SYSFROZEN: ferramenta de proteção de pastas em ambientes windows utilizando Python e agendamento de tarefas

CARVALHO, Luciano Junior[1]; TREVIZANO, Waldir Andrade[2]
NOÉ, Paulo Ricardo Gregório[3]; BAIA, Joás Weslyei[4]

lucianorustlol@gmail.com
waldir@unifagoc.edu.br
joasweslei@gmail.com

[1] Graduado em Ciência da Computação - UNIFAGOC
[2] Docente do curso de Ciência da Computação do UNIFAGOC

RESUMO

Com o avanço tecnológico e a crescente dependência da sociedade em relação aos computadores e à internet, a segurança da informação tornou-se crucial. Este trabalho propõe o desenvolvimento do aplicativo SYSFROZEN, para sistemas Windows, que automatiza a proteção de pastas específicas contra acesso não autorizado, simplificando a segurança contra ameaças cibernéticas como vírus e malware. Inicialmente, foram identificadas vulnerabilidades no acesso a pastas, seguida por uma pesquisa para selecionar tecnologias adequadas para implementar o SYSFROZEN em Python, com foco em eficiência, segurança e facilidade de uso. O aplicativo possui interface gráfica intuitiva e funcionalidades como agendamento de proteção. Testes de segurança e desempenho validaram sua eficácia

Palavras-chave: Segurança. Proteção de dados. Vulnerabilidades. Proteção de pastas. Windows. Processos.

INTRODUÇÃO

O aumento das ameaças cibernéticas, como vírus, malware e phishing, tornou a proteção de dados e sistemas uma prioridade essencial em um mundo digitalizado. Uma das vulnerabilidades significativas reside no acesso não autorizado a pastas sensíveis em sistemas operacionais Windows, onde a presença de vírus pode resultar em danos graves, como perda de dados e exposição de informações confidenciais. Autores como Bruce Schneier (2000), Kevin Mitnick, Steve Wozniak e William Simon (2007), e Ross Anderson (2020) têm contribuído com insights para entender esses desafios e desenvolver soluções que fortaleçam a segurança cibernética.

A segurança da informação é uma preocupação crucial em um mundo cada vez mais digitalizado. Com o avanço da tecnologia, surgem também novas ameaças cibernéticas, que podem comprometer a confidencialidade, integridade e disponibilidade dos dados e sistemas. De acordo com o Relatório de Investigações de Violação de Dados da Verizon de 2023, 83% das violações de dados envolvem ataques cibernéticos baseados em *phishing, malware* ou credenciais comprometidas. Esses incidentes resultaram em prejuízos financeiros significativos, além de danos à reputação das empresas afetadas, destacando a necessidade crítica de práticas de segurança cibernética robustas e proativas para proteger os sistemas computacionais contra possíveis vulnerabilidades.

Objetivo geral

O objetivo deste trabalho é desenvolver a aplicação denominada SYSFROZEN, com o intuito de oferecer uma solução eficaz e automatizada para proteger pastas específicas contra acesso não autorizado em sistemas operacionais Windows, proporcionando uma camada adicional de proteção contra os diversos tipos de ameaças cibernéticas.

Objetivos específicos

1. Pesquisar e analisar as vulnerabilidades relacionadas ao acesso não autorizado a pastas em sistemas operacionais Windows, identificando as ameaças mais comuns e suas consequências.
2. Estudar e selecionar as tecnologias e bibliotecas mais adequadas para o desenvolvimento do aplicativo proposto, levando em consideração a eficiência, a segurança e a facilidade de implementação.
3. Projetar e desenvolver para o aplicativo SYSFROZEN uma interface gráfica intuitiva, que funcionará em desktop, permitindo aos usuários selecionar e congelar as pastas desejadas de forma simples e eficaz.
4. Implementar no software a funcionalidade de agendamento, integrando-o ao agendador de tarefas do sistema operacional, possibilitando a execução automática em intervalos definidos pelo usuário.

REFERENCIAL TEÓRICO

Segurança da Informação

A segurança da informação é uma preocupação crucial em um mundo cada vez mais digitalizado. Com o avanço da tecnologia, surgem também novas ameaças cibernéticas, que podem comprometer a confidencialidade, integridade e disponibilidade dos dados e sistemas. Para mitigar essas ameaças, é essencial compreender os princípios de segurança da informação, os diferentes modelos de controle de acesso e as melhores práticas de desenvolvimento de software.

Em seu estudo "Recent Trends in Cybersecurity Threats: a Comprehensive Analysis" (Smith; Johnson, 2021), os autores destacam a crescente sofisticação das ameaças cibernéticas e a necessidade de soluções proativas para proteger sistemas e dados contra tais ataques maliciosos. Isso ressalta a importância de programas que visam oferecer uma camada adicional de proteção contra acesso não autorizado a pastas sensíveis em sistemas Windows.

A Linguagem Python

No contexto do desenvolvimento de software, Python emerge como uma ferramenta poderosa. Em seu artigo "Python: A Versatile Tool for Software Development" (Gonzalez; Martinez, 2019), os autores exploram as diversas aplicações de Python em projetos de desenvolvimento de software, destacando sua flexibilidade e eficiência.

Ao manipular arquivos e pastas em Python, os desenvolvedores têm à sua disposição diversas bibliotecas que facilitam operações como listagem, criação e exclusão de arquivos e diretórios. Em seu artigo, Samadov (2023) demonstra como

essa integração pode ser realizada de forma eficiente, proporcionando conveniência e segurança para os usuários.

No estudo " Which Python GUI library should you use? " (Martin Fitzpatrick, 2024), o autor compara diferentes bibliotecas de interface gráfica em Python, fornecendo insights sobre as melhores práticas para o desenvolvimento de interfaces responsivas e intuitivas.

Dentre essas bibliotecas, destaca-se a System.Windows.Forms para a criação da interface gráfica devido à sua robustez, facilidade de uso e flexibilidade para o desenvolvimento de aplicativos Windows. Essa biblioteca fornece vários controles prontos para uso, como botões, caixas de texto, caixas de seleção, menus e outros elementos gráficos que facilitam o design da interface, e permite personalizar eventos e propriedades dos elementos, facilitando a criação de interfaces intuitivas e responsivas.

Outro ponto importante é a sua integração direta com o ambiente Windows, o que permite um desempenho eficiente, uso de recursos nativos e uma aparência familiar para os usuários

SOFTWARES RELACIONADOS

O aplicativo **Deep Freeze**, desenvolvido pela Faronics Corporation e criado por Dr. Farid Ali, é um software conhecido por sua capacidade de congelar o estado de um sistema operacional. Isso significa que qualquer alteração feita no sistema, como instalação de software, alterações de configuração ou arquivos criados/deletados, será revertida após um simples reinício do computador. Esse recurso é amplamente utilizado em ambientes onde é essencial manter a integridade e a consistência do sistema, como laboratórios de informática em escolas, universidades, bibliotecas públicas, cibercafés e em computadores compartilhados onde se deseja evitar problemas de segurança.

Além disso, existe o Time Freeze, um software similar desenvolvido pela ToolWiz Software. Assim como o Deep Freeze, o Time Freeze também oferece a capacidade de congelar o estado do sistema operacional. Isso significa que todas as alterações feitas, como instalações de software, modificações de configuração ou arquivos criados/deletados, são temporárias e serão revertidas para o estado original após um simples reinício do sistema. Essa funcionalidade é crucial para garantir a segurança e a integridade do sistema operacional em ambientes onde múltiplos usuários acessam o mesmo computador.

Ambos os softwares, Deep Freeze e Time Freeze, são ferramentas valiosas para administradores de sistema que precisam proteger computadores contra mudanças não autorizadas e manter a estabilidade do ambiente de trabalho.

O Sysfrozen, assim como o Deep Freeze e o Time Freeze, é uma ferramenta de proteção utilizada em ambientes compartilhados, permitindo que administradores garantam a integridade dos dados e a preservação do ambiente contra alterações não autorizadas. Todos os três softwares compartilham o objetivo de manter a estabilidade do sistema e proteger arquivos, com automação para executar essas tarefas de maneira eficiente. O Sysfrozen, desenvolvido em Python e com scripts de limpeza personalizáveis, permite mais controle sobre o que será apagado e quando, diferentemente do Deep Freeze, que aplica uma restauração geral

e fixa. Essa abordagem de personalização também permite que o Sysfrozen seja adaptado para cenários específicos sem necessidade de restauração completa.

MÉTODO DE DESENVOLVIMENTO

Para desenvolver o SYSFROZEN, foram definidos os requisitos funcionais e não funcionais do programa. Isso implicou entender como o congelamento de pastas funciona, quais são as operações permitidas e proibidas, como deveria ser a interface do usuário e quais as medidas de segurança a implementar.

Requisitos Funcionais:

- **Seleção de pastas**: permitir ao usuário escolher pastas específicas para serem protegidas pelo software.
- **Criação de script de limpeza**: gerar automaticamente um script em batch que exclui o conteúdo das pastas protegidas ao reiniciar o sistema.
- **Agendamento de execução**: integrar o script ao Agendador de Tarefas do Windows para que a limpeza ocorra periodicamente, sem necessidade de intervenção do usuário.
- **Interface intuitiva**: desenvolver uma interface gráfica que permita a seleção e configuração das pastas de maneira fácil e acessível.

Requisitos Não Funcionais:

- **Segurança e estabilidade**: garantir que o software opere com privilégios necessários para realizar as exclusões sem comprometer outras partes do sistema.
- **Compatibilidade com Windows**: assegurar que o aplicativo seja totalmente compatível com o sistema Windows, aproveitando recursos nativos para desempenho otimizado.
- **Eficiência de recursos**: o software deve operar sem consumir muitos recursos do sistema, mantendo-se leve e rápido mesmo durante a execução de tarefas agendadas.
- **Facilidade de instalação e configuração**: o SYSFROZEN deve ser fácil de instalar, configurar e utilizar por qualquer usuário, mesmo aqueles com conhecimento técnico limitado.

Requisitos de Sistema

Para garantir o funcionamento adequado do SYSFROZEN, é essencial que o ambiente de execução atenda aos seguintes requisitos:

- **Sistema Operacional**: O SYSFROZEN foi projetado para sistemas Windows 10 ou superiores.
- **Permissões de Usuário**: O software requer permissões administrativas para acesso a pastas protegidas e para o agendamento de tarefas no sistema.
- **Espaço em Disco**: Um mínimo de 100 MB de espaço livre é necessário para instalação e funcionamento.

- **Python Instalado**: O Python 3.8 ou superior deve estar instalado no sistema. O SYSFROZEN utiliza bibliotecas nativas e adicionais do Python para realizar a geração do script e a interface gráfica. O interpretador Python é indispensável, pois todo o código da ferramenta foi implementado nessa linguagem. Caso o Python não esteja instalado, recomenda-se realizar o download e a instalação a partir do site oficial (https://www.python.org) antes de executar o programa.

Definidos os requisitos, foi projetada a arquitetura do sistema, identificando os componentes principais, interface gráfica do usuário, gerenciamento de diretórios, gerador de script de limpeza, integrador agendador de tarefas. Finalmente, foi realizada a implementação do aplicativo, utilizando a linguagem Python, a biblioteca **System.Windows.Forms** para a construção da interface com o usuário e a biblioteca para o gerenciamento de arquivos e permissões do sistema operacional.

Este método de desenvolvimento assegurou que o desenvolvimento do SYSFROZEN ocorresse de maneira organizada e eficiente, atendendo às necessidades dos usuários e mantendo padrões de segurança e usabilidade.

RESULTADOS

Funcionalidade da Ferramenta

A ferramenta foi desenvolvida para oferecer uma solução simples e eficaz para proteger pastas em ambientes Windows, permitindo ao usuário agendar a limpeza automática de arquivos em diretórios selecionados. Sua interface gráfica facilita a navegação e seleção de pastas por meio de um sistema de busca interativo.

O fluxo principal da aplicação permite ao usuário selecionar diretórios específicos para serem protegidos, adicionando-os a uma lista de exclusão automática. Com essa configuração, a ferramenta gera um script customizado em batch que apaga automaticamente os arquivos e subdiretórios contidos nas pastas selecionadas toda vez que o sistema é reiniciado ou o usuário faz logon. Dessa forma, o conteúdo dessas pastas é "congelado" – ou seja, qualquer modificação feita nesses diretórios é removida, garantindo que eles voltem ao estado original a cada reinicialização, sem exigir intervenção manual.

Essa abordagem permite que o usuário mantenha determinados diretórios em um estado limpo e consistente, sendo uma solução prática para ambientes onde é necessário descartar alterações no conteúdo das pastas periodicamente.

A automação de sua execução é gerenciada pelo Agendador de Tarefas do Windows, com privilégios elevados, o que garante que a limpeza seja executada independentemente de o usuário estar conectado. Como resultado, a ferramenta atinge seu objetivo de proteger pastas com uma interface fácil de usar e mecanismos automáticos eficientes.

IMPLEMENTAÇÃO DA FERRAMENTA SYSFROZEN

Estrutura da Interface Gráfica (GUI) *Seletor de Pastas*

Para selecionar as pastas a serem protegidas, foi criado o formulário SeletorDePastasPersonalizadasForm. Esse formulário serve como a tela principal do

aplicativo, permitindo ao usuário navegar pelos diretórios do sistema e escolher as pastas que deseja proteger. Ele é intuitivo e oferece uma interface amigável para facilitar o processo de seleção e proteção de diretórios. e no Código 1 se mostra a definição do array onde a lista de pastas é adicionada:

Código 1 – array para as pastas a serem selecionadas

```
self.pastas_selecionadas = []  # Lista para armazenar diretórios
```

Navegação e Seleção de Diretórios

A navegação pelos diretórios no código é implementada de forma que o usuário possa explorar as pastas de seu sistema, visualizando o conteúdo de cada uma delas e selecionando aquelas que deseja proteger. Ao clicar duas vezes em uma pasta listada, o programa carrega os subdiretórios presentes e atualiza a lista exibida na interface. Além disso, há um botão que permite "voltar um nível", ou seja, retornar à pasta anterior na hierarquia de diretórios. O código também lida com possíveis erros, como tentativas de acessar pastas protegidas ou com permissões restritas, exibindo uma mensagem de erro amigável ao usuário.

No Código 2 a seguir, se mostra como este carregamento é feito, e como tais erros são tratados.

Código 2 – Carregamento dos diretórios selecionados

```
def carregar_diretorio(self, caminho): if not
        os.path.isdir(caminho):
            MessageBox.Show("Diretório inválido","Erro", MessageBoxButtons.OK) return

        self.diretorio_atual = caminho self.label_diretorio_atual.Text =
        f'Diretório Atual:
{self.diretorio_atual}'
        self.lista_pastas.Items.Clear()

# Adiciona diretórios à lista de itens try:
    for entrada in sorted(os.listdir(self.diretorio_atual)): caminho_completo =
        os.path.join(self.diretorio_atual, entrada) if os.path.isdir(caminho_completo):
            self.lista_pastas.Items.Add(self.prefixo_pasta + entrada +

except PermissionError:
    MessageBox.Show("Você não tem permissão para acessar esta pasta.", os.sep)
"Erro", MessageBoxButtons.OK)
```

Criação de Scripts Automatizados

O aplicativo gera automaticamente um script de limpeza no formato *.bat* baseado nas pastas que o usuário seleciona para serem protegidas. Esse script permite que o conteúdo das pastas selecionadas seja removido de forma automatizada quando a tarefa agendada for executada. Uma vez que as pastas são escolhidas pelo usuário, esse script executa os comandos de exclusão de arquivos e subdiretórios dentro dessas pastas. Esse arquivo é salvo em um diretório

apropriado, garantindo que o sistema execute a limpeza quando necessário, sem exigir interação manual. No Código 3 se mostra como tal tarefa é efetuada.

Código 3 – Criação automática dos scripts para excluir os arquivos

```
def criar_script(self):
    # Definir o caminho para a pasta Documentos do usuário pasta_documentos =
    os.path.join(os.path.expanduser("~"), "Documents")

    # Criar a pasta sysfrozen dentro da pasta Documentos, caso não exista pasta_sysfrozen
    = os.path.join(pasta_documentos, "sysfrozen")
    if not os.path.exists(pasta_sysfrozen): os.makedirs(pasta_sysfrozen)

    # Caminho completo do script dentro da pasta sysfrozen caminho_script =
    os.path.join(pasta_sysfrozen, "limpeza_pastas.bat")

    # Criação do script de limpeza
    with open(caminho_script, 'w') as script_file: for pasta in
        self.pastas_selecionadas:
            script_file.write(f'del /q "{pasta}\\*"\n') script_file.write(f'FOR /D %%p IN
            ("{pasta}\\*.*") DO rmdir
"%%p" /s /q\n')
    MessageBox.Show(f"Script criado em: {caminho_script}", "Sucesso",
MessageBoxButtons.OK)
    return caminho_script
```

Agendamento Automático de Tarefas

A integração da ferramenta SYSFROZEN com o Agendador de Tarefas do Windows garante que o script de proteção e limpeza de pastas seja executado automaticamente durante eventos de logon, sem a necessidade de intervenção do usuário. Isso é possível graças à criação de uma tarefa agendada que roda com privilégios elevados, assegurando que o processo seja executado mesmo que o usuário não esteja conectado.

Este trecho de código usa o comando schtasks para agendar a tarefa, garantindo sua execução no momento em que o usuário faz logon, com privilégios elevados, através da conta SYSTEM. Isto automatiza a proteção, proporcionando uma camada extra de segurança e conveniência. O Código 4 mostra como esse agendamento foi implementado.

Código 4 – Agendamento da tarefa pelo sistema operacional

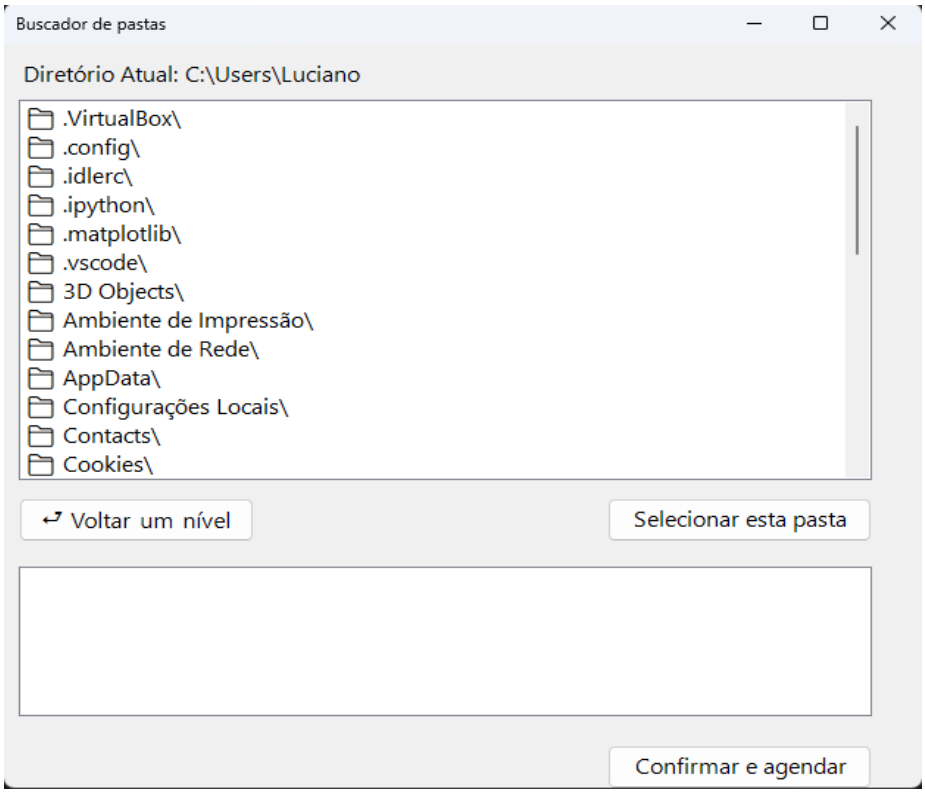
```
def agendar_tarefa(self, caminho_script):
  # nome_tarefa = "LimpezaPastas"
    # Comando para agendar a tarefa no Agendador de Tarefas para ser executada com
privilégios elevados
    # e independentemente de logon, usando a conta SYSTEM
    comando = f'schtasks /create /tn "{nome_tarefa}" /tr
"{caminho_script}" /sc onlogon /RL HIGHEST /RU SYSTEM /NP /f'
```

```
        try:
            # Executa o comando para criar a tarefa
            subprocess.run(comando, shell=True, check=True)
            MessageBox.Show(f"Tarefa agendada com sucesso para ser executada com
privilégios elevados ao fazer logon, mesmo sem usuário conectado.", "Sucesso",
MessageBoxButtons.OK)
        except subprocess.CalledProcessError as e:
            MessageBox.Show(f"Erro ao agendar a tarefa: {str(e)}", "Erro",
MessageBoxButtons.OK)
```

Execução da ferramenta

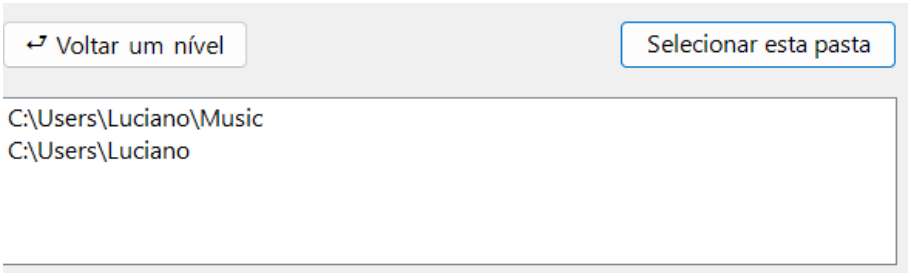
Ao iniciar a aplicação, será exibida uma interface gráfica, que apresenta um diretório navegador, com o caminho atual configurado para abrir sempre com a credencial do usuário que executou o programa. No caso em exemplo, o caminho configurado para abertura foi "C:\Users\Luciano", listando os diretórios e subdiretórios contidos nesse local, conforme pode ser visto na figura 1. A caixa para a seleção de diretórios, que pode ser observada na parte inferior da figura 1, é destacada na figura 2, e exibe o caminho completo da pasta atualmente selecionada, facilitando a confirmação da localização antes de gerar o script.

Figura 1 – Tela inicial do aplicativo



Fonte: dados da pesquisa.

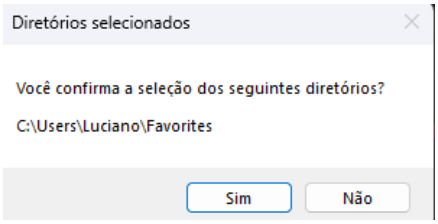
Figura 2 – Seleção de diretórios



Fonte: dados da pesquisa.

Caso o usuário não possua credenciais de acesso adequadas, o sistema apresentará uma mensagem de erro informando permissão insuficiente para o acesso. Estando as permissões corretas, uma janela de confirmação com a lista dos diretórios selecionados, conforme a Figura 3, é exibida.

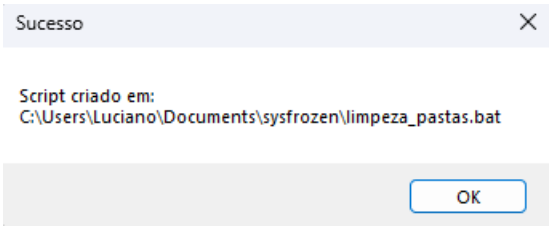
Figura 3 – Confirmação dos diretórios selecionados



Fonte: dados da pesquisa.

Estando tudo correto, o script será criado, e uma mensagem informando em que pasta o arquivo de script foi criado é exibida, conforme mostra a Figura 4.

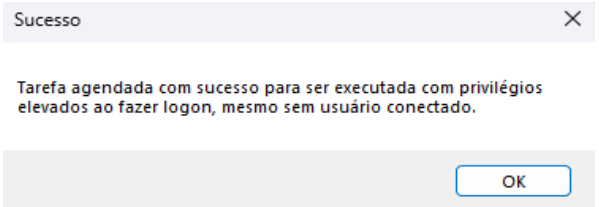
Figura 4 – Mensagem informativa sobre o local de criação do script



Fonte: dados da pesquisa.

Caso o usuário que executou o aplicativo tenha permissões a nível de administrador do sistema, uma mensagem de êxito no agendamento da tarefa será mostrada, conforme se vê na Figura 5.

Figura 5 – Mensagem de confirmação do agendamento da tarefa



Fonte: dados da pesquisa.

Testes Realizados

No laboratório de informática do Unifagoc foram conduzidos testes para avaliar a funcionalidade e a eficiência da ferramenta SYSFROZEN. O objetivo foi garantir que todas as funcionalidades do software operassem conforme o esperado em um ambiente controlado.

O software foi instalado em máquinas que atendem aos requisitos mínimos do sistema. Durante os testes, foram utilizados diferentes perfis de usuário, incluindo um usuário comum e um usuário administrador, para simular a experiência de diversos tipos de usuários.

A ferramenta foi iniciada sem problemas, permitindo que os usuários navegassem pelos diretórios de forma intuitiva. O processo de seleção de pastas ocorreu sem falhas, e a interface gráfica se mostrou responsiva e fácil de usar.

Foram realizados testes específicos para verificar as seguintes funcionalidades, apresentados a seguir.

Navegação nos Diretórios: O software permitiu a navegação pelos diretórios do sistema, mas a experiência variou entre os tipos de usuários. O usuário comum enfrentou restrições ao tentar acessar pastas de outros usuários e algumas pastas do sistema, devido à falta de permissões. Em contraste, o usuário administrador conseguiu navegar livremente, selecionando pastas para proteção.

Geração de Script: O sistema gerou automaticamente o script de limpeza conforme as pastas selecionadas, confirmando a funcionalidade desejada.

Agendamento de Tarefas: O agendador de tarefas foi configurado corretamente pelo usuário administrador, garantindo que a execução automática do script ocorra em eventos de logon.

Teste com Usuários

Os testes foram realizados com um usuário comum e um usuário administrador para avaliar como a ferramenta se comporta em diferentes contextos de permissão:

Usuário Comum: O usuário comum conseguiu navegar por suas próprias pastas e selecionar pastas para proteção. No entanto, ao tentar agendar a tarefa, o sistema retornou um erro devido à falta de permissões administrativas. Além disso, o usuário comum não pôde acessar pastas de outros usuários e algumas pastas do sistema, o que limitou a funcionalidade da ferramenta.

Usuário Administrador: O usuário administrador teve uma experiência similar na navegação e seleção de pastas, mas com a capacidade de agendar tarefas. O agendamento foi realizado com sucesso, garantindo que a execução do script ocorra em eventos de logon, mesmo sem o usuário estar conectado.

Resultados dos Testes

Os resultados mostraram que a ferramenta operou de maneira eficaz, cumprindo os objetivos propostos. Todos os testes foram bem-sucedidos, sem a ocorrência de erros críticos em funcionalidades acessíveis ao usuário comum. O feedback dos usuários foi positivo, destacando a simplicidade e a eficiência do SYSFROZEN, embora a necessidade de permissões administrativas para algumas funcionalidades e a limitação de acesso a pastas de outros usuários tenham sido

identificadas como considerações importantes

CONSIDERAÇÕES FINAIS

O desenvolvimento da ferramenta SYSFROZEN representa uma contribuição na proteção de pastas em ambientes Windows, utilizando uma abordagem simples e eficiente. Ao longo deste trabalho, foram abordados os principais aspectos da implementação e operação da ferramenta, demonstrando como ela foi projetada para atender às necessidades dos usuários que buscam maior segurança e facilidade na gestão de arquivos sensíveis.

A interface gráfica, construída com as bibliotecas System.Windows.Forms e System.Drawing, oferece uma experiência intuitiva para o usuário. A navegação pelos diretórios do sistema é facilitada, permitindo que os usuários selecionem facilmente as pastas que desejam proteger. O sistema é eficiente, gerando automaticamente um script de limpeza baseado nas pastas selecionadas, o que minimiza a necessidade de intervenção manual e maximiza a eficácia do processo.

Outro ponto crucial da ferramenta é sua integração com o Agendador de Tarefas do Windows, que garante a execução automática do script de proteção em eventos de logon. Isso proporciona uma camada adicional de segurança, permitindo que os usuários se sintam mais seguros em relação à proteção de seus dados, mesmo quando não estão ativamente monitorando suas pastas.

Comparada a outras ferramentas comerciais disponíveis no mercado, a SYSFROZEN se destaca pela sua simplicidade e eficiência. Muitas soluções comerciais podem ser complexas e exigem configurações extensivas, enquanto a SYSFROZEN foi desenvolvida com um foco na usabilidade, permitindo que usuários de diferentes níveis de habilidade possam proteger suas pastas sem dificuldades. A personalização e a automação proporcionadas pela ferramenta oferecem uma vantagem significativa em relação a abordagens mais tradicionais.

Em resumo, a SYSFROZEN não apenas atende às expectativas de proteção de dados, mas também redefine a forma como os usuários interagem com suas pastas no Windows. Através da combinação de uma interface amigável, automação de processos e integração com ferramentas nativas do sistema operacional, a SYSFROZEN se apresenta como uma solução robusta e eficaz para aqueles que buscam proteger informações críticas em seus computadores. A continuidade deste projeto pode abrir novas possibilidades para o aprimoramento e expansão das funcionalidades da ferramenta, garantindo que ela permaneça relevante em um cenário de ameaças em constante evolução.

REFERÊNCIAS

ANDERSON, Ross. **Security engineering**: a guide to building dependable distributed systems. Indianapolis, EUA: Wiley Publishing, 2020. Disponível em: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119644682. Acesso em: 23 maio 2024.

FARONICS CORPORATION. Deep Freeze. Disponível em: https://www.faronics.com/products/deep-freeze. Acesso em: 18 jun. 2024.

FITZPATRICK, M. Which Python GUI library should you use? **Python GUI Development Journal**, 2024.

GONZALEZ, M.; MARTINEZ, L. Python: a versatile tool for software development. **Software Development Journal**, 2019.

MITNICK, Kevin D.; WOZNIAK, Steve; SIMON, Willian L. **The art of deception**: controlling the human element of security. Indianapolis, EUA: Wiley Publishing, 2007. Disponível em https://www.academia.edu/1396282/The_art_of_deception. Acesso em: 20 maio 2024.

SAMADOV, Ismat. Automating tasks with Windows Task Scheduler and Python. **Python in Plain English,** Sep. 2023**.** Disponível em: https://pythoninplainenglish.com/automating-tasks-with-windows-task-scheduler-and-python. Acesso em: 18 jun. 2024.

SCHNEIER, Bruce. **Secrets and lies**: digital security in a networked world. Indianapolis, EUA: Wiley Publishing, 2000. Disponível em: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119183631. Acesso em: 5 maio 2024.

SMITH, J.; JOHNSON, R. Recent trends in cybersecurity threats: a comprehensive analysis. **Journal of Cybersecurity Research**, 2021.

TOOLWIZ SOFTWARE. **Time Freeze**. Disponível em: https://www.toolwiz.com/products/toolwiz-time-freeze. Acesso em: 18 jun. 2024