UNIVERSIDADE DA INTEGRAÇÃO INTERNACIONAL DA LUSOFONIA AFRO-BRASILEIRA

INSTITUTO DE ENGENHARIAS E DESENVOLVIMENTO SUSTENTÁVEL

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

MARCOS DOMINGOS SIMÃO KIACOLA

ANÁLISE DE VULNERABILIDADES COM COMANDO E CONTROLE EM SISTEMAS OPERACIONAIS WINDOWS: COMPARANDO OS RISCOS E AVALIANDO TÉCNICAS DE PERSISTÊNCIA

REDENÇÃO

2025

MARCOS DOMINGOS SIMÃO KIACOLA

ANÁLISE DE VULNERABILIDADES COM COMANDO E CONTROLE EM SISTEMAS OPERACIONAIS WINDOWS: COMPARANDO OS RISCOS E AVALIANDO TÉCNICAS DE PERSISTÊNCIA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto de Engenharias e Desenvolvimento Sustentável da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. André Luís da Costa Mendonça

REDENÇÃO

2025

Universidade da Integração Internacional da Lusofonia Afro-Brasileira
Sistema de Bibliotecas da UNILAB
Catalogação de Publicação na Fonte.

Kiacola, Marcos Domingos Simão.

K46a

Análise de vulnerabilidades com comando e controle em sistemas operacionais windows: comparando os riscos e avaliando técnicas de persistência / Marcos Domingos Simão Kiacola. - Redenção, 2025.
60f: il.

Monografia - Curso de Engenharia de Computação, Instituto De Engenharias E Desenvolvimento Sustentável, Universidade da Integração Internacional da Lusofonia Afro-Brasileira, Redenção, 2025.

Orientador: Prof. Dr. André Luís da Costa Mendonça.

1. Segurança da informação. 2. Vulnerabilidades. 3. Windows 10. 4. Windows 11. 5. Comando e Controle (C2). I. Título

CE/UF/BSCA CDD 005.8

MARCOS DOMINGOS SIMÃO KIACOLA

ANÁLISE DE VULNERABILIDADES COM COMANDO E CONTROLE EM SISTEMAS OPERACIONAIS WINDOWS: COMPARANDO OS RISCOS E AVALIANDO TÉCNICAS DE PERSISTÊNCIA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto de Engenharias e Desenvolvimento Sustentável da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. André Luís da Costa Mendonça (Orientador)
Universidade da Integração Internacional da Lusofonia Afro-Brasileira (UNILAB)

Prof. Dr. Jonas Costa Ferreira da Silva
Universidade da Integração Internacional da Lusofonia Afro-Brasileira (UNILAB)

Esp. Daniel Rezende de Barros
Laboratório de Sistemas e Bancos de Dados
(LSBD/UFC)

Dedico este trabalho ao meu pai Miguel Gaspar Kicola e minha mãe Marta Domingos Francisco, também para os meus irmãos por todo o apoio que me concederam em cada etapa desta jornada, especialmente para o meu filho Natanael Pereira Kiacola, espero que seja um incentivo para ele.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força e sabedoria concedida durante esta caminhada acadêmica. Aos meus pais, deixo minha mais profunda gratidão pelo amor incondicional, paciência e por acreditarem em meu potencial, mesmo quando os desafios pareciam maiores que os sonhos. Este trabalho é reflexo não apenas do meu esforço, mas também do apoio e da dedicação que sempre recebi de vocês.

De forma muito especial, agradeço ao meu tio Lourenço Domingos Martins, cujo aconselhamento constante e apoio foram essenciais em cada etapa desta jornada. Seus investimentos, compreensão e incentivo me deram tranquilidade e motivação para seguir firme. Sem a sua ajuda e confiança, esta conquista não teria sido possível.

Agradeço especialmente ao meu orientador, o Professor Dr. André Luís da Costa Mendonça, por sua orientação atenciosa e inspiradora. Expresso também o meu sincero reconhecimento pelo profissionalismo e pelos conselhos que ajudaram a transformar ideias em resultados concretos. Cada conversa, orientação e desafio compartilhado contribuíram para a construção deste trabalho. A clareza com que conduziu o processo de pesquisa e o compromisso com a qualidade acadêmica são exemplos que levarei comigo. Sou profundamente grato por suas ideias instigantes, pelas observações que ampliaram minha visão crítica e pelo incentivo constante, que transformaram esta experiência em um verdadeiro aprendizado. Sua sabedoria e dedicação foram determinantes não apenas para o desenvolvimento deste estudo, mas também para o meu crescimento pessoal e profissional.

Por fim, sou grato a todos que, de alguma forma, fizeram parte desta trajetória e contribuíram para esta conquista. Cada palavra de incentivo ou gesto de encorajamento e cada demonstração de confiança foram importantes para esta trajetória e me impulsionaram a seguir em frente para que eu conseguisse chegar até aqui. Levarei comigo a lembrança e a gratidão por todos que fizeram parte desta caminhada e ajudaram a transformar este sonho em realidade.

"Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas."

(Sun Tzu)

RESUMO

Diante do fim iminente do suporte ao Windows 10, em outubro de 2025, e sua persistente e expressiva base de usuários, este trabalho apresenta uma análise comparativa das vulnerabilidades entre o Windows 10 e o Windows 11. O trabalho se destaca por sua abordagem prática, que envolveu o desenvolvimento de uma ferramenta de comando e controle (C2) própria, implementada na linguagem Go. Para simular um vetor de ataque realista, o agente malicioso da ferramenta foi mascarado dentro de um instalador de software legítimo, utilizando a técnica de arquivo SFX. Os experimentos foram conduzidos em ambientes controlados, onde foram executados testes de exploração e exfiltração de dados, com foco especial em três técnicas de persistência: pasta de inicialização, registro do Windows e tarefas agendadas. Os resultados demonstraram que o Windows 10 apresenta fragilidades significativas, permitindo a persistência do agente através da criação de entradas no registro e do agendamento de tarefas. Em contraste, o Windows 11 demonstrou uma resiliência superior, bloqueando ativamente a tentativa de persistência por tarefas agendadas, embora a técnica via registro ainda tenha se mostrado eficaz. Por fim, o trabalho fornece evidências práticas que reforçam a migração para o Windows 11 não como uma simples atualização, mas como uma estratégia indispensável de gestão de riscos em segurança da informação.

Palavras-chave: Segurança da Informação. Vulnerabilidades. Windows 10. Windows 11. Comando e Controle (C2).

ABSTRACT

Given the imminent end of support for Windows 10 in October 2025, and its persistent and significant user base, this work presents a comparative analysis of vulnerabilities between Windows 10 and Windows 11. The work stands out for its practical approach, which involved the development of a custom command and control (C2) tool, implemented in the Go language. To simulate a realistic attack vector, the tool's malicious agent was masked within a legitimate software installer using the SFX file technique. The experiments were conducted in controlled environments, where exploitation and data exfiltration tests were performed, with a special focus on three persistence techniques: the startup folder, the Windows registry, and scheduled tasks. The experiments demonstrated that Windows 10 has significant weaknesses, allowing agent persistence through the creation of registry entries and scheduled tasks. In contrast, Windows 11 presented superior resilience, actively blocking the persistence attempt via scheduled tasks, although the registry technique still proved effective. Finally, the work provides practical evidence that reinforces the migration to Windows 11 not as a simple update, but as an indispensable risk management strategy in information security.

Keywords: Information Security. Vulnerabilities. Windows 10. Windows 11. Command and Control (C2).

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

C2	Comando e Controle
CVE	*Common Vulnerabilities and Exposures*
CVSS	*Common Vulnerability Scoring System*
CWE	*Common Weakness Enumeration*
DDoS	*Distributed Denial of Service*
DLL	*Dynamic Link Library*
DoS	*Denial of Service*
EoP	Elevação de Privilégio
HID	*Human Interface Device*
LOLBins	*Living Off the Land Binaries*
LSASS	*Local Security Authority Subsystem Service*
MaaS	*Malware as a Service*
MFA	*Multi-Factor Authentication*
RCE	Execução de Código Remoto
RDP	*Remote Desktop Protocol*
SFX	*Self-Extracting File*
SMBv1	*Server Message Block*
SO	Sistema Operacional
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TPM 2.0	*Trusted Platform Module*
VBS	*Virtualization Based Security*
VM	Máquina Virtual
WDAC	*Windows Defender Application Control*

SUMÁRIO

1 INTRODUÇÃO

A crescente digitalização da sociedade, impulsionada pelo avanço das tecnologias da informação e comunicação, tornou os sistemas operacionais elementos centrais na rotina de usuários domésticos, empresas e instituições governamentais. Nesse cenário, os sistemas operacionais Windows aceleraram a dependência de sistemas operacionais confiáveis, por serem amplamente adotados globalmente e desempenharem um papel estratégico tanto no suporte à infraestrutura tecnológica quanto na segurança dos dados processados em domicílios.

Com milhões de usuários no mundo todo, os sistemas operacionais Windows tornaram-se a espinha dorsal de muitas infraestruturas de TI. O Windows 10, lançado em julho de 2015, trouxe melhorias significativas em compatibilidade e desempenho em relação aos seus antecessores, sendo concebido como uma plataforma unificadora, capaz de rodar em desktops, notebooks, tablets e até em dispositivos embarcados. No entanto, manteve diversas vulnerabilidades herdadas de versões anteriores. Falhas em segurança da informação comprometem dados sensíveis, interrompem operações e podem levar a prejuízos diretos (como em ataques de *ransomware* que exigem resgate financeiro) ou indiretos (como a perda de confiança dos clientes e multas regulatórias no caso de vazamento de dados pessoais). Dessa forma, é importante destacar que a segurança não é um produto, mas um conjunto de decisões coerentes, repetidas com disciplina ao longo do tempo. No Windows, isso significa combinar princípios sólidos, configurações cuidadosas, atualizações contínuas e transparência do que acontece nas máquinas e na rede.

Por sua vez, o Windows 11, disponibilizado em outubro de 2021, representa uma ruptura arquitetural ao elevar a linha de base de segurança. Com o aumento das ameaças digitais, os sistemas operacionais passaram a incorporar mecanismos de segurança nativos. O Windows 11, por exemplo, exige o *Trusted Platform Module* (TPM 2.0) e *Secure Boot* como pré-requisitos de instalação, garantindo que apenas softwares confiáveis sejam executados durante a inicialização (MICROSOFT, 2025e). Além disso, o recurso *Virtualization Based Security* (VBS) isola processos privilegiados em ambientes virtualizados, dificultando a escalação de privilégios por *malwares* e restringindo elevações administrativas (MICROSOFT, 2024) (MICROSOFT, 2025b).

No entanto, apesar das atualizações dos sistemas operacionais Windows, as versões mais antigas continuam sendo utilizadas de maneira massiva pelos usuários. Segundo dados da Statcounter (STATCOUNTER, 2025), em setembro de 2025, a participação de usuários do

Windows 11 era de 52,9% contra 43,78% de usuários do Windows 10, enquanto as versões mais antigas completavam o quadro. Isso acaba soando como um alerta, dado que versões mais antigas tendem a se tornar obsoletas após um período de tempo e passam a não mais receber atualizações. O não recebimento de atualizações é extremamente crítico, uma vez que vulnerabilidades já conhecidas podem não ser corrigidas, além de permitir que novas vulnerabilidades sejam exploradas cada vez mais por grupos de usuários maliciosos.

Diversas organizações e usuários individuais se deparam com um cenário preocupante: a vulnerabilidade crescente de seus sistemas operacionais. A ausência de atualizações de segurança periódicas transforma o sistema em um alvo atrativo para cibercriminosos. Historicamente, versões descontinuadas do Windows, como o Windows XP, Windows Vista, Windows 7, Windows 8 e Windows 8.1 sofreram um aumento significativo de ataques após o fim do suporte oficial, especialmente por *exploits* públicos que deixam de ser corrigidos. O cenário se torna ainda mais crítico com o anúncio oficial do fim do suporte ao Windows 10, previsto para outubro de 2025 (MICROSOFT, 2025c; MICROSOFT, 2025d). O Windows 10 acumulou, ao longo de seu ciclo de vida útil, uma série de vulnerabilidades conhecidas e catalogadas em bancos de dados, como o CVE (*Common Vulnerabilities and Exposures*). Mesmo com atualizações frequentes, muitas falhas foram exploradas repetidamente por agentes maliciosos devido à ausência de políticas de atualização eficazes em diversas organizações. Entre as vulnerabilidades críticas, destacam-se aquelas relacionadas ao protocolo *Server Message Block* (SMBv1), macros em documentos do Microsoft Office e permissões mal configuradas por usuários. Em um ecossistema no qual o Windows segue majoritário (PROCURRI, 2025), falhas exploráveis em versões ativas, combinadas com o atraso na aplicação de patches de correção e configurações permissivas demais, ampliam o risco de ataques e invasões bem-sucedidas.

Diante dessa problemática, este trabalho analisa algumas vulnerabilidades presentes no Windows 10 e fraquezas no Windows 11, além de propor um ataque aos sistemas operacionais Windows, com o objetivo de comparar o comportamento de vulnerabilidades no Windows 10 e Windows 11. O estudo também apresenta por que os ataques funcionam com regularidade no Windows 10 e como esses riscos são mitigados no Windows 11. Por fim, desejamos auxiliar tanto profissionais de TI quanto indivíduos casuais a tomarem decisões estratégicas para a proteção de ambientes computacionais.

1.1 Objetivos

Os objetivos geral e específicos são descritos a seguir.

1.1.1 Objetivo Geral

Realizar uma análise comparativa de vulnerabilidades e mecanismos de segurança dos sistemas operacionais Windows 10 e Windows 11 por meio de experimentos práticos conduzidos com uma ferramenta própria de Comando e Controle (C2).

1.1.2 Objetivos Específicos

- Desenvolver e implementar uma ferramenta de comando e controle (C2) para simular cenários de ataque e persistência.
- Implementar três técnicas de persistência: Startup Folder, Registro do Windows e Tarefa Agendada.
- Avaliar a eficácia dos mecanismos de segurança do Windows 11 em comparação com o Windows 10.
- Sugerir boas práticas de segurança baseadas nos resultados obtidos.

1.2 Estrutura do Trabalho

O restante deste trabalho está estruturado da seguinte forma:

- O Capítulo 2 apresenta a fundamentação teórica, abordando conceitos sobre sistemas operacionais, redes de computadores e segurança da informação.
- O Capítulo 3 reúne os trabalhos relacionados, destacando os estudos anteriores sobre vulnerabilidades e ataques em sistemas operacionais Windows.
- O Capítulo 4 descreve a metodologia adotada, incluindo o desenvolvimento e o funcionamento da ferramenta de comando e controle (C2) proposta.
- O Capítulo 5 apresenta os resultados obtidos a partir dos testes realizados nos sistemas operacionais Windows 10 e Windows 11.
- O Capítulo 6 traz as conclusões, discutindo os principais resultados obtidos durante o desenvolvimento do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, exploraremos os conceitos e teorias fundamentais que são essenciais para a compreensão do tema em estudo. Aqui, abordaremos as principais definições relevantes que formam a base deste trabalho.

2.1 Sistemas Operacionais

Um Sistema Operacional (SO) coordena recursos de hardware e software, gerencia processos, memória, arquivos de dispositivos e oferece serviços para que outras aplicações sejam executadas com integridade. Em termos práticos, os SOs são responsáveis por gerenciar os recursos computacionais, permitindo que o hardware e os softwares se comuniquem. De acordo com (TANENBAUM; BOS, 2015), o SO atua como uma camada de abstração entre o usuário e o hardware, controlando processos, memória, dispositivos de entrada e saída (E/S), além de oferecer mecanismos de segurança e isolamento. Em termos práticos, o SO decide quem usa o quê, quando e como. Essa mediação é crítica porque qualquer brecha pode levar a uma execução não autorizada, elevação de privilégio e persistência nas máquinas. Dentre os sistemas operacionais modernos da Microsoft, o Windows 10 e o Windows 11 são os mais utilizados na atualidade (STATCOUNTER, 2025).

No contexto dos sistemas operacionais Windows, a evolução dos sistemas é particularmente evidente. O Windows 10, lançado há uma década, trouxe melhorias significativas em compatibilidade e desempenho em relação aos seus antecessores, mas manteve diversas vulnerabilidades herdadas de versões anteriores. Em cibersegurança, o sistema operacional é a peça chave na proteção dos dados, já que é responsável por aplicar políticas de acesso, autenticação e integridade das informações.

Com o aumento das ameaças digitais, os sistemas operacionais passaram a incorporar mecanismos de segurança nativos. O Windows 11, por exemplo, exige TPM 2.0 e *Secure Boot* como pré-requisitos de instalação, garantindo que apenas softwares confiáveis sejam executados durante a inicialização. Além disso, o recurso VBS isola processos privilegiados em ambientes virtualizados, dificultando a escalação de privilégios por *malwares* e restringindo elevações administrativas (MICROSOFT, 2025b).

2.2 Redes de Computadores

(TANENBAUM; WETHERALL, 2010) definem rede de computadores como um conjunto de dispositivos interligados por meios físicos ou lógicos, que compartilham dados, recursos e serviços de forma estruturada. Essa definição revela que redes não são apenas cabos e roteadores, mas sim sistemas que precisam ser planejados e mantidos de forma segura, visto que, quando os serviços ficam expostos, sem filtragem ou com autenticação fraca, a rede acaba se tornando uma via de trânsito para o atacante.

Considere uma rede de computadores como uma cidade digital, onde os dados trafegam como carros em avenidas invisíveis. As redes de computadores são como avenidas digitais por onde trafegam dados, comandos e conexões. Mas, como qualquer via, elas podem ter buracos, desvios e pontos cegos. Os buracos em redes são falhas que permitem que agentes maliciosos interceptem, modifiquem ou bloqueiem esse tráfego. Para que essa comunicação ocorra de forma eficiente, são utilizados protocolos de rede, os quais funcionam como regras de trânsito digital. O modelo mais conhecido é o *Transmission Control Protocol/Internet Protocol* (TCP/IP), que organiza a comunicação em camadas, desde o nível físico (cabos e sinais) até o nível de aplicação (navegadores, e-mail, etc). Embora a pilha TCP/IP estruture a comunicação em camadas, alguns riscos emergem de combinações de má configuração, software desatualizado e credenciais fracas.

2.3 Segurança da Informação

Segundo (HINTZBERGEN; HINTZBERGEN, 2015), a segurança da informação envolve tanto práticas quanto tecnologias voltadas a proteger dados contra diferentes tipos de ameaças, desde acessos não autorizados até a perda de integridade ou interrupções nos serviços.

A segurança em redes e sistemas computacionais não é um destino, mas uma jornada contínua feita de escolhas repetidas com disciplina e conhecimento. Nessa jornada, entender as vulnerabilidades é o primeiro passo para construir ambientes mais resilientes e confiáveis. Mais do que *firewalls* e antivírus, a segurança digital exige consciência coletiva, bons hábitos e costumes, visto que cada ação indevida pode ser o início de um incidente.

A confidencialidade, integridade e disponibilidade formam a base essencial que sustenta a segurança da informação, complementada por autenticidade e responsabilidade (HINTZBERGEN; HINTZBERGEN, 2015). Na prática, isso se traduz em política de acesso, criptografia,

controle de auditoria, monitoramento e resposta a incidentes. Os pilares da segurança da informação representam os princípios básicos que orientam a proteção dos dados em qualquer organização. As definições desse tripé são apresentadas a seguir.

Confidencialidade: Garante que a informação seja acessada somente por pessoas autorizadas, aplicando o nível de sigilo necessário em cada elemento de processamento de dados e impedindo a divulgação não autorizada.

Integridade: Assegura que as informações permaneçam em seu estado original, completas e sem alterações maliciosas ou acidentais. A corrupção, modificação maliciosa ou substituição de dados por informações incorretas comprometem a integridade.

Disponibilidade: Garante que os usuários autorizados tenham acesso às informações e sistemas sempre que necessário. Falhas em dispositivos ou softwares, ataques de negação de serviço e fatores ambientais podem comprometer a disponibilidade

Além desses princípios fundamentais, a segurança da informação abrange outros requisitos, como autenticidade (garantir que a identidade das partes envolvidas seja verdadeira) e não repúdio (assegurar que uma transação ou comunicação não possa ser negada por uma das partes). Cada um desses aspectos é vital para proteger os ativos informacionais em uma organização ou sistema computacional.

2.4 Vulnerabilidades

Vulnerabilidade é uma fraqueza técnica, processual ou humana que pode ser explorada para violar uma propriedade de segurança ou causar impacto operacional. Em sistemas operacionais, as vulnerabilidades surgem em serviços de rede, *drivers*, *kernel*, mecanismos de autenticação, carregamento de *Dynamic Link Library* (DLL) e políticas de execução. Em um estudo prático com o framework Metasploit em ambientes educacionais, a exploração de falhas conhecidas em sistemas operacionais Windows evidenciou o acesso a arquivos sensíveis e a manipulação do sistema quando *patches* e proteções estavam ausentes (SOBRINHO, 2024).

As vulnerabilidades podem ser classificadas como:

Vulnerabilidades de Software: Bugs no código-fonte de sistemas operacionais e aplicativos.

Vulnerabilidades de Configuração: Permissões excessivas, portas abertas e ausência de políticas de acesso;

Vulnerabilidades Humanas: Engenharia social, *phishing* e falhas na atualização de

sistemas.

Vulnerabilidades Físicas: Acesso não autorizado a hardware ou mídias de armazenamento.

Os catálogos *Common Vulnerabilities and Exposures* (CVE) (CVE Program, 2025) e *Common Weakness Enumeration* (CWE) (The MITRE Corporation, 2025) são responsáveis por organizar falhas e classes de vulnerabilidades. Por sua vez, o catálogo *Common Vulnerability Scoring System* (CVSS) (FIRST.org, Inc., 2025) fornece uma noção da severidade das vulnerabilidades catalogadas. Na prática, o que decide a prioridade de uma vulnerabilidade é o risco, entendido como a combinação do impacto *versus* a probabilidade, acrescida da responsabilidade técnica de segurança individual no ambiente.

Em ambientes Windows, as vulnerabilidades frequentemente decorrem de configurações padrão inseguras ou frágeis, como permissões excessivas e autenticação baseada apenas em senha, falhas de software e serviços expostos desnecessariamente, e uso de *drivers* vulneráveis. Um exemplo é o protocolo SMBv1, cuja exploração permitiu ataques massivos, como o *ransomware* WannaCry (JÚNIOR, 2023).

2.4.1 Principais Vulnerabilidades no Windows 10

O Windows 10, lançado oficialmente em 2015, teve um longo ciclo de vida, recebendo constantes atualizações e *patches* de segurança por uma década. Contudo, ao longo dos anos, um grande número de vulnerabilidades foram descobertas e exploradas em suas várias versões, desde a sua versão inicial (1507) até a sua versão final (22H2). Aqui, destacaremos algumas das principais vulnerabilidades que afetam o sistema operacional Windows 10, juntamente com alguns exemplos.

- **Execução de Código Remoto (RCE):** São talvez as vulnerabilidades mais críticas, pois permitem que um atacante, de forma remota, execute comandos arbitrários no sistema sem autorização. O Windows 10 herdou algumas fragilidades de componentes presentes em versões anteriores do Windows (como o Windows 7, Windows 8 e Windows 8.1). Um exemplo marcante é a falha EternalBlue (MS17-010) no protocolo SMBv1. Além do EternalBlue, estudos sobre vulnerabilidades no Windows 10 Pro (1903) indicaram que várias falhas graves do sistema estavam concentradas justamente no serviço de *Remote Desktop Protocol* (RDP), as quais já foram corrigidas em 2020 pela Microsoft, mas que antes permitiam a RCE não autenticada.

- **Elevação de Privilégio (EoP):** Essas vulnerabilidades permitem que um usuário local comum obtenha privilégios de "Administrador" ou até mesmo de "Sistema". Essas falhas são muito valorizadas por atacantes, pois permitem a aplicação de controles de persistência em programas maliciosos. Houve casos de EoP explorando desde os componentes do *kernel* (como falhas em *drivers* do sistema) até os utilitários (como serviços de agendamento de tarefas). Por conta desse tipo de vulnerabilidade, a Microsoft emite *patches* quase todos os meses para corrigir alguma brecha de elevação. Portanto, um sistema não atualizado pode facilmente sucumbir a *exploits* locais publicamente disponíveis.
- ***Bypass* dos Mecanismos de Segurança:** Esse tipo de vulnerabilidade permite que um atacante contorne as proteções de segurança do sistema operacional, como o antivírus, ou outras medidas que impedem o acesso ou a execução não autorizada. Ao longo do tempo, os pesquisadores encontraram maneiras de burlar as proteções integradas do Windows 10, como o *sandbox* AppContainer, o Device Guard e até mesmo o *Windows Defender Application Control* (WDAC). Por exemplo, métodos de execução de código via *Living Off the Land Binaries* (LOLBins), que utilizam binários legítimos do sistema para fins maliciosos, não são vulnerabilidades no sentido tradicional (não geram CVE), mas exploram funcionalidades do Windows 10. Um atacante no Windows 10 poderia utilizar ferramentas legítimas, como o `powershell.exe`, `mshta.exe` ou `rundll32.exe`, para contornar as restrições de execução de software impostas pelo administrador, caso não houvesse políticas estritas. Assim, enquanto o Windows 10 oferece recursos de segurança, sua eficácia depende das configurações que, quando fracas ou padrão, podem ser insuficientes.

Uma das grandes qualidades do Windows 10 é a sua flexibilidade quanto à adoção em diferentes cenários, formados por hardware e perfis de usuários distintos. No entanto, essa flexibilidade também se torna um risco quando somamos três fatores: os recursos legados que permanecem habilitados, as políticas de segurança que não entram em vigor e as atualizações irregulares (SOBRINHO, 2024).

2.5 Ameaças

Ameaças são elementos que podem explorar vulnerabilidades e causar problemas aos sistemas de informação. Conforme (WHITMAN *et al.*, 2009), as ameaças podem ser classificadas como:

Naturais: Causadas por fenômenos da natureza (incêndios, inundações, entre outros). Embora não explorem vulnerabilidades de software em si, enquadram-se na gestão de segurança pela potencial indisponibilidade ou destruição de ativos que elas causam.

Involuntárias: Causadas por desconhecimento, acidentes ou erros por negligência, mas sem a intenção deliberada de causar dano. Dentre os possíveis exemplos, podemos citar um administrador que configura incorretamente um servidor, expondo-o indevidamente, falhas de software não intencionais ou até a ação de um funcionário despreparado que abre um anexo de e-mail malicioso sem saber ou por curiosidade.

Voluntárias: São propositais, causadas por hackers mal intencionados, criadores e disseminadores de vírus, ou agentes com motivação clara para causar dano. Dentre os possíveis exemplos, podemos citar um invasor que ataca um sistema para roubar informações e credenciais ou derrubar um serviço.

No âmbito deste trabalho, daremos ênfase às ameaças voluntárias, oriundas de agentes maliciosos, pois estas são as que tipicamente exploram vulnerabilidades técnicas em sistemas operacionais.

2.5.1 Malwares

Dentre as ameaças voluntárias, os *malwares* estão entre as mais comuns e persistentes em sistemas operacionais. A palavra *malware* deriva de "software malicioso", ou seja, é um termo genérico para programas maliciosos, desenvolvidos com o objetivo de causar danos, roubar informações ou obter controle indevido de sistemas. Uma vez instalado, o *malware* tende a operar de forma silenciosa, muitas vezes sem que o usuário perceba sua presença, buscando persistência e evasão. Em geral, trata-se de um software criado com a intenção de prejudicar uma organização ou um indivíduo, danificando softwares ou quebrando uma infraestrutura de redes de computadores e sistemas computacionais (STALLINGS, 1995).

No Windows, esses softwares podem se infiltrar por meio de anexos de e-mails, downloads em sites não oficiais, dispositivos removíveis infectados, macros de documentos e por explorações de vulnerabilidades não corrigidas em sistemas operacionais (MICROSOFT, 2025a). Por exemplo, o Crocodilus é *malware* que observa os hábitos das vítimas para atacar na hora certa e roubar credenciais bancárias, chaves de criptomoedas e até códigos 2FA, resultando em perdas financeiras e dados expostos e vulneráveis na internet. Por sua vez, o FormBook, detectado pela primeira vez em 2016, é um *infostealer* para o Windows vendido como um

Malware as a Service (MaaS), cujo propósito consiste em roubar credenciais de navegadores. O FormBook tem sido utilizado em diversas campanhas de *phishing* e spam de e-mail, muitas vezes embutido em documentos do Microsoft Office ou em arquivos compactados. Sua versatilidade e constantes atualizações o mantêm como uma ameaça ativa e relevante, especialmente em ambientes corporativos e instituições governamentais, mas também é utilizado para atacar usuários comuns.

2.5.1.1 Principais Famílias de Malwares

Os *malwares* continuam sendo um dos vetores mais utilizados em ataques, sobretudo em ambientes que utilizam sistemas operacionais Windows, visto que apresentam um maior número de usuários e, portanto, maior atratividade para cibercriminosos (KASPERSKY, 2025). Cada tipo de *malware* tem um propósito específico, e entender suas diferenças é essencial para combatê-los. A diversidade de *malwares* é tão grande quanto a sua capacidade de adaptação. A seguir, apresentamos os principais tipos existentes.

2.5.1.1.1 Vírus

Os vírus se anexam a arquivos ou programas legítimos e são ativados apenas quando esses hospedeiros são executados. A sua replicação e propagação dá-se através de compartilhamentos. Os vírus corrompem ou apagam dados e permanecem ocultos até o momento da ativação, pois dependem de um hospedeiro para serem executados, tornando-os um dos tipos mais antigos de *malware*, sendo amplamente utilizados em diferentes tipos de ataques hacker até os dias atuais.

Em 1971, nasceu o Creeper, o primeiro vírus da história, criado por Bob Thomas como um experimento na ARPANET. Seu comportamento consistia apenas em se replicar entre computadores e mostrar a frase: "*I'M THE CREEPER. CATCH ME IF YOU CAN!*". Embora inofensivo na época, abriu caminhos para o surgimento dos *worms* e, mais tarde, dos principais *malwares* que conhecemos até hoje.

2.5.1.1.2 *Worm*

Os *worms* atuam na busca por falhas, tais como portas abertas ou protocolos inseguros, além de abrirem caminho para outros *malwares*. Tais características fazem com que os

worms necessitem de um maior consumo de banda (mais pacotes nas redes), o que pode resultar na lentidão do sistema. São mais independentes do que os vírus e espalham-se por redes sem precisar de um hospedeiro, sendo capazes de se autorreplicar, mesmo sem nenhuma interação.

Em 2004, surgiu o Sasser, um *worm* que explorava uma falha no *Local Security Authority Subsystem Service* (LSASS) do Windows. À época, ele se espalhou em alta velocidade, atingindo sistemas como Windows 2000, Windows XP e Windows Server 2003, em apenas duas semanas após a divulgação da vulnerabilidade, tendo sido considerado um dos ataques mais rápidos da história.

2.5.1.1.3 *Trojan* (Cavalo de Tróia)

Os *trojans* se disfarçam de programas legítimos ou úteis e enganam o usuário para que sejam instalados. Diferente dos vírus e *worms*, os cavalos de tróia não se replicam sozinhos, mas, uma vez instalados ou ativos, podem abrir portas do sistema para invasores (*backdoors*) e executar outras ações maliciosas, como roubar dados ou permitir controle remoto do sistema. Por conta de suas características, são amplamente utilizados em ataques direcionados e espionagem corporativa.

Em 2000, o *trojan* ILOVEYOU espalhou-se pelo mundo via e-mail com o assunto "ILOVEYOU". Por conta do título, milhares de pessoas abriam o anexo acreditando ser uma carta de amor, enquanto, na verdade, tratava-se de um anexo malicioso que apagava arquivos e se compartilhava com todos os contatos na lista de endereços da vítima. Até hoje, é considerado um dos ataques mais devastadores da história da internet.

Por sua vez, o Qbot, também conhecido como Qakbot, é um tipo de cavalo de tróia especializado em roubar dados bancários e financeiros. Identificado pela primeira vez em 2008, ele se espalha, primariamente, através de spam de e-mail, além de utilizar técnicas sofisticadas para evitar a sua detecção.

2.5.1.1.4 *Ransomware*

O *ransomware* consiste em um tipo de *malware* que criptografa as informações ou bloqueia o dispositivo e exige o pagamento de um resgate (OLIVEIRA, 2018). Considere, por exemplo, que você perca o acesso a todos os seus arquivos e seja obrigado a pagar para recuperá-los, é justamente isso que o *ransomware* faz. Em algumas variantes, ele também pode bloquear completamente o sistema, combinando cifragem com exfiltração para realizar

a extorsão. Geralmente, os hackers exigem pagamentos em criptomoedas para desbloquear os arquivos ou dispositivos. No entanto, não há nenhum tipo de garantia de que, mesmo após o pagamento, o acordo será cumprido. Esse tipo de ataque cresceu tanto em frequência quanto em sofisticação nos últimos anos, utilizando-se de técnicas de evasão avançadas e aproveitando-se de falhas conhecidas em sistemas operacionais desatualizados, atingindo desde usuários domésticos até grandes corporações e órgãos públicos (FILHO; FREITAS, 2023).

Em 2017, o *ransomware* WannaCry mostrou como esses tipos de *malwares* podem paralisar hospitais, empresas e governos. O WannaCry foi um *ransomware* que combinou a exploração da vulnerabilidade EternalBlue (CVE-2017-0144) no SMBv1 do Windows com a propagação automática em rede, causando um ataque global que afetou milhares de máquinas que executavam Windows 7, Windows 8, Windows 8.1 e até mesmo Windows 10 desatualizados (sem *patches* de atualização). A Figura 1 apresenta a tela que é exibida por um dispositivo infectado pelo *ransomware* WannaCry. É possível observar as instruções e as exigências de pagamento em criptomoedas para que os dados sejam liberados.

Figura 1 – *Ransomware* WannaCry.

Fonte: (OLIVEIRA, 2018).

2.5.1.1.5 *Adware*

Embora sejam menos perigosos, os *adwares* são um tipo de *malware* que exibem anúncios indesejados e podem coletar dados de navegação para fins publicitários, como o

monitoramento de hábitos e costumes do usuário. Os *adwares* não chegam a ser considerados maliciosos, mas exibem anúncios que rastreiam os *cookies* de navegação. Além disso, podem ser incômodos, degradando a experiência do usuário em alguns casos, e podem ser utilizados para aumentar a superfície de ataque para ameaças mais graves, servindo de porta de entrada para outros *malwares*.

O Sendori é um software bastante conhecido que prometia "corrigir" URLs digitadas de maneira errada e direcionar o usuário ao site correto. No entanto, na prática, era um *adware* que se instalava junto de programas gratuitos e redirecionava a navegação do usuário para outros links, como anúncios, propagandas enganosas ou até mesmo fraudes, tudo sem o consentimento do usuário.

2.5.1.1.6 *Spyware*

O *spyware* consiste em um espião digital que coleta informações, como credenciais, histórico de navegação e dados sensíveis de maneira oculta. Pode operar por meses sem ser detectado, comprometendo a privacidade do usuário e facilitando fraudes e falsificações de informações. Uma vez realizada a invasão, a sua detecção é muito difícil, e sua duração pode perdurar por bastante tempo.

Em algumas ocasiões, algumas entidades utilizam *spywares* para fins de espionagem, sob a justificativa de serem uma ferramenta aliada no combate ao crime. Nesse contexto, o Pegasus, criado pela israelense NSO Group, é um *spyware* capaz de invadir sistemas sem ser detectado. Embora seja vendido a governos sob a justificativa de combater crimes e terrorismo, o software já foi denunciado pelo uso em abusos contra jornalistas, ativistas e opositores políticos.

Ainda no contexto de *spywares*, os *keyloggers* e *screenloggers* são tipos específicos de *spywares*. Como mencionado, os *spywares* são uma ampla categoria de *malwares* projetados para monitorar as atividades de um sistema e enviar as informações coletadas para terceiros sem o conhecimento ou consentimento do usuário. O que difere um *keylogger* de um *screenlogger* é o tipo de informação que será capturada do usuário e enviada ao agente malicioso. Enquanto os *keyloggers* capturam tudo o que é digitado pelo usuário, os *screenloggers* capturam imagens e vídeos do dispositivo do usuário.

2.5.1.1.7 *Rootkit*

O *rootkit* não é exatamente um tipo de *malware* autônomo, mas um conjunto de ferramentas automatizadas que exploram vulnerabilidades conhecidas em sistemas ou aplicativos, frequentemente entregando um *malware* no final. Eles operam em níveis profundos do sistema, alterando o *kernel* ou bibliotecas críticas para ocultar a presença de outros *malwares*. São muito difíceis de serem detectados e permitem que outros *malwares* se escondam, tornando a remoção deles extremamente complexa ou, até mesmo, impossível sem restauração ou backups.

2.5.1.1.8 *Bots* e *Botnets*

Um *bot* consiste em um *malware* que coloca o computador sob o comando e controle remoto de um atacante. Múltiplos *bots* coordenados formam uma *botnet*, uma rede zumbi de computadores escravizados que pode receber ordens de um servidor (comando e controle). As *botnets* são utilizadas, principalmente, para ataques coordenados de negação de serviço, como os ataques *Denial of Service* (DoS) e sua variante distribuída, o *Distributed Denial of Service* (DDoS). Além dos ataques de negação de serviço, as *botnets* podem ser utilizadas em atividades de envio de spam em larga escala.

2.6 Ataques

De maneira geral, um ataque consiste em qualquer ação intencional que visa comprometer a segurança de um sistema, violando sua política de segurança ou explorando alguma fraqueza para obter acesso indevido, alterar dados e degradar o funcionamento operacional da infraestrutura.

Com base na ABNT NBR ISO/IEC 27032 (ABNT – Associação Brasileira de Normas Técnicas, 2024), um ataque cibernético é uma ação intencional realizada por indivíduos, grupos ou organizações com o objetivo de explorar vulnerabilidades em sistemas de informação, redes de computadores ou infraestruturas digitais, visando comprometer a confidencialidade, integridade ou disponibilidade desses recursos, causando danos, interrupções e obtendo benefícios ilícitos.

Os tipos de ataques cibernéticos que exploram as vulnerabilidades nos sistemas computacionais podem ser classificados quanto ao seu efeito nos sistemas. Os ataques podem ser classificados como: diretos, indiretos, ativos, passivos, internos ou externos (PFLEEGER;

PFLEEGER, 2002; TANENBAUM; BOS, 2015).

Os ataques diretos ocorrem contra os alvos principais, como, por exemplo, ao explorar serviços expostos na Web. Por sua vez, os ataques indiretos utilizam terceiros, engenharia social, cadeias de suprimentos ou pivôs internos, como recursos de infraestrutura, para abrir caminhos até o alvo final e alcançar o objetivo.

Os ataques ativos utilizam técnicas para alterar, de alguma forma, o estado do sistema ou dos dados em trânsito, afetando o funcionamento normal do sistema. Os ataques ativos geralmente deixam rastros devido à perturbação causada, mas podem ser devastadores. Os ataques ativos subdividem-se em categorias, como:

- ***Masquerade* (falsificação de identidade):** O atacante se passa por outra entidade autorizada.
- **Replay:** Repete mensagens ou transações capturadas previamente para ganho indevido.
- **Modificação de Dados:** Altera dados legítimos.
- **Negação de Serviço (DoS):** Indisponibiliza os recursos dos sistemas.

Por exemplo, um *ransomware* que criptografa arquivos é um ataque ativo de modificação de dados. Um *worm* que se espalha e derruba redes também pode ser considerado um ataque ativo de DoS.

Por sua vez, os ataques passivos são aqueles em que o atacante observa ou monitora as comunicações na rede e os sistemas operacionais. O objetivo é coletar informações sigilosas sem ser detectado, nem tentar alterar o estado ou degradar serviços. Um exemplo muito famoso de ataque passivo é o *sniffing* de pacotes, com o objetivo de monitorar e interceptar o tráfego de dados na rede para capturar informações sensíveis e sigilosas. Como esses ataques não causam alterações, eles são discretos e difíceis de detectar.

Já os ataques internos são realizados por alguém dentro do perímetro de segurança, ou seja, por quem já tem acesso ao sistema ou ao local físico da infraestrutura, como um funcionário, prestador de serviço ou qualquer pessoa com acesso à rede interna. Esses ataques podem utilizar privilégios legítimos para fins maliciosos ou explorar sistemas sem as barreiras dos *firewalls* externos. Por exemplo, um empregado insatisfeito que explora uma vulnerabilidade em um servidor interno e instala um *malware* na rede é considerado um ataque interno por abuso de privilégio.

Por fim, os ataques externos são aqueles conduzidos por agentes que não têm acesso autorizado à rede ou ao sistema alvo, partindo de fora do perímetro de segurança. Em outras

palavras, esses ataques vêm através da internet, a partir de outras conexões externas, por meio da exploração de serviços expostos na Web. Por exemplo, um hacker que explora uma porta aberta no *firewall* da empresa se enquadra nesse tipo de ataque. Em ambientes Windows, um ataque externo comum seria a exploração de um serviço exposto indevidamente por meio do RDP, diretamente na rede.

3 TRABALHOS RELACIONADOS

Neste capitulo, iremos revisar alguns dos trabalhos mais recentes e relevantes na literatura que focam no estudo e análise de vulnerabilidades em sistemas.

3.1 Análises de Vulnerabilidades em Sistemas Operacionais (SOBRINHO, 2024)

A realização de testes de intrusão (*pentest*) e a análise de vulnerabilidades em ambientes Windows como instrumento didático e de verificação de controles, reforçam a importância de uma configuração segura, telemetria acionável e políticas de atualização para reduzir a exposição do sistema. Nesse contexto, o trabalho analisou as vulnerabilidades em sistemas operacionais utilizando testes de intrusão com o framework Metasploit em ambientes virtuais. Os testes evidenciaram como as falhas conhecidas no Windows XP podem ser exploradas para ganhar controle total do sistema.

3.2 *Ransomware*: Análise, Técnica e Prevenção (JÚNIOR, 2023)

O *ransomware* é uma das maiores ameaças digitais da atualidade, podendo criptografar seus arquivos e exigir o pagamento de um resgate em troca da chave. Trabalhos focados em *ransomwares* mostram um padrão de ataque que se repete: entrada (acesso), escalação de privilégio, descoberta, exfiltração e cifragem. Uma eventual defesa que aparenta ser mais eficaz combina *Multi-Factor Authentication* (MFA), segmentação, bloqueio de execução não autorizada e backup. Nesse contexto, o trabalho aborda que a melhor defesa está na prevenção, como a realização de backups regulares, a manutenção de sistemas atualizados e a utilização de *firewalls* bem configurados, capazes de bloquear portas potencialmente vulneráveis.

3.3 Estudo e Análise de Vulnerabilidades em Serviços Publicados na Internet (KROTH, 2018)

O trabalho realiza um estudo e análise de vulnerabilidades em serviços publicados na internet, com o objetivo de identificar e quantificar falhas previamente definidas e documentadas. A pesquisa foca em vulnerabilidades de sistemas operacionais da Microsoft, especificamente nas versões Windows XP, Windows 7 e Windows 10, detalhando as vulnerabilidades MS12-020 (RDP) e MS17-010 (EternalBlue). O autor utiliza uma amostragem de mais de 3 milhões de

endereços IP brasileiros para verificar a incidência dessas falhas, demonstrando em máquinas virtuais os métodos de exploração e propondo as devidas correções e boas práticas para mitigá-las. No entanto, a pesquisa limita-se a investigar a borda externa da rede (as conexões da rede externa com destino à rede interna) e as vulnerabilidades já conhecidas na camada de aplicação. Dessa forma, o trabalho não aborda a análise de possíveis falhas na topologia da rede local (LAN) nem a descoberta de vulnerabilidades novas ou desconhecidas (*zero-day*).

3.4 Vulnerabilidade do Sistema Operacional Windows com Inserção de *Malware* por HID (AZEVEDO *et al.*, 2022)

Nesse trabalho, os autores mostraram que um hardware adulterado, contendo um microcontrolador, pode injetar um *malware* no Windows assim que é conectado via USB, passando-se por um teclado USB. Uma vez conectado, o sistema reconhece o dispositivo como um *Human Interface Device* (HID) legítimo, permitindo a execução de scripts maliciosos ocultos, como de *keyloggers* e *reverse shells*. O estudo investiga como os dispositivos HID, especificamente USBs com microcontroladores ocultos, podem ser explorados para injetar *malwares* nos sistemas operacionais Windows. O protótipo foi desenvolvido com uma placa Arduino Digispark, demonstrando que o sistema operacional reconhece o dispositivo sem as validações adequadas, acendendo o alerta quanto à necessidade de um melhor controle físico e de políticas de mídia removível.

3.5 Pentest Baseado em Imagens EnCase: Uma Análise de Vulnerabilidades em Servidores Web (SILVA, 2024)

Esse trabalho apresenta a criação e implementação de um laboratório para testes de intrusão (*pentest*) utilizando imagens geradas pelo software forense EnCase. O estudo de caso demonstra como obter acesso aos arquivos do Windows e redefinir a senha de administrador utilizando ferramentas como Slax e Hiren's Boot, além de configurar uma máquina Kali Linux para futuros ataques de rede. O trabalho discute vários tipos de vulnerabilidades, como falhas de programação (*injection*), erros no projeto do sistema (*insecure design*) e configurações de segurança incorretas (*security misconfiguration*). Também são mencionados ataques comuns, como engenharia social e negação de serviço. As vulnerabilidades exploradas no laboratório incluem um arquivo de política de *firewall* sem proteção para leitura e escrita e um sistema

operacional desativado. No entanto, o estudo simula um cenário pós-invasão, não abordando a exploração inicial da vulnerabilidade do site que originou o acesso.

3.6 Resumo

Os trabalhos apresentados focam, predominantemente, em sistemas operacionais legados (como Windows XP e Windows 7), em ataques específicos e limitados (como HID físico) ou em cenários pós-invasão, sem detalhar a exploração inicial. Os trabalhos que abordam o Windows 10 limitam-se a vulnerabilidades de borda externa e já bem documentadas (como MS12-020 e MS17-010). Diante disso, o presente trabalho se distingue dos demais ao focar em sistemas operacionais mais modernos da Microsoft: o Windows 10 e o Windows 11. A principal contribuição da pesquisa consiste na demonstração prática do desenvolvimento de um ataque que não apenas explora vulnerabilidades nesses ambientes, mas que implementa mecanismos de persistência, garantindo o acesso contínuo para a exfiltração de dados sensíveis do usuário. Além disso, a análise se mostra crítica diante do iminente fim do suporte ao Windows 10, o que torna as técnicas demonstradas uma ameaça ainda mais duradoura e perigosa, capaz de comprometer milhões de sistemas que deixarão de receber atualizações de segurança.

4 METODOLOGIA

Neste capítulo, descreveremos os procedimentos adotados para desenvolver a ferramenta de ataque utilizada na exploração de vulnerabilidades no Windows 10 e Windows 11.

4.1 Apresentação da Ferramenta

A ferramenta desenvolvida consiste em uma ferramenta de comando e controle (C2), a qual se enquadra nas fases de exploração e pós-exploração de um teste de intrusão (*pentest*). Em termos gerais, o seu funcionamento ocorre por meio da ativação de um agente em um dispositivo alvo, enquanto um servidor monitora a atividade do agente, enviando-lhe comandos para extrair e recuperar informações do dispositivo alvo de maneira não consentida. A ferramenta foi desenvolvida na linguagem de programação Go (Golang), a qual foi escolhida por sua eficiência, portabilidade e capacidade de gerar binários nativos para Windows 10 e Windows 11. Portanto, a ferramenta é composta por duas entidades: servidor e agente, que se comunicam entre si por meio do protocolo TCP/IP.

O servidor atua como o centro de controle de toda a operação. É ele quem envia instruções, recebe dados e coordena as ações executadas nos sistemas alvo. Assim, o servidor pode ser comparado a uma central de monitoramento, responsável por organizar e supervisionar as atividades em campo. O agente, por sua vez, é o componente que interage diretamente com o sistema operacional do usuário alvo. Ele funciona como uma extensão operacional do servidor, recebendo instruções, executando tarefas e retornando os resultados obtidos para o servidor.

O código-fonte da ferramenta está dispónivel e pode encontrado no repositório[1] do GitHub.

4.2 Funcionalidades da Ferramenta

A ferramenta possui comandos que simulam práticas utilizadas tanto por administradores de redes quanto por agentes maliciosos. Os comandos foram agrupados em 4 (quatro) categorias distintas: (i) comandos de arquivos e diretórios; (ii) comandos de sistema; (iii) comandos de persistência; e (iv) comandos de servidor.

[1] https://github.com/MarcosKiacola/TCC-de-Engenharia-da-Computa-o-da-UNILAB.git

4.2.1 Comandos de Arquivos e Diretórios

Consistem em comandos que simulam cenários de exfiltração de dados e movimentação lateral em sistemas comprometidos.

- `ls` - Lista os arquivos do diretório atual.
- `pwd` - Mostra o diretório atual.
- `cd` - Troca o diretório.
- `cat` - Exibe o conteúdo do arquivo.
- `send` - Envia o arquivo do servidor para o agente.
- `get` - Copia o arquivo do agente para o servidor.

4.2.2 Comandos de Sistema

Consistem em comandos que permitem explorar informações sensíveis do sistema e avaliar riscos operacionais.

- `whoami` - Identifica o usuário atual.
- `ps` - Lista os processos ativos.
- `sysinfo` - Informa detalhes do sistema, como: SO, arquitetura, *hostname*, usuário, CPUs, memória, disco e rede.
- `sleep` - Define um tempo de espera entre reconexões (útil em testes de persistência furtiva).
- `screenshot` - Captura a tela do usuário, simulando uma espionagem visual.
- `exec` - Executa comandos no `powershell.exe` (do próprio Windows), permitindo rodar qualquer instrução não reconhecida pelo agente, mas que seja válida pelo `cmd.exe` dos sistemas.

4.2.3 Comandos de Persistência

Consistem em comandos que aplicam persistência maliciosa no processo agente.

- `start` - Copia o agente para a pasta de inicialização do Windows (*Startup*).
- `registro` - Cria uma entrada no registro (*Run*) para execução automática.
- `tarefa` - Cria uma tarefa agendada para executar sempre que um usuário fizer login.

4.2.4 Comandos de Servidor

Consistem em comandos que permitem gerenciar múltiplos alvos, reproduzindo os cenários de ataques.

- `show agentes` - Lista os agentes conectados.
- `select <AgentID>` - Seleciona um agente específico para o envio de comandos.

4.3 Mascaramento do Agente

O processo de instalação do agente no Windows se dá através da execução de um arquivo executável (`agente.exe`). A ideia é que o executável `agente.exe` seja incorporado a um software legítimo, como, por exemplo, ao instalador do navegador Opera Mini. Dessa forma, um usuário não desconfiaria de que se trata de um arquivo manipulado, o qual foi combinado a um software malicioso. A motivação para a escolha desse ataque se dá pelo fato de ser uma situação que ocorre com bastante frequência no cotidiano dos usuários, a qual passa despercebida na maioria das vezes.

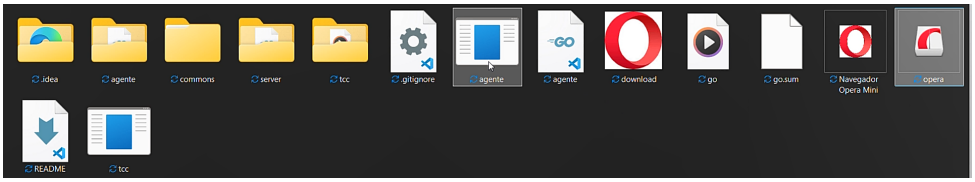
É comum que os usuários façam o download de arquivos e os executem automaticamente. Quando essa execução acontece a partir de um arquivo malicioso, não é necessário que o usuário chegue até o fim da execução para que a contaminação ocorra. Por se tratar de um arquivo contaminado com algum *malware*, basta que o usuário tente executar o arquivo e dê permissão para a execução. Isso será suficiente para que o *malware* entre em execução e permita que atacantes (usuários maliciosos) tenham acesso aos dados ou, até mesmo, às máquinas dos usuários.

No nosso cenário, onde combinamos nosso `agente.exe` ao instalador do Opera Mini, gerando um instalador malicioso, o usuário não precisaria chegar até o fim da instalação do Opera Mini. Caso deseje, ele poderia abortar o processo de instalação. No entanto, uma vez concedida a permissão para a execução do instalador malicioso, a máquina do usuário já estaria contaminada, visto que o `agente.exe` entra em execução a partir da execução do instalador modificado.

A seguir, apresentaremos o passo a passo para combinar os dois arquivos, `agente.exe` e o instalador do Opera Mini, em um único arquivo modificado que, embora continue funcional, agora será malicioso. Para tal, utilizamos a ferramenta WinRAR para unir os dois arquivos em um só, por meio da compactação de arquivos.

Inicialmente, selecionamos o arquivo malicioso (`agente.exe`) e o arquivo do instalador do navegador Opera Mini (`opera.exe`) para a compactação, conforme a Figura 2.

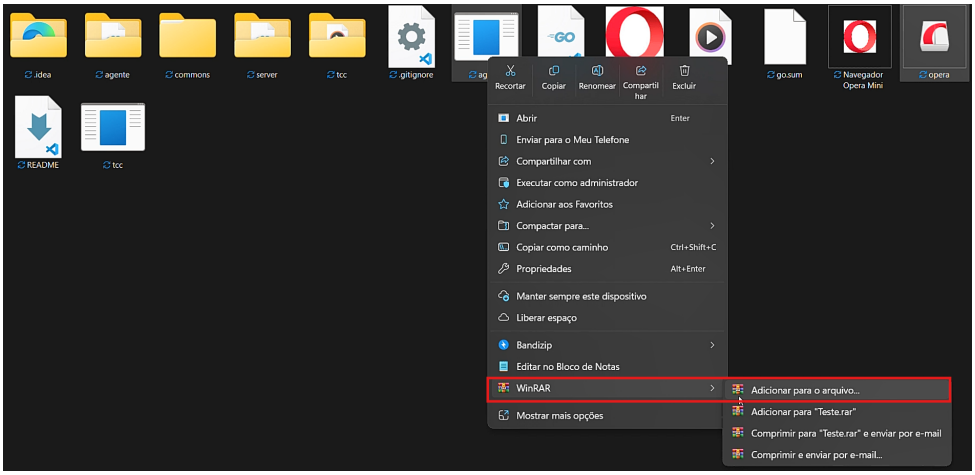
Figura 2 – Selecionando o arquivo malicioso (`agente.exe`) e o arquivo legítimo (`opera.exe`) para serem compactados.



Fonte: Autor.

Em seguida, iniciamos a compactação dos arquivos com as opções avançadas que a ferramenta WinRAR permite, conforme a Figura 3.

Figura 3 – Selecionando a compactação dos arquivos.



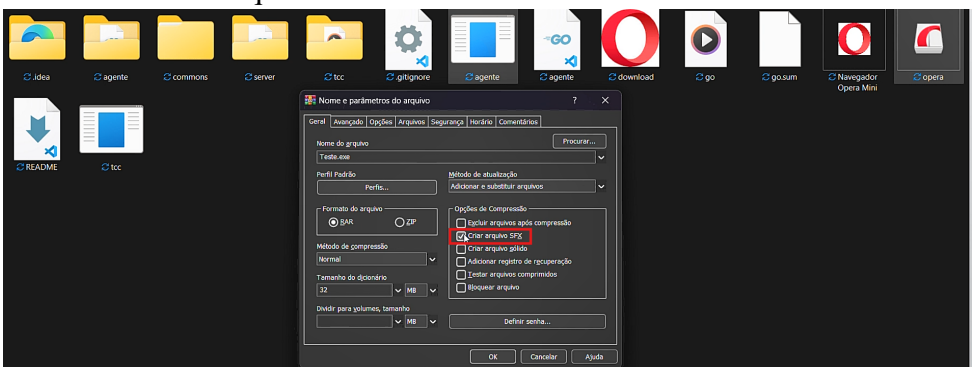
Fonte: Autor.

Após abrir o menu, damos um nome ao arquivo (`Teste.exe`) e marcamos a opção "Criar arquivo SFX", a qual criará um arquivo do tipo *Self-Extracting File* (SFX) que gera um arquivo .exe compactado, conforme a Figura 4. Um arquivo SFX é um tipo especial de arquivo compactado que inclui, além dos dados comprimidos, um módulo executável responsável por extrair automaticamente o conteúdo. Isso permite que os arquivos internos sejam descompactados simplesmente ao executar o SFX, sem depender de nenhum outro programa de descompactação. Na prática, ele funciona como um instalador simplificado. Ao ser aberto, o arquivo SFX realiza a extração e, se configurado, executa automaticamente os arquivos incluídos, sendo um formato bastante útil para distribuir programas ou atualizações de pacotes de forma prática.

Em seguida, clicamos na aba "Avançado", conforme a Figura 5.

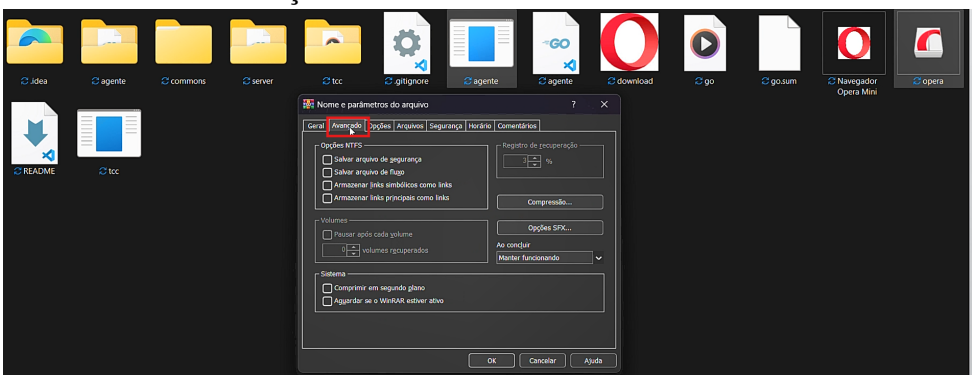
Depois, clicamos na opção "Opções SFX...", conforme a Figura 6.

Figura 4 – Renomeando o arquivo SFX.



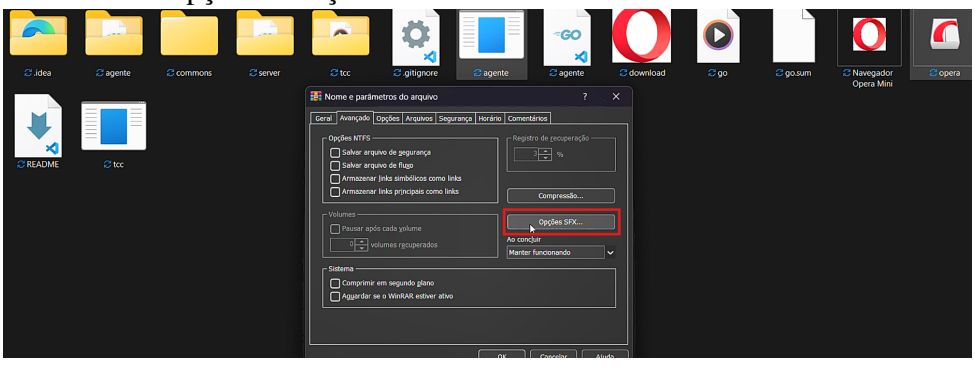
Fonte: Autor.

Figura 5 – Abrindo a aba “Avançado”.



Fonte: Autor.

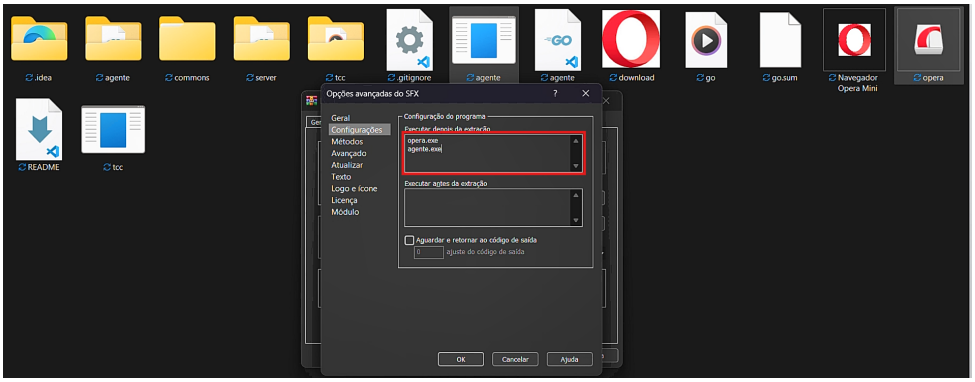
Figura 6 – Abrindo as opções avançadas do SFX.



Fonte: Autor.

No menu que se abre, clicamos na opção “Configurações”. Na primeira área de edição, sob o menu “Configurações do programa → Executar depois da extração”, adicionamos a ordem de execução dos programas após a descompactação, conforme a Figura 7. Desejamos que os usuários não saibam que algo malicioso está sendo executado. Portanto, o arquivo legítimo deve ser sempre o primeiro a ser executado (`opera.exe`), seguido pelo arquivo malicioso (`agente.exe`).

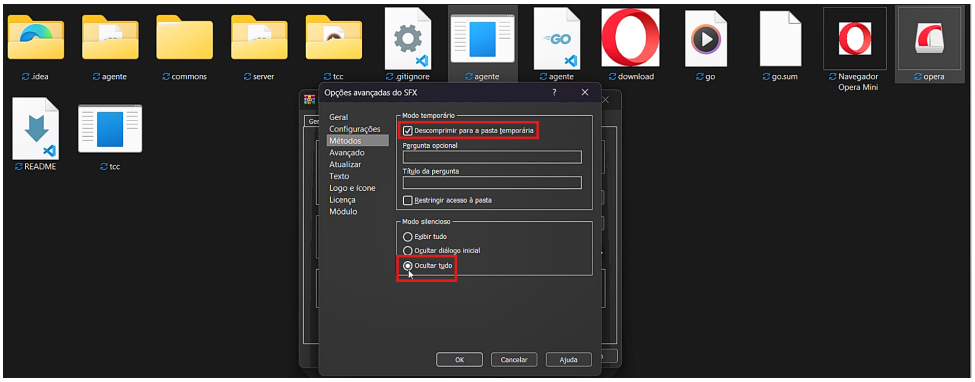
Figura 7 – Configurando as opções avançadas do arquivo SFX (parte 01).



Fonte: Autor.

Em seguida, clicamos na opção “Métodos”. Na primeira área de edição, sob o menu “Modo temporário”, marcamos a opção “Descomprimir para a pasta temporária”. Na segunda área de edição, sob o menu “Modo silencioso”, marcamos a opção “Ocultar tudo”, conforme a Figura 8.

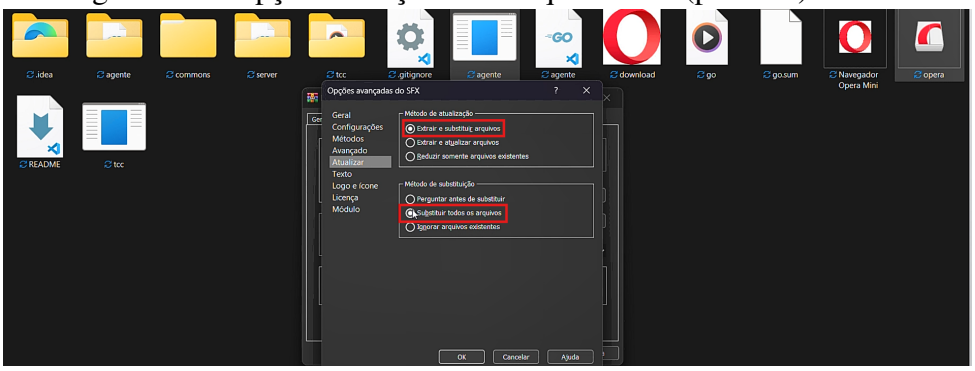
Figura 8 – Configurando as opções avançadas do arquivo SFX (parte 02).



Fonte: Autor.

Depois, clicamos na opção “Atualizar”. Na primeira área de edição, sob o menu “Método de atualização”, marcamos a opção “Extrair e substituir arquivos”. Na segunda área de edição, sob o menu “Método de substituição”, marcamos a opção “Substituir todos os arquivos”, conforme a Figura 9.

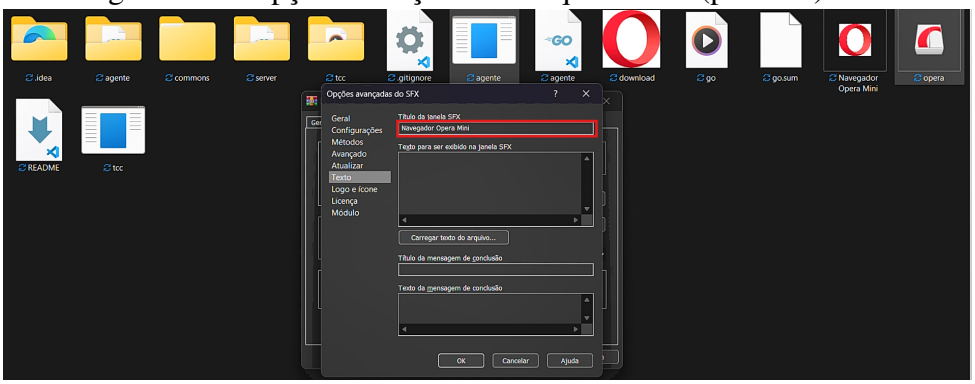
Figura 9 – Configurando as opções avançadas do arquivo SFX (parte 03).



Fonte: Autor.

Agora, clicamos na opção “Texto”. Na primeira área de edição, sob o menu “Título da janela SFX”, colocamos o valor “Navegador Opera Mini”, que será o título que aparecerá na janela de execução quando o arquivo SFX for executado, conforme a Figura 10.

Figura 10 – Configurando as opções avançadas do arquivo SFX (parte 04).



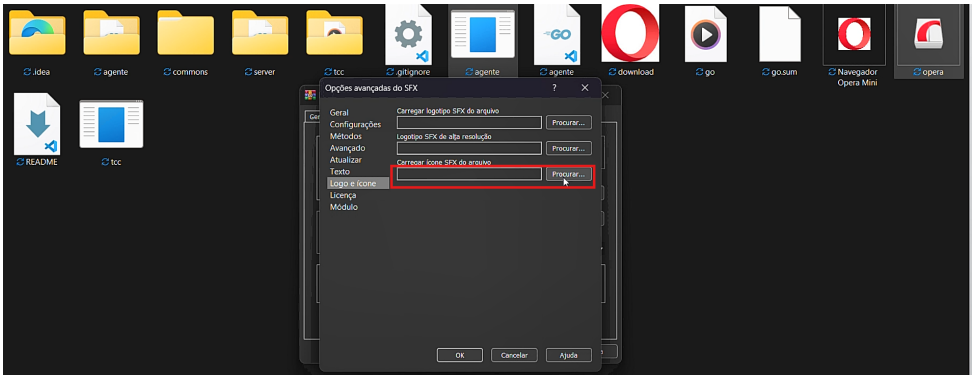
Fonte: Autor.

Para o último passo, clicamos na opção “Logo e ícone”. Essa etapa é crucial, pois nela definimos o ícone do arquivo combinado. É desejável que o arquivo tenha um ícone que remeta ao ícone do arquivo legítimo, para que o usuário não desconfie da maliciosidade do arquivo. Na terceira área de edição, sob o menu “Carregar ícone SFX do arquivo”, clicamos em “Procurar”, conforme a Figura 11, e selecionamos o ícone desejado, conforme a Figura 12.

Por fim, clicamos em “OK” e aguardamos a geração do arquivo SFX, com o nome escolhido no início do processo de configuração, conforme a Figura 13.

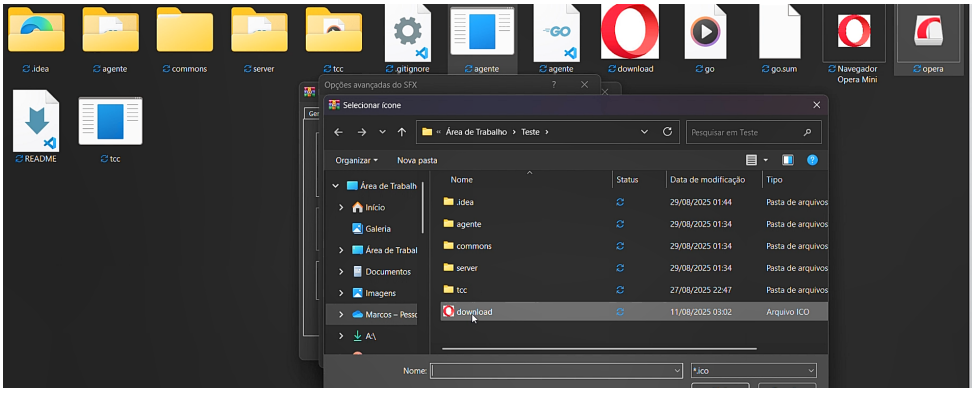
Uma vez criado o arquivo SFX, este pode ser renomeado para qualquer outro nome que soe mais familiar ao usuário e facilite a sua execução.

Figura 11 – Configurando as opções avançadas do arquivo SFX (parte 05).



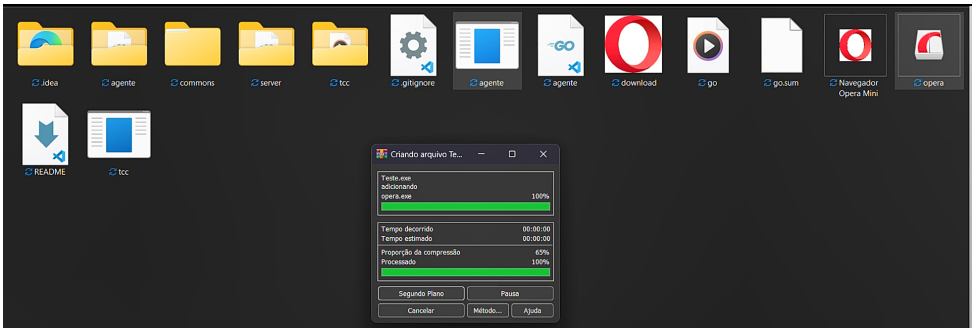
Fonte: Autor.

Figura 12 – Selecionando o ícone do arquivo SFX.



Fonte: Autor.

Figura 13 – Gerando o arquivo SFX.



Fonte: Autor.

4.4 Limitações da Ferramenta

- A ferramenta foi desenvolvida com foco acadêmico e não possui todos os recursos de frameworks avançados, como o Metasploit (Rapid7, 2025) ou Cobalt Strike (Fortra, 2025).
- A ferramenta considera principalmente vulnerabilidades relacionadas à persistência e exploração de informações locais.

4.5 Preocupações Éticas

É fundamental destacar que a ferramenta foi utilizada apenas em ambientes isolados e controlados. Os usuários foram simulados através de máquinas virtuais e sem conexão com redes externas. O uso da ferramenta de comando e controle (C2) seguiu princípios éticos e teve como único objetivo avaliar as vulnerabilidades de forma acadêmica, sem intenção de uso malicioso.

Portanto, somos **terminantemente** contra o uso da ferramenta pra fins maliciosos, e recomendamos o uso da ferramenta apenas pra fins de estudo e acadêmicos.

5 RESULTADOS EXPERIMENTAIS

Neste capítulo, apresentamos os resultados obtidos a partir da execução da ferramenta desenvolvida em ambientes controlados com os sistemas operacionais Windows 10 e Windows 11, e analisaremos se estratégias de mitigação já foram empregadas nas respectivas versões para corrigir as vulnerabilidades. A análise permite compreender de que forma as vulnerabilidades conhecidas do Windows 10 permanecem relevantes e como o Windows 11 responde a essas mesmas tentativas de exploração.

Para a realização dos experimentos, a seguinte configuração foi utilizada:

- Os ambientes controlados foram criados através da ferramenta Oracle VirtualBox 7.2.4, que nos permite criar máquinas virtuais isoladas, com sistemas operacionais e recursos próprios.
- Uma instância de máquina virtual com o sistema operacional Windows 10, versão 22H2, que executará o processo agente.
- Uma instância de máquina virtual com o sistema operacional Windows 11, versão 24H2, que executará o processo agente.
- Uma instância de máquina virtual com o sistema operacional Kali Linux, versão 2025.2, que executará o processo servidor.

5.1 Pré-execução

Antes de inicializar os testes, vamos considerar que a vítima (usuário) já tenha realizado o download do arquivo malicioso, conforme detalhado no Capítulo 4. É importante mencionar que o arquivo malicioso poderia ter sido baixado a partir de qualquer site, conforme exemplifica a Figura 14. Para a realização dos testes, consideramos que o arquivo malicioso se chama `OperaGXSetup.exe`.

O primeiro passo consiste em iniciar o processo do servidor na Máquina Virtual (VM) contendo o Kali Linux e esperar por uma conexão maliciosa através da execução do `agente.exe` na máquina da vítima. Assim que a vítima executar o arquivo malicioso, o servidor receberá imediatamente uma conexão com o endereço IP da máquina alvo e um identificador (ID) do agente que foi executado naquele momento. É importante mencionar que um mesmo agente de uma mesma máquina pode ter vários IDs, visto que ele se baseia na data/hora da execução. Como a data/hora da execução é variável, esse ID pode variar a cada execução, assim

Figura 14 – Download do arquivo malicioso `OperaGXSetup.exe`.

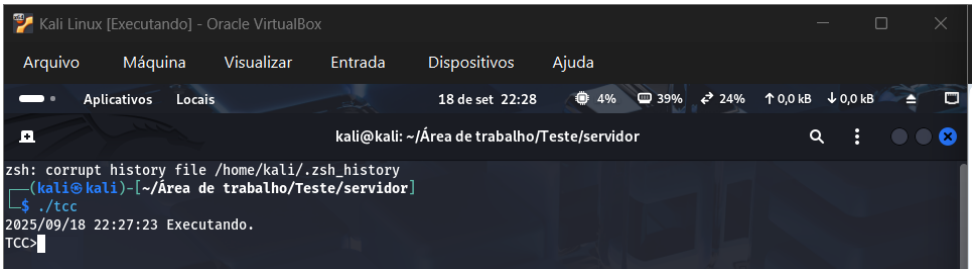
Fonte: Autor.

como o próprio IP da vítima também pode alterar, por exemplo, ao mudar de conexão de internet (de Wi-Fi para cabo e vice-versa).

O servidor foi projetado para receber várias conexões de maneira simultânea e interagir com elas uma por vez, selecionando o alvo específico pelo comando `select <AgentID>` e passando o ID da vítima que pretende escolher. Vale ressaltar que, ao abrir mais de um terminal por vez, pode-se executar vários processos de servidor simultaneamente e, assim, efetuar um ataque em massa. No entanto, esse não é o foco do trabalho. Iremos demonstrar apenas um ataque direcionado com os comandos descritos na metodologia (Capítulo 4).

A Figura 15 apresenta o processo do servidor sendo executado no Kali Linux, o qual representa a máquina de um eventual atacante ou usuário malicioso.

Figura 15 – Servidor aguardando por uma conexão do agente.

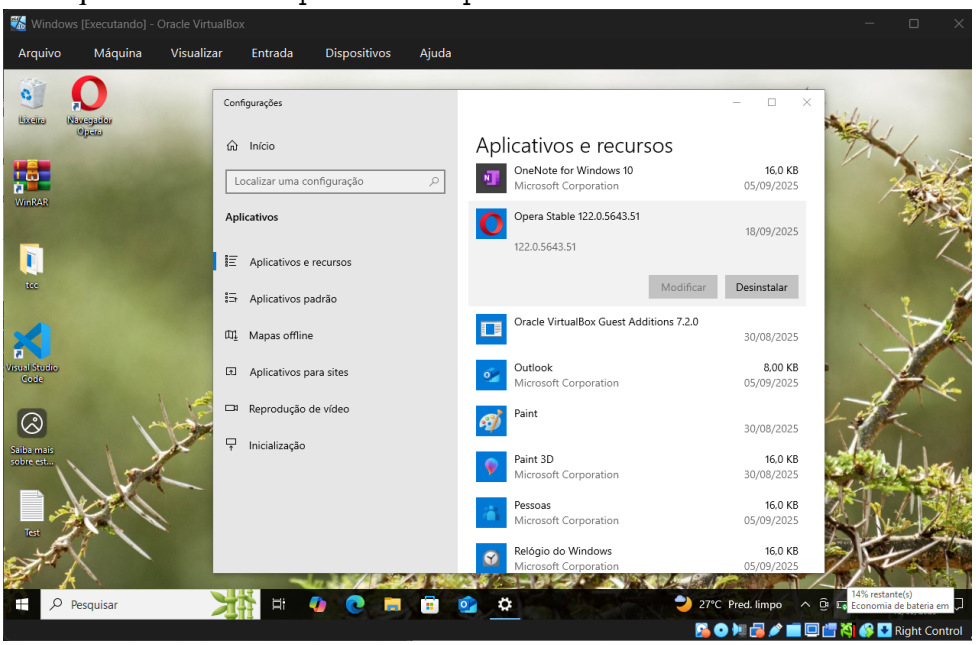


Fonte: Autor.

Em seguida, executamos o programa malicioso (`OperaGXSetup.exe`) no dispositivo da vítima alvo. Vale lembrar que não é preciso instalar o programa. É necessário apenas dar permissão para a execução do arquivo malicioso, pois isso fará com que o `agente.exe` seja executado e permitirá que o servidor acesse a máquina alvo.

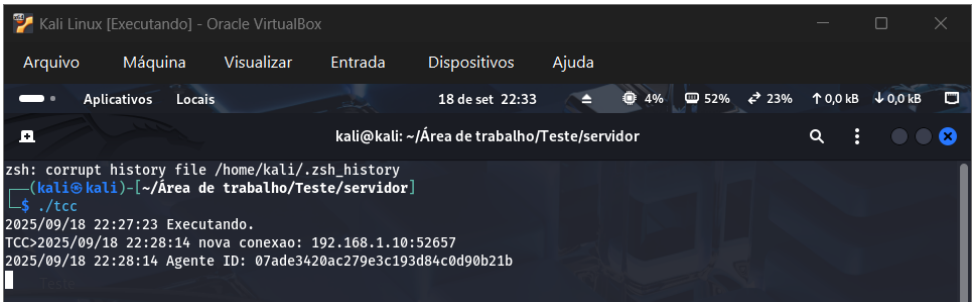
No primeiro cenário, executamos e instalamos o arquivo malicioso `OperaGXSetup.exe` na VM que contém o sistema operacional Windows 10. Conforme a Figura 16, o navegador Opera foi instalado com sucesso no dispositivo. Note que, imediatamente após a execução do arquivo malicioso, o servidor recebe o aviso da nova conexão com o agente instalado no Windows 10, conforme a Figura 17.

Figura 16 – Arquivo malicioso `OperaGXSetup.exe` instalado no Windows 10.



Fonte: Autor.

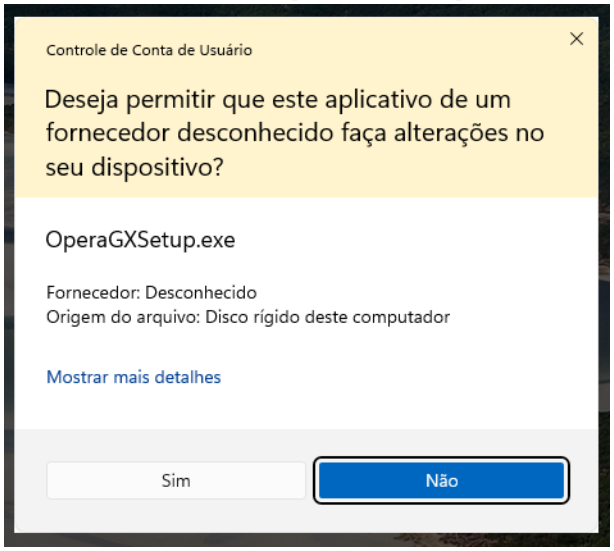
Figura 17 – Servidor recebendo a conexão do agente no Windows 10.



Fonte: Autor.

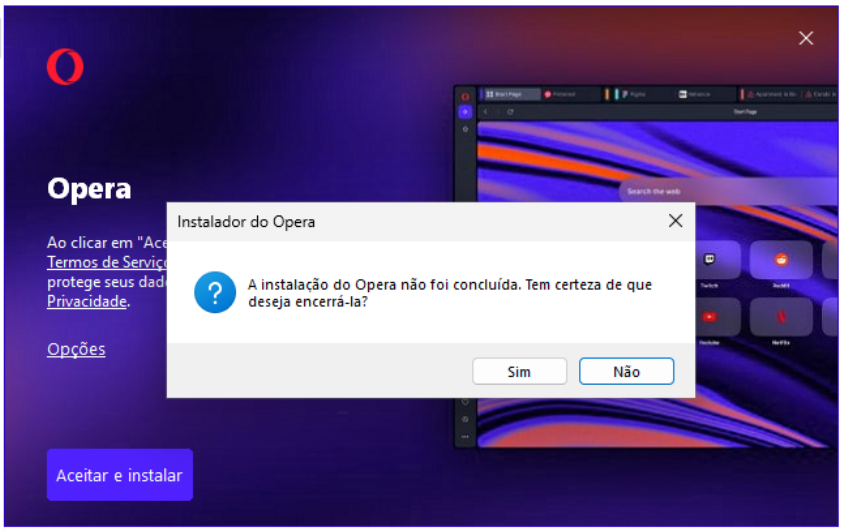
Por sua vez, no segundo cenário, executamos o arquivo malicioso `OperaGXSetup.exe` na VM que contém o sistema operacional Windows 11. Porém, diferentemente do cenário anterior, não instalamos o navegador, apenas demos permissão para a execução do arquivo e interrompemos a instalação logo em seguida, conforme as Figuras 18 e 19, respectivamente. Note que, imediatamente após a execução do arquivo malicioso, o servidor também recebe o aviso da nova conexão com o agente instalado no Windows 11, conforme a Figura 20.

Figura 18 – Executando o arquivo malicioso `OperaGXSetup.exe` no Windows 11.



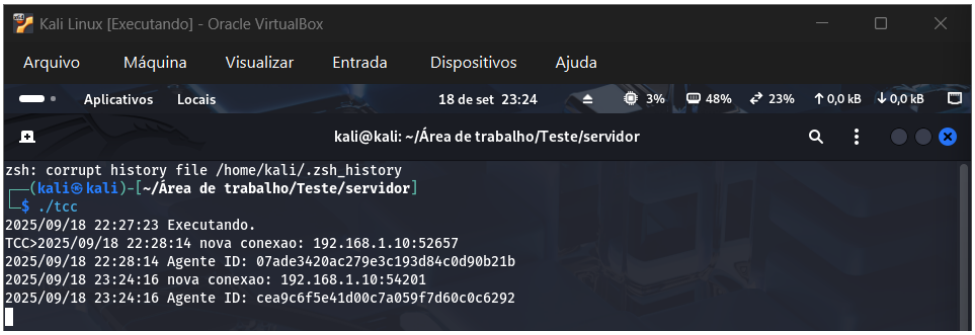
Fonte: Autor.

Figura 19 – Interrompendo a instalação do navegador Opera.



Fonte: Autor.

Figura 20 – Servidor recebendo a conexão do agente no Windows 11.



Fonte: Autor.

Nas próximas seções, apresentaremos os resultados obtidos após a execução de um conjunto de comandos nos dispositivos Windows 10 e Windows 11, respectivamente.

5.2 Resultados no Windows 10

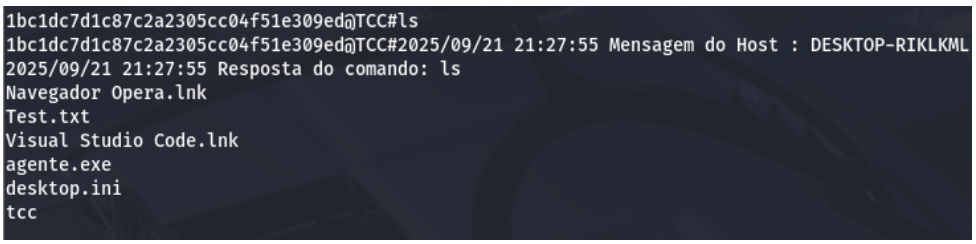
Primeiramente, selecionamos o agente instalado no Windows 10 para enviar os comandos à máquina alvo, através do comando `select <AgentID>`, conforme a Figura 21.

Figura 21 – Selecionando o agente no Windows 10.

Fonte: Autor.

Em seguida, enviamos o primeiro comando para a vítima, o comando para listar os arquivos em seu diretório atual (`ls`). A Figura 22 apresenta o comando enviado pelo servidor e a resposta enviada pela vítima.

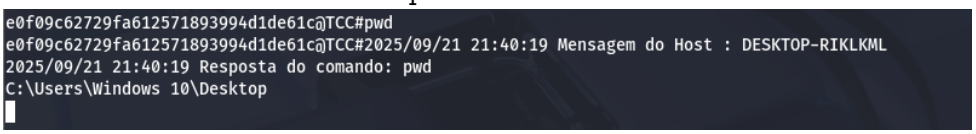
Figura 22 – Funcionamento do comando `ls` no Windows 10.



Fonte: Autor.

Depois, enviamos o segundo comando para a vítima, o comando para recuperar o seu diretório atual (`pwd`). A Figura 23 apresenta o comando enviado pelo servidor e a resposta enviada pela vítima.

Figura 23 – Funcionamento do comando `pwd` no Windows 10.



Fonte: Autor.

A partir da execução do comando `ls`, conforme a Figura 22, podemos notar que existe um diretório chamado `tcc`. Dessa forma, o próximo passo consiste em executar o comando

`cd tcc` para mudar para esse diretório. A Figura 24 apresenta o comando enviado pelo servidor e a resposta enviada pela vítima.

Figura 24 – Funcionamento do comando `cd tcc` no Windows 10.

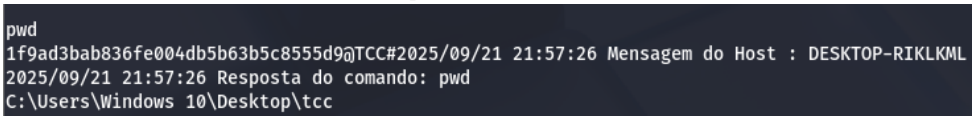
Fonte: Autor.

Em seguida, repetimos os comandos `ls` e `pwd` para listar os arquivos presentes no diretório `tcc` e nos certificar de que estamos no novo diretório. As Figuras 25 e 26 apresentam as respostas enviadas pela vítima.

Figura 25 – Funcionamento do comando `ls` no diretório `tcc` no Windows 10.

Fonte: Autor.

Figura 26 – Funcionamento do comando `pwd` no diretório `tcc` do Windows 10.

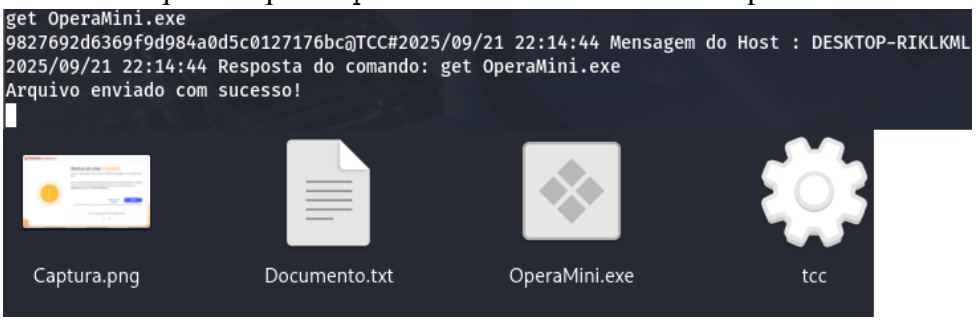


Fonte: Autor.

Agora, iremos fazer o download do arquivo `OperaMini.exe`, o qual está presente no diretório `tcc`, conforme evidenciado na Figura 25. O download consiste em transferir o arquivo da máquina da vítima para a máquina do servidor. Através do comando `get OperaMini.exe`, enviamos o arquivo para a máquina do servidor. Na Figura 27, podemos observar que o arquivo `OperaMini.exe` foi transferido e está localizado dentro do diretório atual do servidor.

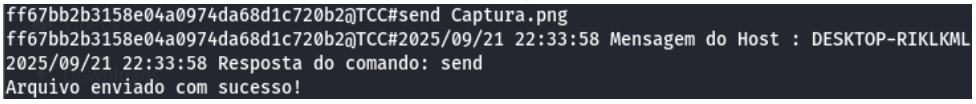
Por fim, vamos realizar a operação contrária. Em vez de transferir um arquivo da máquina da vítima para o servidor, enviaremos um arquivo do servidor para a máquina da vítima. Assim, executamos o comando `send Captura.png` para enviar o arquivo `Captura.png` da máquina do servidor para a máquina da vítima. Conforme a Figura 27, podemos notar que o arquivo, de fato, existe na máquina do servidor. A Figura 28 exibe o comando sendo executado para enviar o arquivo, enquanto a Figura 29 exibe a lista de arquivos atualizados no diretório `tcc` da máquina da vítima.

Figura 27 – Funcionamento do comando `get OperaMini.exe` no Windows 10. É possível observar que o arquivo `OperaMini.exe` foi transferido para o servidor.



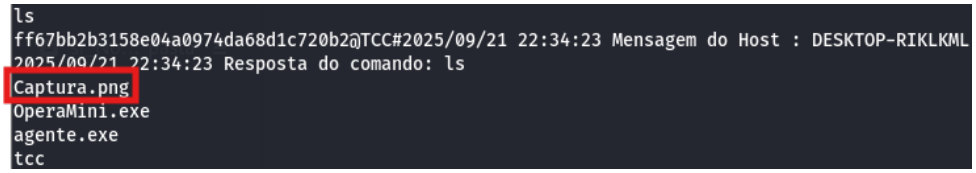
Fonte: Autor.

Figura 28 – Funcionamento do comando `send Captura.exe` no Windows 10.



Fonte: Autor.

Figura 29 – Listagem dos arquivos no diretório `tcc` do Windows 10 após receber o arquivo Captura.txt transferido do servidor.



Fonte: Autor.

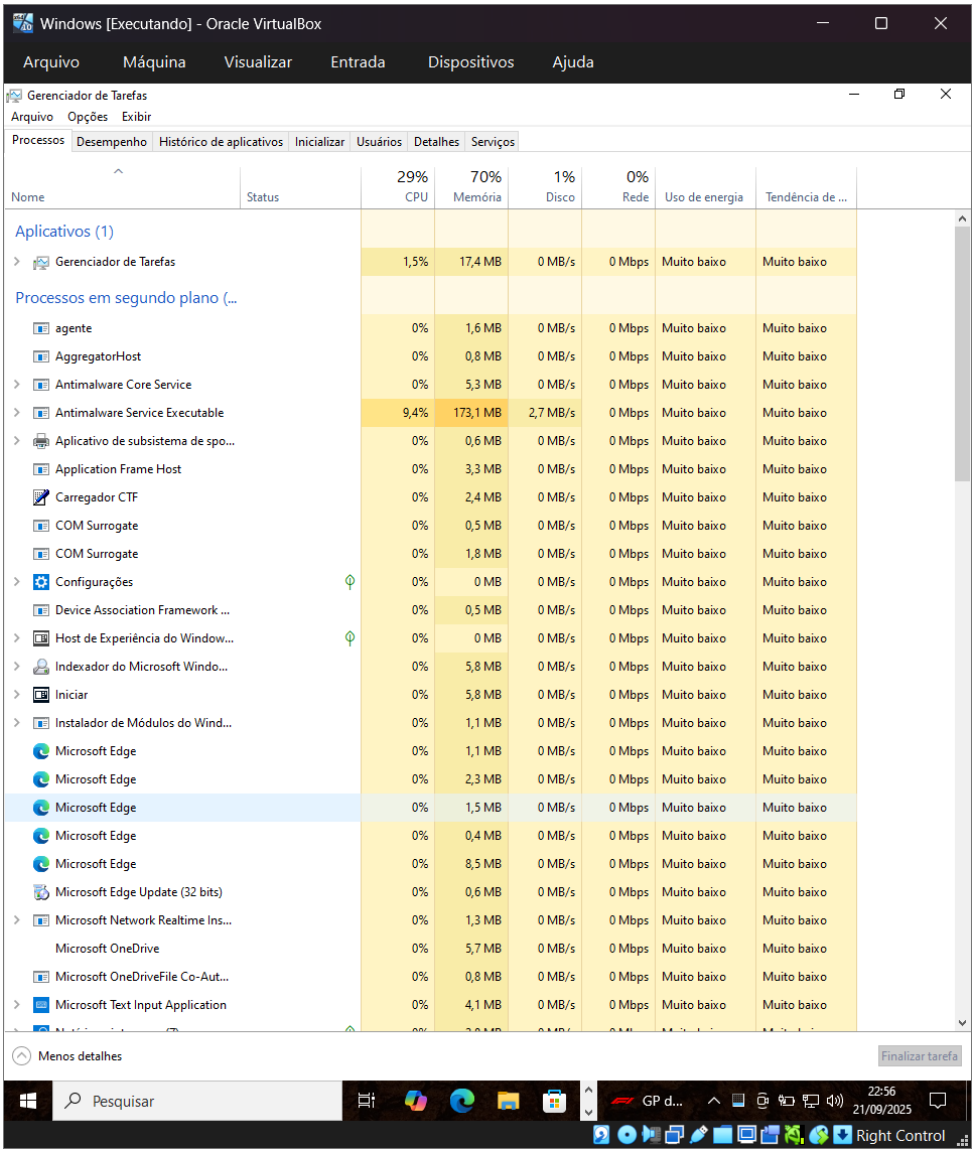
Depois de executados uma série de comandos de arquivos e diretórios e de comandos de servidor, abordaremos agora os comandos de persistência. Os comandos de persistência visam garantir que o agente malicioso (`agente.exe`) seja sempre executado, mesmo quando a máquina da vítima for reiniciada.

Primeiramente, nos certificamos de que o processo `agente.exe` continua em execução na máquina da vítima (Figura 30).

Uma vez verificado que o processo `agente.exe` continua em execução, executamos o primeiro comando de persistência (`start`), o qual copia o agente malicioso `agente.exe` para a lista de programas que devem ser executados automaticamente ao iniciar o Windows 10. A Figura 31 apresenta o comando sendo executado.

Com isso, vamos reinicializar a máquina da vítima para analisar se o processo `agente.exe` será executado automaticamente após a inicialização do Windows 10. No entanto, conforme evidenciado pela ausência do processo `agente.exe` na Figura 32, o Microsoft Defender bloqueia e não permite que o arquivo `agente.exe` seja executado automaticamente após a reinicialização. Portanto, a primeira técnica de persistência foi bloqueada pelo Windows 10.

Figura 30 – Lista de processos em execução no Windows 10.



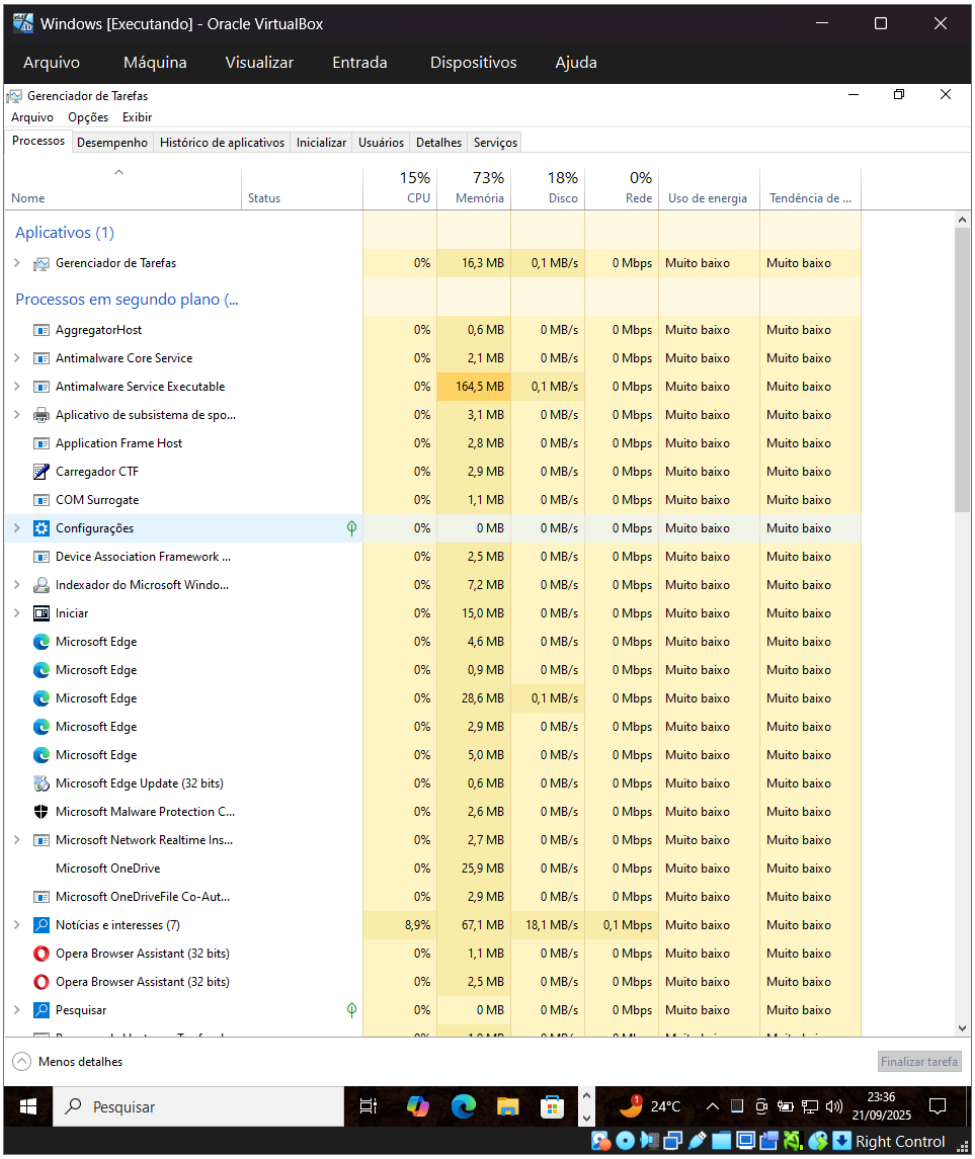
Fonte: Autor.

Figura 31 – Execução do primeiro comando de persistência (`start`) no Windows 10.

```
b27229dd768e71d3efad0b99ed811ff5@TCC#start
b27229dd768e71d3efad0b99ed811ff5@TCC#2025/09/21 23:18:24 Mensagem do Host : DESKTOP-RIKLKML
2025/09/21 23:18:24 Resposta do comando: start
Agente copiado para: C:\Users\Windows 10\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\agente.exe
```

Fonte: Autor.

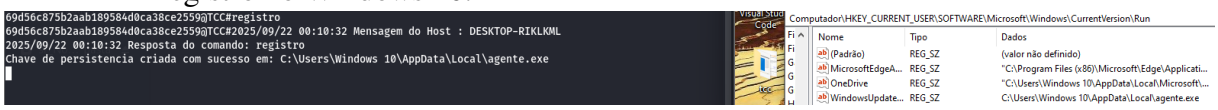
Figura 32 – Lista de processos em execução no Windows 10 após a reinicialização.



Fonte: Autor.

A segunda técnica de persistência consiste em criar uma entrada de registro na pasta `Run` do sistema operacional Windows 10, através do comando `registro`. A Figura 33 apresenta a execução do comando com a respectiva entrada de registro criada, sob o nome de "WindowsUpdate".

Figura 33 – Execução do segundo comando de persistência (`registro`) e criação da entrada de registro no Windows 10.

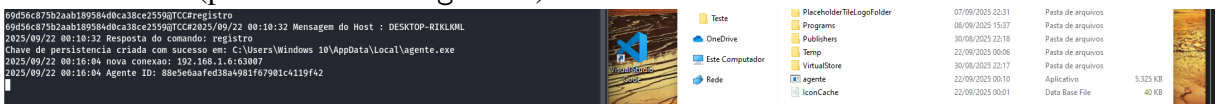


Fonte: Autor.

Depois de reiniciar a máquina da vítima, podemos observar que essa técnica funcio-

nou corretamente. Após a reinicialização da máquina, a conexão entre o agente e o servidor foi restabelecida automaticamente no Windows 10, conforme apresentado na Figura 34.

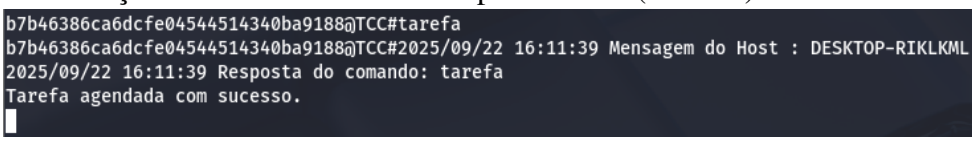
Figura 34 – Conexão restabelecida entre o agente e o servidor após a reinicialização do Windows 10 (persistência de `registro`).



Fonte: Autor.

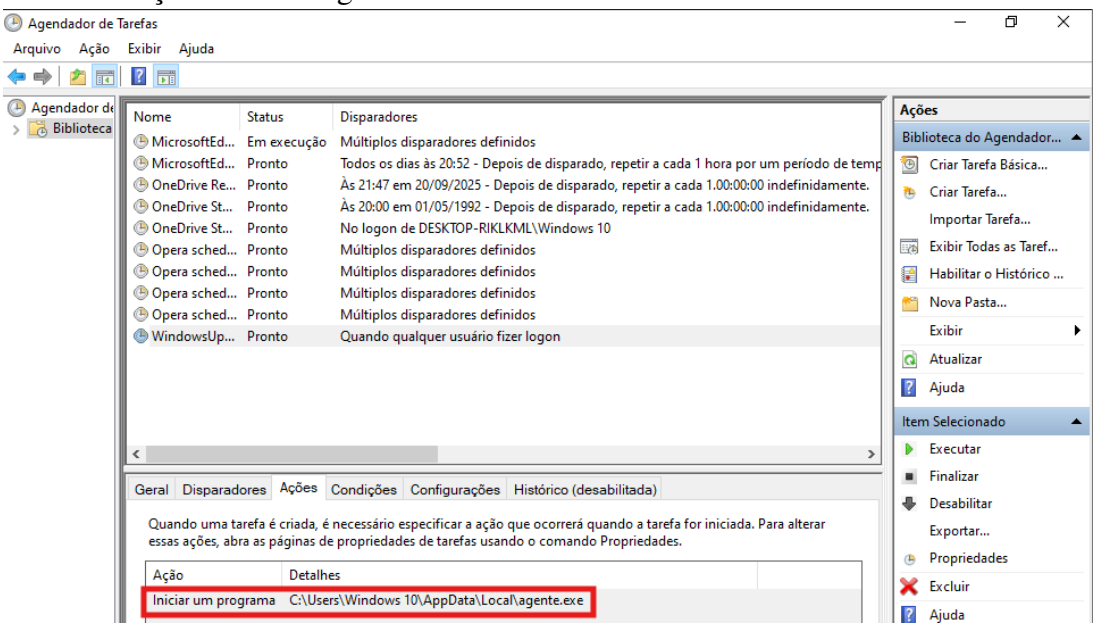
Por fim, a última técnica de persistência consiste em agendar uma tarefa no sistema operacional para que, sempre que o sistema for inicializado e o usuário fizer login, o programa `agente.exe` seja executado. Para essa técnica, executamos o comando `tarefa`, conforme apresentado na Figura 35. Após a execução bem sucedida do comando, uma nova tarefa agendada é criada (Figura 36).

Figura 35 – Execução do terceiro comando de persistência (`tarefa`).



Fonte: Autor.

Figura 36 – Criação da tarefa agendada no Windows 10.

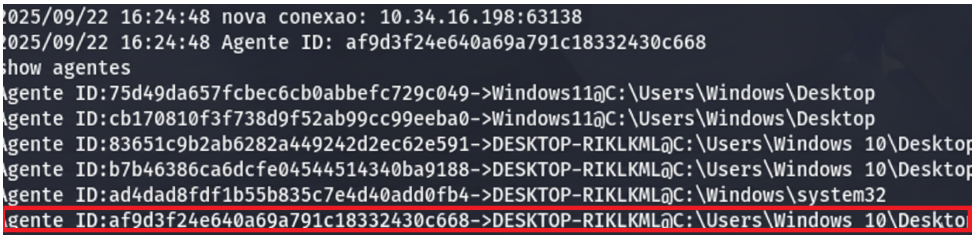


Fonte: Autor.

Depois de reiniciar a máquina da vítima, podemos observar que essa técnica também funcionou corretamente. Após a reinicialização da máquina, a conexão entre o agente e o servidor

foi restabelecida automaticamente no Windows 10, conforme apresentado na Figura 37.

Figura 37 – Conexão restabelecida entre o agente e o servidor após a reinicialização do Windows 10 (persistência de `tarefa`).



Fonte: Autor.

Assim, concluímos que apenas a primeira técnica de persistência, por meio do comando `start`, falhou durante os experimentos no sistema operacional Windows 10.

5.3 Resultados no Windows 11

Nesta seção, apresentaremos os resultados obtidos após a execução dos comandos de persistência no Windows 11. Pouparemos os detalhes das execuções das demais famílias de comandos (arquivos e diretórios, sistema e servidor), visto que, assim como no Windows 10, também funcionaram no sistema Windows 11.

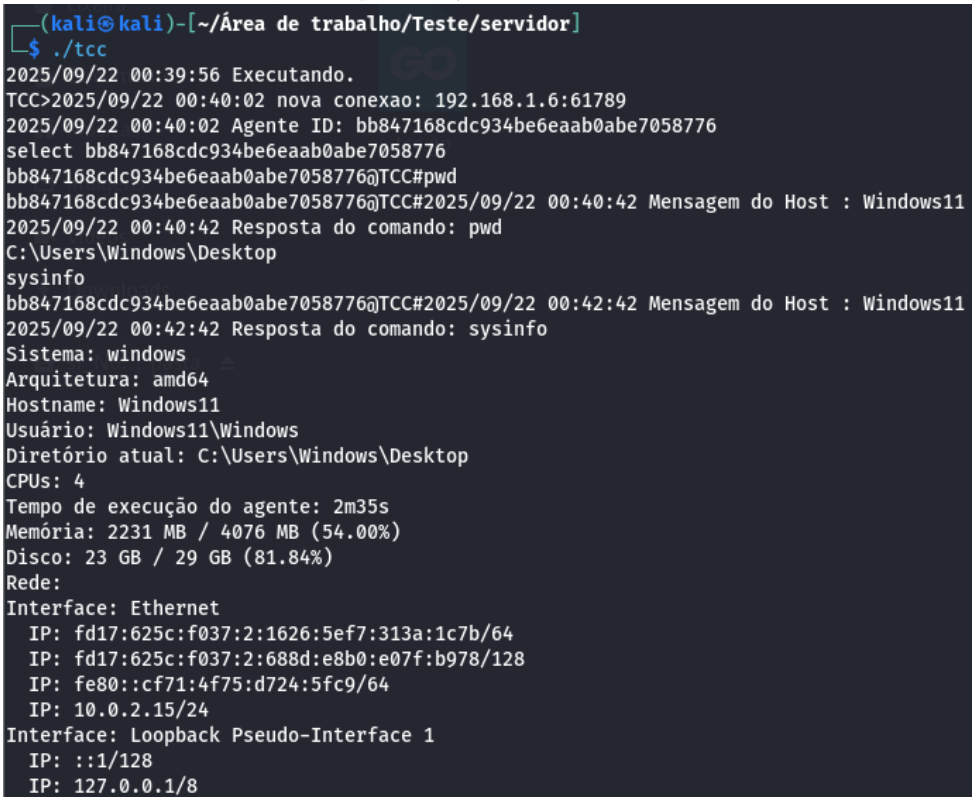
Primeiramente, vamos executar os comandos `sysinfo` e `pwd` para comprovar que se trata da máquina com o sistema operacional Windows 11 instalado e mostrar o diretório atual dessa máquina. A Figura 38 apresenta os comandos enviados pelo servidor e as respostas enviadas pela vítima.

Assim como foi conduzido nos experimentos do Windows 10, aplicaremos as três técnicas de persistência, na mesma ordem, no Windows 11. Assim como no Windows 10, a primeira técnica de persistência, por meio do comando `start`, também não funcionou. A razão para o não funcionamento dessa técnica no Windows 11 deve-se ao fato de que essa vulnerabilidade já foi corrigida no Windows 10.

A segunda técnica de persistência consiste em criar uma entrada no registro do sistema, através do comando `registro`. A Figura 39 apresenta a execução do comando com a respectiva entrada de registro criada, sob o nome de “WindowsUpdate”.

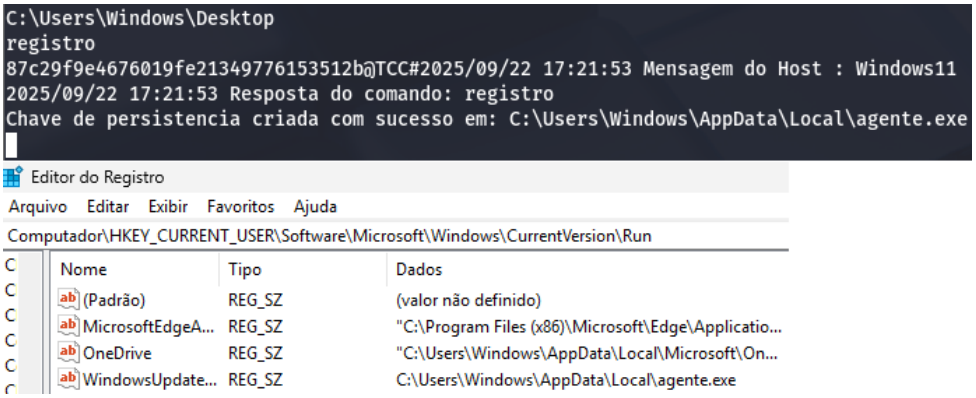
Depois de reiniciar a máquina da vítima, podemos observar que essa técnica funcionou corretamente. Após a reinicialização da máquina, a conexão entre o agente e o servidor foi restabelecida automaticamente no Windows 11, conforme apresentado na Figura 40.

Figura 38 – Execução dos comandos `pwd` e `sysinfo` no Windows 11.



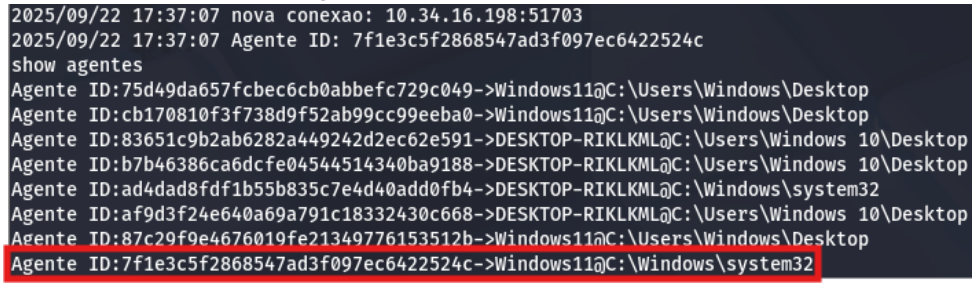
Fonte: Autor.

Figura 39 – Execução do segundo comando de persistência (`registro`) e criação da entrada de registro no Windows 11.



Fonte: Autor.

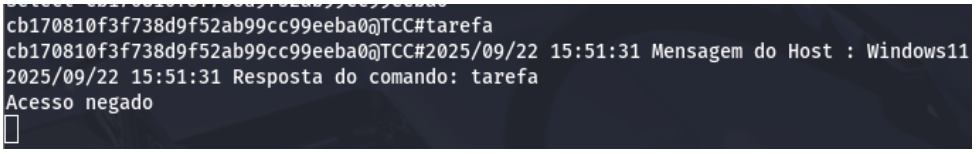
Figura 40 – Conexão restabelecida entre o agente e o servidor após a reinicialização do Windows 11 (persistência de `registro`).



Fonte: Autor.

Por fim, a última técnica de persistência se dá por meio do agendamento de tarefas, através do comando `tarefa`. No entanto, o Windows 11 bloqueia automaticamente a criação de tarefas agendadas, conforme exibido na Figura 41.

Figura 41 – Execução, e bloqueio, do terceiro comando de persistência (`tarefa`) no Windows 11.



Fonte: Autor.

Como podemos notar, apenas uma técnica de persistência foi executada com sucesso no Windows 11. É importante destacar que as demais famílias de comandos (arquivos e diretórios, sistema e servidor) podem ser executadas no Windows 11 e, consequentemente, no Windows 10 também, pois não precisam de elevação de privilégios, como os comandos de persistência.

6 CONCLUSÕES

Este capítulo busca consolidar os resultados alcançados ao longo do trabalho, trazendo uma reflexão sobre os resultados obtidos, os impactos práticos para a segurança da informação e as diretrizes para uma migração estruturada do Windows 10 para o Windows 11.

Ao longo deste estudo, foi possível demonstrar que o Windows 10, embora ainda amplamente utilizado, apresenta fragilidades significativas quando comparado ao Windows 11. Os experimentos mostraram que técnicas clássicas de persistência, exfiltração e coleta de informações podem ser realizadas no Windows 10 sem gerar alertas relevantes, o que reforça as vulnerabilidades dessa versão, especialmente diante do fim oficial de seu suporte.

O Windows 11, por sua vez, mesmo não sendo imune, revelou uma maior capacidade de registro e monitoramento. Várias das mesmas ações executadas no Windows 10 geram eventos nos logs do sistema operacional, criando oportunidades para a detecção precoce e a resposta rápida. Em termos práticos, isso não significa que o Windows 11 elimina o risco, mas sim que aumenta as chances de defesa, desde que as ferramentas de segurança estejam devidamente configuradas e monitoradas.

Como dito, o Windows 11 traz melhorias relevantes, principalmente em auditoria e monitoramento, mas não pode ser visto como uma solução definitiva. A principal segurança está na soma entre tecnologia, processos e pessoas conscientizadas. Em termos acadêmicos, o uso da ferramenta de comando e controle (C2), desenvolvida de forma ética e especificamente para este estudo, mostrou, de maneira clara, como os ataques são realizados na prática, simulando um atacante mal intencionado explorando as falhas do sistema. Os resultados obtidos com a ferramenta reforçam que algumas fragilidades permanecem até mesmo em ambientes modernos.

Portanto, a transição do Windows 10 para o Windows 11 não deve ser entendida apenas como uma atualização estética ou de desempenho, mas como um processo estratégico de segurança. É preciso enxergar o processo como parte de uma estratégia contínua de segurança, que envolve a atualização de ferramentas, a revisão de políticas de acesso, o monitoramento de logs e a conscientização dos usuários, que, além das vulnerabilidades, são os principais fatores explorados. A migração para o Windows 11 já é uma necessidade urgente para empresas e usuários domésticos que desejam operar de forma segura em um cenário cada vez mais hostil. Permanecer no Windows 10 após o fim do suporte oficial é assumir riscos desnecessários que podem comprometer tanto informações pessoais quanto os dados de grandes corporações.

REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 27032: Segurança cibernética – Diretrizes para segurança na internet**. 2024.

AZEVEDO, N. de F.; PEREIRA, R. A.; SEMENTE, V. G.; WAGNER, J. C. Trabalho de Conclusão de Curso (Graduação Engenharia da Computação), **Vulnerabilidade do sistema operacional Windows com inserção de malware por HID**. 2022.

CVE Program. **CVE: Common Vulnerabilities and Exposures**. 2025. <https://www.cve.org/>. Acessado em: 30-10-2025.

FILHO, N. P. R.; FREITAS, D. P. de. Ransomware: origens, consequências e prevenção. **Studies in Engineering and Exact Sciences**, v. 4, n. 1, p. 427–438, 2023.

FIRST.org, Inc. **Common Vulnerability Scoring System SIG**. 2025. <https://www.first.org/cvss/>. Acessado em: 30-10-2025.

Fortra. **Cobalt Strike**. 2025. <https://www.cobaltstrike.com/>. Acessado em: 30-10-2025.

HINTZBERGEN, J.; HINTZBERGEN, K. **Foundations of Information Security Based on ISO27001 and ISO27002**. [S.l.]: Van Haren, 2015.

JÚNIOR, L. C. S. Ransomware: análise técnica e prevenção. 2023.

KASPERSKY. **The cyber surge: Kaspersky detected 467,000 malicious files daily in 2024**. 2025. Disponível em: <https://www.kaspersky.com/about/press-releases/the-cyber-surge-kaspersky-detected-467000-malicious-files-daily-in-2024>.

KROTH, J. A. Estudo e análise de vulnerabilidades em serviços publicados na internet. 2018.

MICROSOFT. **Prevent malware infection**. 2025. <https://learn.microsoft.com/en-us/defender-endpoint/malware/prevent-malware-infection>. Acessado em: 30-10-2025.

MICROSOFT. **Virtualization-based Security (VBS)**. 2025. <https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs>. Acessado em: 30-10-2025.

MICROSOFT. **Windows 10 reaching end of support**. 2025. <https://learn.microsoft.com/en-us/lifecycle/announcements/windows-10-end-of-support>. Acessado em: 30-10-2025.

MICROSOFT. **Windows 10 support has ended on October 14, 2025**. 2025. <https://support.microsoft.com/en-us/windows/windows-10-support-has-ended-on-october-14-2025-2ca8b313-1946-43d3-b55c-2b95b107f281>. Acessado em: 30-10-2025.

MICROSOFT. **Windows 11 requirements**. 2025. <https://learn.microsoft.com/en-us/windows/whats-new/windows-11-requirements>. Acessado em: 30-10-2025.

OLIVEIRA, J. C. d. Trabalho de Conclusão de Curso (Graduação em Engenharia de Software), **Ransomware - Laboratório de Ataque do WannaCry**. [S.l.]: Universidade de Brasília (UnB), 2018.

PFLEEGER, C. P.; PFLEEGER, S. L. **Security in Computing**. 3rd. ed. [S.l.]: Prentice Hall Professional Technical Reference, 2002. ISBN 0130355488.

PROCURRI. **Global OS Market Share 2025: Key Stats, Trends, and Insights for Mobile and Desktop**. 2025. <https://www.procurri.com/knowledge-hub/global-os-market-share-2025-key-stats-trends-and-insights-for-mobile-and-desktop/>. Acessado em: 25-11-2025.

Rapid7. **Metasploit Framework**. 2025. <https://www.metasploit.com/>. Acessado em: 30-10-2025.

SILVA, L. P. d. O. Pentest baseado em imagens encase: uma análise de vulnerabilidades em servidores web. Pontifícia Universidade Católica de Goiás, 2024.

SOBRINHO, R. A. d. O. Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas), **Análise de Vulnerabilidades em Sistemas Operacionais**. [S.l.]: Fundação Educacional do Município de Assis (FEMA), 2024.

STALLINGS, W. **Network and internetwork security: principles and practice**. [S.l.]: Prentice-Hall, Inc., 1995.

STATCOUNTER. **Desktop Windows Version Market Share Worldwide**. 2025. <https://gs.statcounter.com/windows-version-market-share/desktop/worldwide>. Acessado em: 30-10-2025.

TANENBAUM, A. S.; BOS, H. **Modern operating systems**. [S.l.]: Pearson Education, Inc., 2015.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks**. 5th. ed. USA: Prentice Hall Press, 2010. ISBN 0132126958.

The MITRE Corporation. **CWE - Common Weakness Enumeration**. 2025. <https://cwe.mitre.org/>. Acessado em: 30-10-2025.

WHITMAN, M. E.; MATTORD, H. J. *et al.* **Principles of information security**. [S.l.]: Thomson Course Technology Boston, MA, 2009.