PRIMEIRA VERSÃO — DETECÇÃO COMPORTAMENTAL 12

2.1.1 SEGUNDA VERSÃO — INTEGRAÇÃO DE HONEYPOTS 12

2.1.2 VERSÃO FINAL — ARQUITETURA COMPLETA COM INTERFACE GRÁFICA 12

3 ARQUITETURA E DIFERENCIAIS 14

4 MODOS DE DETECÇÃO 15

5 FLUXO DE FUNCIONAMENTO 16

6 LINGUAGEM E BIBLIOTECAS 17

1 INTRODUÇÃO

O Fawkes é um agente anti-ransomware desenvolvido em Python que combina monitoramento em tempo real, criação de honeypots e verificação de assinatura digital para detectar e mitigar atividades maliciosas em sistemas Windows. O motor opera observando diretórios sensíveis, respondendo a eventos de criação e modificação de arquivos, criando artefatos deliberadamente atraentes para identificar atacantes e bloqueando processos ou removendo arquivos quando um comportamento suspeito é confirmado. A implementação integra componentes de baixo nível do Windows para verificação de assinatura, monitoramento de processos e observação de sistema de arquivos, além de uma interface gráfica desenvolvida com PySide6 para controle e visualização das operações.

1.1 Justificativa e Motivação

O aumento constante de ataques de ransomware nas últimas décadas tem causado prejuízos significativos a empresas, órgãos públicos e usuários domésticos. Esses ataques exploram falhas de segurança para criptografar dados e exigir resgate financeiro, comprometendo a integridade e a disponibilidade das informações.

Durante o curso de Cyber Segurança, identificou-se a oportunidade de aplicar conhecimentos práticos de programação, redes e segurança da informação em uma solução voltada à proteção de dados. A escolha pelo desenvolvimento do Fawkes surgiu da necessidade de criar um sistema que fosse simples de operar, mas eficiente na detecção precoce de ameaças.

A motivação principal para o projeto foi desenvolver uma ferramenta capaz de atuar de forma proativa, detectando comportamentos maliciosos antes que o dano ocorra, algo que muitas soluções tradicionais de antivírus não realizam em tempo real. Além disso, o Fawkes busca servir como prova de conceito acadêmica, demonstrando a viabilidade de combinar diferentes métodos de defesa — *honeypots*, assinaturas digitais e análise comportamental — em um único sistema funcional.

1.2 Objetivos

1.2.1 Objetivo geral.

Desenvolver e validar um agente anti-ransomware capaz de detectar atividades maliciosas em sistemas Windows e responder automaticamente para mitigar impactos, preservando evidências para auditoria.

1.2.2 Objetivos específicos

Os objetivos específicos são:

- Implementar um sistema de honeypots que crie artefatos atrativos em diretórios sensíveis e dispare alertas em caso de acesso ou modificação.

- Integrar a verificação de assinatura digital de binários via API nativa do Windows (WinVerifyTrust) para identificar executáveis não confiáveis.

- Desenvolver um mecanismo de monitoramento comportamental (contagem de operações por PID em janela temporal) para identificar padrões típicos de ransomware.

- Registrar logs detalhados e mover arquivos suspeitos para quarentena, permitindo análise posterior.

- Empacotar a aplicação como executável (.exe) e documentar o processo de geração para reprodutibilidade.

1.3 Materiais e métodos

Metodologia de desenvolvimento e teste:

O projeto adotou uma abordagem iterativa de desenvolvimento seguida de testes em ambiente controlado. Os testes foram realizados em máquinas virtuais Windows isoladas, garantindo segurança e possibilidade de restauração por snapshots.

Ambiente de teste:

- Sistema: Windows (VM) com rede isolada;
- Pastas monitoradas: Desktop, Documents, Downloads, Pictures e Videos;
- Diretórios gerados automaticamente pelo Fawkes: FAWKES_AV_Logs, FAWKES_AV_Quarantine e FAWKES_AV_Honeypots.

Ferramentas utilizadas:

- Linguagem: **Python 3.11+**;

- Bibliotecas principais: watchdog, psutil, PySide6, wmi, pythoncom, pywin32;
- Empacotamento: **PyInstaller**;
- Ferramentas auxiliares: Sysinternals (Process Explorer, Handle).

1.4 Problema

Atualmente, soluções baseadas exclusivamente em assinaturas apresentam limitações diante de variantes novas de ransomware e execuções por script. É necessária uma defesa proativa que combine validação de executáveis, detecção comportamental e preservação de evidências. O Fawkes foi projetado para suprir essa lacuna com uma abordagem em múltiplas camadas.

2 EVOLUÇÃO DO PROJETO

O desenvolvimento do **Fawkes** passou por três etapas principais de aprimoramento, refletindo a evolução da proposta inicial até a entrega do sistema final.

2.1 Primeira Versão — Detecção Comportamental

A primeira versão do projeto tinha como foco principal o monitoramento comportamental de processos em tempo real. A ideia original era identificar possíveis ransomwares observando o volume de modificações e criações de arquivos realizadas por cada processo ativo. Quando um processo ultrapassava um limite seguro de operações, ele era automaticamente encerrado, evitando a propagação da criptografia de arquivos. Essa abordagem serviu como base para compreender padrões de atividade suspeita e implementar um mecanismo de resposta automatizada.

2.1.1 Segunda Versão — Integração de Honeypots

Na segunda fase, o projeto foi aprimorado com a adição de um sistema de honeypots — arquivos isca distribuídos em diretórios sensíveis, projetados para atrair ações maliciosas. Esse novo método permitiu uma detecção mais precoce e assertiva, já que qualquer tentativa de modificação em um honeypot era interpretada como sinal imediato de ataque. A integração entre o monitoramento comportamental e o módulo de honeypots elevou significativamente a precisão da detecção.

2.1.2 Versão Final — Arquitetura Completa com Interface Gráfica

A versão final consolidou todas as melhorias anteriores e adicionou novos recursos, como a verificação de assinaturas digitais de executáveis, o sistema de

quarentena automática, e uma interface gráfica desenvolvida em PySide6, que permite o controle e a visualização em tempo real das operações.
O empacotamento do projeto em formato .exe possibilitou a execução autônoma em sistemas Windows, tornando o Fawkes uma solução completa, modular e de fácil distribuição.

3 ARQUITETURA E DIFERENCIAIS

O Fawkes possui uma arquitetura centrada em um motor de tempo real (RealtimeEngine), responsável por coordenar observadores de sistema de arquivos, um módulo de bloqueio de executáveis não assinados (BlockUnsigned), um gerenciador de honeypots (HoneypotManager) e utilitários de proteção de pastas. A aplicação inclui também uma interface gráfica desenvolvida em PySide6, com bandeja do sistema, painéis de controle e visualização de logs. Entre seus principais diferenciais estão a defesa em múltiplas camadas, resposta automatizada, modularidade, interface intuitiva e registros forenses detalhados.

4 MODOS DE DETECÇÃO

O Fawkes detecta ameaças por três vetores principais:

- **Honeypot:** cria arquivos falsos em diretórios monitorados e dispara bloqueio imediato em caso de modificação.
- **Assinatura Digital:** valida executáveis (.exe, .dll, .sys, .ocx) por meio da API **WinVerifyTrust**, bloqueando binários não confiáveis.
- **Monitoramento Comportamental:** conta operações de processos (PID) em janelas de tempo; se um processo exceder o limite configurado de modificações, é classificado como suspeito e encerrado automaticamente.

Além disso, o sistema aplica verificações heurísticas que identificam nomes e extensões de arquivos potencialmente maliciosos.

5 FLUXO DE FUNCIONAMENTO

O Fawkes monitora continuamente diretórios sensíveis, cria honeypots, intercepta novos processos e verifica assinaturas digitais. Ao detectar comportamento suspeito, o sistema executa ações automáticas, como encerramento do processo, quarentena de arquivos e registro detalhado no log. Cada evento é documentado com PID, nome do processo, horário e tipo de ação, permitindo auditoria completa das atividades e análise forense posterior.

6 LINGUAGEM E BIBLIOTECAS

O projeto foi desenvolvido em **Python 3.11+**

As principais bibliotecas utilizadas incluem:

- **watchdog:** monitoramento de alterações em arquivos;
- **psutil:** controle e análise de processos;
- **PySide6:** criação da interface gráfica (GUI);
- **wmi** e **pythoncom:** monitoramento e interação com processos no Windows;
- **ctypes** e **win32api:** comunicação com APIs nativas;
- **hashlib**, **json** e **sqlite3:** armazenamento e manipulação de dados locais.

7 GERAÇÃO DO EXECUTÁVEL (.EXE)

O **Fawkes** foi empacotado como uma aplicação independente para Windows utilizando o **PyInstaller**, que converte o código Python em um arquivo executável (.exe) contendo todas as dependências necessárias. O código-fonte completo e as instruções de compilação estão disponíveis no repositório oficial do projeto:

Repositório GitHub: https://github.com/garrotii/FAWKES

Para gerar o executável a partir do código, basta seguir os seguintes passos:

1. **Baixar o projeto** pelo GitHub ou clonar o repositório.
2. **Instalar as dependências** indicadas no arquivo requirements.txt.
3. **Executar o comando:**
4. python -m PyInstaller --onefile --windowed fa.py

O executável será criado automaticamente na pasta dist, permitindo que o Fawkes seja executado em qualquer máquina Windows sem a necessidade do Python instalado.

8 TRABALHOS FUTUROS

- Implementar detecção baseada em regras YARA;
- Centralizar registros de logs em banco de dados remoto;
- Criar instalador gráfico com opções de configuração;
- Expandir compatibilidade para sistemas Linux;
- Otimizar desempenho e reduzir consumo de recursos.

9 CONCLUSÃO

O Fawkes representa uma abordagem moderna e eficiente de defesa proativa contra ransomwares. Sua arquitetura modular e suas três camadas de detecção — *honeypots*, verificação de assinatura digital e monitoramento comportamental — permitem resposta imediata a ameaças em execução. O empacotamento em formato executável e a documentação clara de uso tornam o Fawkes uma ferramenta funcional, escalável e reprodutível, adequada tanto para testes acadêmicos quanto para ambientes corporativos controlados.