UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS CRATEÚS

CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LAÉRCIO LEVI SILVA DE MELO

OFUSCAÇÃO EM MALWARES BANCÁRIOS NO BRASIL: UM ESTUDO DE CASO SOBRE O TROJAN BANBRA

CRATEÚS

2026

LAÉRCIO LEVI SILVA DE MELO

OFUSCAÇÃO EM MALWARES BANCÁRIOS NO BRASIL: UM ESTUDO DE CASO SOBRE O TROJAN BANBRA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da computação do Campus Crateús da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da computação.

Orientador: Prof. Dr. Antonio Emerson Barros Tomaz.

CRATEÚS

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M485o Melo, Laércio Levi Silva de.
Ofuscação em malwares bancário no Brasil: um estudo de caso sobre o trojan Banbra / Laércio Levi Silva de Melo. – 2026.
55 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Crateús, Curso de Ciência da Computação, Crateús, 2026.
Orientação: Prof. Dr. Antônio Emerson Barros Tomaz.

1. Trojans bancários. 2. Malwares. 3. Técnicas de ofuscação. 4. Banbra. I. Título.

CDD 004

LAÉRCIO LEVI SILVA DE MELO

OFUSCAÇÃO EM MALWARES BANCÁRIOS NO BRASIL: UM ESTUDO DE CASO SOBRE O TROJAN BANBRA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da computação do Campus Crateús da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da computação.

Aprovada em: 27/01/2026.

BANCA EXAMINADORA

Prof. Dr. Antonio Emerson Barros Tomaz (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bruno de Castro Honorato Silva
Universidade Federal do Ceará (UFC)

Prof. Me. Francisco Anderson de Almada Gomes
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família, por todo o apoio ao longo dessa jornada. Foram todos fonte de inspiração e motivação para que eu chegasse até aqui.

AGRADECIMENTOS

Ao Prof. Dr. Antonio Emerson Barros Tomaz, pelo acompanhamento e orientação ao longo deste trabalho

À minha amiga e colega Camila Paiva, que esteve junta comigo ao longo de quase todo o período de minha formação acadêmica.

Ao meu pai Odilon Gomes de Melo e minha mãe Maria de Fátima Silva Sousa, por terem fornecido todo o apoio possível nessa minha jornada. Eles sempre se fizeram presentes em todos os momentos da minha vida.

À plataforma de inteligência de ameaças VirusTotal por ter cedido as amostras necessárias para a pesquisa.

À todos os meus colegas e amigos que durante este período de graduação, me apoiaram e me ajudaram de alguma forma.

"No que diz respeito ao empenho, ao compromisso, ao esforço, à dedicação, não existe meio termo. Ou você faz uma coisa bem feita ou não faz."

(Ayrton Senna)

RESUMO

O avanço tecnológico permitiu que muitos sistemas bancários passassem por um amplo processo de modernização, tornando a comunicação com os usuários mais intuitiva. Tais mudanças possibilitaram que, atualmente, a maioria das transações bancárias seja realizada de forma virtual. No entanto, juntamente com essas facilidades, também surgiram mecanismos maliciosos que visam coletar e utilizar informações pessoais dos usuários dessas plataformas financeiras, causando prejuízos econômicos e transtornos. Esses mecanismos, conhecidos como trojans bancários, são divididos em famílias que apresentam características e comportamentos distintos entre si. Uma das técnicas utilizadas por esses *malwares*, envolve o abuso de arquivos no formato *.cpl*, essa técnica consiste em um método de evasão de mecanismos de detecção de ameaças e de inicialização disfarçada. Com base em relatórios de plataformas especializadas, observa-se que o trojan bancário mais associado ao uso dessa técnica é o *Banbra*. Embora diversas técnicas de detecção para esse tipo de ameaça tenham sido desenvolvidas ao longo do tempo, ainda há uma significativa presença dessas famílias de *trojans*, mesmo com o passar dos anos. Este trabalho teve como objetivo realizar a análise e o mapeamento das características e técnicas de ofuscação do *trojan* bancário pertencente à família *Banbra*. A análise foi realizada por meio da observação dos *scripts* presentes nos arquivos maliciosos, com foco na identificação de técnicas únicas de ofuscação que dificultam a detecção por ferramentas tradicionais. A análise revelou uma sofisticada arquitetura de ofuscação em múltiplos estágios, na qual a camada externa consistia no uso de um *packer* para comprimir o código. Sob essa camada, foi identificada uma série de táticas de evasão ativas, como técnicas anti-análise e de reconhecimento de ambiente. A aplicação de engenharia reversa por meio de depuração permitiu contornar essas proteções e extrair o *payload* final. Essa investigação pode servir de base para estudos futuros voltados à criação de mecanismos de detecção mais eficazes, capazes de se adaptar às estratégias utilizadas por essa família de *trojans* bancários.

Palavras-chave: *trojans* bancários; *Banbra*; técnicas de ofuscação; detecção de ameaças; *.cpl*; análise de *malware*.

ABSTRACT

Technological advancement has enabled many banking systems to undergo an extensive modernization process, making communication with users more intuitive. Such changes have made it possible for the majority of banking transactions today to be conducted virtually. However, along with these conveniences, malicious mechanisms have also emerged, aiming to collect and use the personal information of users of these financial platforms, causing financial losses and disruptions. These mechanisms, known as banking trojans, are divided into families, each presenting distinct characteristics and behaviors. One of the techniques used by this type of malware involves the abuse of files in the .cpl format; this technique serves as a method for evading threat detection mechanisms and for disguised execution. Based on reports from specialized platforms, the banking trojan most frequently associated with the use of this technique is Banbra. Although various detection techniques for this type of threat have been developed over time, a significant presence of these trojan families persists, even after many years. This work aims to analyze and map the characteristics and obfuscation techniques of the banking trojan belonging to the Banbra family. The analysis was conducted by observing the scripts within the malicious files, focusing on identifying unique obfuscation techniques that hinder detection by traditional tools. The analysis revealed a sophisticated multi-stage obfuscation architecture, in which the outer layer consisted of the use of a packer to compress the code. Beneath this layer, a series of active evasion tactics were identified, such as anti-analysis techniques and environment reconnaissance. The application of reverse engineering through debugging made it possible to bypass these protections and extract the final payload. This investigation may serve as a foundation for future studies aimed at developing more effective detection mechanisms capable of adapting to the strategies employed by this family of banking trojans.

Keywords: *trojans* bankers; *Banbra*; obfuscation techniques; threat detection; CPL; malware analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

COOPS	Code Obfuscation through Overloading and Preservation of Semantics
DLL	*Dynamic Link Library*
IAT	Endereços de Importação
IP	*Internet Protocol*
MFA	Autenticção de Múltiplos Fatores
MitB	*Man-in-the-Browser*
MitM	*Man-in-the-Middle*
OEP	Ponto de Entrada Original
opcode	*Operation Code*
PE	*Portable Executable*
ProcMon	*Process Monitor*
SSL	*Secure Sockets Layer*
TLS	*Transport Layer Security*
VM	*Virtual Machine*

SUMÁRIO

1 INTRODUÇÃO

Com o constante avanço da Tecnologia da Informação, diversos setores vêm se adaptando para fornecer produtos que atendam às necessidades dos consumidores. Um desses setores é o de serviços, em especial o bancário, no qual há uma constante mudança das agências físicas para os meios digitais. Essa mudança ocorre principalmente por meio de aplicativos móveis que fornecem diversas funcionalidades para o usuário, eliminando a necessidade de deslocamentos frequentes até as agências bancárias presenciais.

Essa digitalização vem se refletindo no comportamento dos usuários. Segundo uma pesquisa realizada em 2024 pelo instituto Ipsos para o Nubank, cerca de 66% dos brasileiros utilizam aplicativos para realizar transações financeiras ou consultar saldo (Nubank, 2024). Além disso, uma pesquisa realizada pela Deloitte para a Febraban, também em 2024, mostra que 75% das operações bancárias feitas pelos brasileiros são realizadas pelo celular (Federação Brasileira de Bancos, 2024). Esses números reforçam como o sistema bancário no Brasil é extremamente digitalizado.

Com a migração dessas instituições bancárias para o meio digital, diversos agentes maliciosos têm desenvolvido mecanismos que buscam extrair informações das contas dos usuários, de forma que possam realizar transações financeiras em nome dessas pessoas, causando prejuízos financeiros e frustrações.

Dentre esses tipos de agentes maliciosos, destacam-se os *trojans*, ou cavalos de Troia, que possuem como característica o uso de programas de aparência confiável, mas que agregam *scripts* maliciosos em seu interior. Esses códigos são executados sem que o usuário tenha conhecimento, permitindo que o invasor tenha acesso a várias informações presentes no sistema que foi invadido. Esses *trojans* podem ser divididos em grupos denominados de famílias. Uma família de *malware* compreende todas as variantes de um *malware* que contêm comportamentos maliciosos comuns (Christodorescu, 2005 apud (BLACK *et al.*, 2017)).

Dentre essas técnicas, destaca-se a que se utiliza predominantemente da extensão de arquivo *.cpl* do Windows, tal técnica se caracteriza pela disseminação de *trojans* por meio de táticas como o *phishing*. Essa técnica consiste no envio de *e-mails* falsos contendo mensagens de texto ou *sites* que imitam instituições confiáveis. Esses conteúdos geralmente apresentam um tom de urgência, com o objetivo de persuadir o usuário a baixar e executar o arquivo anexado. A partir dessa ação, os *scripts* maliciosos são ativados, permitindo a captura de dados do usuário.

Embora não exista uma forma precisa de determinar quais *trojans* são os mais atu-

antes, conforme o Sindicato dos Trabalhadores em Processamento de Dados e Tecnologia da Informação do Estado de São Paulo (2025), foi divulgado um relatório da plataforma especializada Kaspersky, realizado entre julho de 2023 e julho de 2024 com base nos ciberataques bloqueados, apontando que 13,71% das infecções foram originadas pela família de *trojans* denominada *Banbra*, representando a maior taxa entre todas as famílias analisadas.

O *Banbra*, no entanto, possui duas definições distintas. Ele é predominantemente descrito como uma família específica de *trojans* bancários por empresas que são consolidadas no ramo, como a Karspersky e a Microsoft. A definição da Kaspersky, por exemplo, é apresentada de forma clara na pesquisa de Porolli e Ramos (2015) onde fala que essa família de malware é projetada para roubar informações pessoais dos clientes de bancos brasileiros. Da mesma forma a Microsoft mantém uma classificação específica como família para a detecção do *Trojan:Win32/Banbra* em sua enciclopédia de ameaças (Microsoft, 2024). Contudo, o termo também tem sido utilizado ocasionalmente, de forma genérica, por pesquisadores brasileiros para se referir a *trojans* bancários desenvolvidos ou amplamente distribuídos no Brasil. Nessa pesquisa iremos abordar o termo *Banbra* para se referir a família de ameaças bancárias.

Considerando que *trojans* bancários da família *Banbra* estão em atividade desde os anos 2000, com sua detecção sendo descrita pelo Laboratório de Pesquisa da ESET América Latina (2009), surge o interesse em compreender como, mesmo com os avanços das técnicas de detecção de programas maliciosos ao longo de mais de vinte anos, essa família continua sendo relevante no cenário nacional.

Segundo Ghaleb (2019), os antivírus tradicionais empregam diversos tipos de técnicas para detectar o *malware* ou qualquer atividades suspeita. A maioria dessas técnicas utiliza algoritmos de detecção baseados em assinaturas. Esse processo funciona buscando padrões específicos dessas assinaturas (*hashes*) em um grande banco de dados previamente catalogado. O Microsoft Windows Defender, por exemplo, pode ser considerado um antivírus tradicional.

Diante disso, este trabalho busca abordar a seguinte questão de pesquisa: *Quais são as técnicas de ofuscação do trojan bancário Banbra que dificultam sua detecção por antivírus tradicionais?*

1.1 Objetivo geral

Identificar e categorizar as técnicas de ofuscação empregadas por a família de *trojans* bancários *Banbra*, com o propósito de elucidar como essas técnicas contornam os mecanismos

de detecção de antivírus tradicionais.

1.2 Objetivos específicos

- Realizar a análise das técnicas de ofuscação em nível estático, ou seja, sem executar o código, examinando diretamente seu conteúdo para identificar padrões ou indícios de ofuscação.
- Realizar a análise das técnicas de ofuscação em nível dinâmico, ou seja, análise em tempo de execução em um ambiente controlado.
- Catalogar os resultados de cada nível de análise.

1.3 Contribuição

A principal contribuição proposta por este trabalho é a criação de um mapeamento detalhado e atualizado das técnicas de ofuscação utilizadas pela família de *trojans* bancários *Banbra*, uma ameaça cibernética de alta relevância e persistência no cenário brasileiro.

Considerando que análises técnicas aprofundadas sobre este tipo de ameaça precisam ser constantemente renovadas devido à rápida evolução das táticas de ataque, esta pesquisa busca preencher uma lacuna de conhecimento técnico específico. Dessa forma, as contribuições esperadas podem ser divididas em duas frentes:

Para a comunidade de cibersegurança: Ao final do estudo, pretende-se fornecer um material de referência prático para analistas de *malware* e desenvolvedores de soluções de defesa. A catalogação das táticas de evasão do *Banbra* poderá auxiliar na otimização de assinaturas de detecção, no desenvolvimento de heurísticas mais eficazes e no aprimoramento de estratégias contra *malwares* com características semelhantes.

Para a pesquisa acadêmica: O trabalho visa oferecer uma base de dados que poderá fundamentar estudos futuros. Pesquisas posteriores poderão utilizar os resultados aqui apresentados para, por exemplo, mensurar a eficácia de novas ferramentas de detecção, desenvolver modelos de aprendizado de máquina para identificar código ofuscado ou analisar a evolução das táticas de ataque no ecossistema de *malware* brasileiro.

2 FUNDAMENTAÇÃO TEÓRICA

Com o intuito de fornecer um melhor embasamento sobre as técnicas e conceitos utilizados na área, este capítulo tem como objetivo apresentar aspectos relevantes dos temas que fundamentam esta pesquisa, a fim de proporcionar uma melhor compreensão do objeto de estudo.

2.1 *Malware*

A presença de *softwares* maliciosos em sistemas computacionais constitui uma das principais preocupações no campo da segurança da informação. Nesse contexto, o termo *malware* é definido como um programa inserido em um sistema, geralmente de forma secreta, com a intenção de comprometer a confidencialidade, a integridade ou a disponibilidade dos dados, aplicativos ou sistema operacional da vítima, ou de incomodá-la ou perturbá-la de alguma outra forma (NIST, 2011).

Para que a ameaça representada pelo *malware* se concretize, é essencial que exista, primeiramente, um meio de se instalar no sistema da vítima. Essa fase de infiltração, conhecida como *vetor de infecção*, é fundamental no ciclo de ataque. Os métodos de transmissão apresentam grande variedade e evoluíram significativamente ao longo do tempo. O código malicioso pode ser transmitido de muitas maneiras. As mídias mais comuns são as mídias removíveis, como *pen drives*; arquivos baixados da Internet e anexos de *e-mail*. Um tipo de ataque particularmente insidioso é o *drive-by download*, no qual o computador de um cliente é infectado apenas por visitar uma página *web* maliciosa (PFLEEGER; PFLEEGER, 2015).

Conforme as definições apresentadas por Stallings e Brown (2018), o termo *malware* engloba uma vasta gama de programas maliciosos que, embora compartilhem a mesma intenção de causar danos, se diferenciam nos métodos de propagação, nos mecanismos de ativação e no comportamento no sistema infectado. Já uma abordagem melhor para classificar um *software* malicioso é agrupá-lo com base em seu mecanismo de propagação e nas ações que ele realiza. Uma distinção chave é se o *software* malicioso requer um programa hospedeiro. Com base nisso podemos fazer as distinções da seguinte forma:

- **Vírus:** É um pedaço de código que é prefixado, anexado ou de outra forma embutido em algum programa executável existente.
- ***Worm*:** É um programa que pode ser executado de forma independente e propagar uma

versão completa e funcional de si mesmo para outros hospedeiros em uma rede.

- **Cavalo de Troia:** É um programa que aparenta ter uma função útil, mas também possui uma função oculta e potencialmente maliciosa que burla os mecanismos de segurança, às vezes explorando as autorizações legítimas de uma entidade do sistema que o invoca.
- ***Ransomware*:** Foca na extorsão financeira, criptografando os dados das vítimas e exigindo um pagamento de resgate.
- ***Spyware*:** Atua de forma furtiva, monitorando as atividades do usuário e coletando informações confidenciais, como senhas e dados de navegação.

Pode-se então observar que o universo dos *malwares* é bastante diversificado, sendo que cada categoria apresenta objetivos e métodos de propagação distintos. Dentre os *malwares* citados, os cavalos de Troia, mais conhecidos como *trojans*, destacam-se por suas técnicas de engano e ataques direcionados. Dado o seu impacto, a seção seguinte abordará uma análise de uma classe específica dentro dessa categoria: os chamados *trojans* bancários.

2.2 *Trojan* bancário

Nomeados em homenagem ao famoso Cavalo de Troia grego que invadiu Troia, os *trojans* são instalados em arquivos aparentemente inócuos para induzir o usuário a executá-los. Este código malicioso opera de forma oculta, sem indicar vestígios de sua atividade ao usuário, enquanto executa uma série de processos anômalos em segundo plano (KUMARI *et al.*, 2024).

Trojans bancários são uma subclasse dentro dos *trojans*, que são responsáveis apenas por se aproveitar de vulnerabilidades para fazer ataques financeiros. Conforme Kumari *et al.* (2024), o *trojan* bancário foi especialmente criado para minar as defesas em torno de sistemas bancários online e roubar informações financeiras confidenciais, como senhas de login, números de cartão de crédito, códigos PIN para caixas eletrônicos de bancos e Números de Identificação Pessoal (PINs).

Essa evolução dos *trojans* fazendo a criação de uma categoria especializada de *malware*, como os *trojans* bancários, surgiu como uma forma de adaptação à evolução da segurança dos serviços financeiros online. Antes disso, os *malwares* utilizavam apenas *key-loggers* simples, que eram eficazes na captura de dados estáticos. No entanto, com a adoção de mecanismos de defesa, como Autenticção de Múltiplos Fatores (MFA), teclados virtuais e tokens de segurança, o uso dessas técnicas passou a ser ineficaz. Os cibercriminosos então desenvolveramum método de ataque que não apenas roubasse informações, mas que pudesse

operar dentro do dispositivo do usuário de forma legítima, contornando ativamente as novas camadas de proteção. Esses *trojans* são capazes de realizar ataques *Man-in-the-Browser* (MitB), nos quais podem interceptar e manipular as transações do usuário com os *sites* de *online banking* (SOOD ADITYA K.; ENBODY, 2011).

Ultimamente, esses *trojans* bancários vêm ganhando força, principalmente no cenário brasileiro. De acordo com um relatório da Kaspersky, mencionado pelo Ministério Público do Estado de Mato Grosso (2024), o país se destaca no número de ataques por *malware* bancário na américa latina. Conforme aponta Kaspersky Lab (2023), o Brasil já é o país mais atacado por *trojan* bancário (*desktop*) no mundo e o quinto quando leva-se em consideração apenas esses golpes no celular.

Diante desse cenário preocupante, surge a necessidade de compreender o funcionamento dos mecanismos utilizados por esses *malwares*. A subseção a seguir apresentará, de forma introdutória, o funcionamento da técnica conhecida como MitB.

2.2.1 Man-in-the-Browser (MitB)

Esse tipo de ataque representa o núcleo de um *trojan* bancário. É justamente esse componente que os *malwares* financeiros buscam proteger contra a detecção por mecanismos de segurança. Tommasi *et al.* (2021) diz que, ataques do tipo MitB são uma forma de ameaça à Internet relacionada ao conhecido ataque *Man-in-the-Middle* (MitM). Ele combina o uso de abordagens de *phishing* com uma tecnologia de cavalo de Troia para modificar as páginas da *web*, o conteúdo das transações e inserir transações adicionais. Tudo é feito com a vítima e o aplicativo *web* hospedeiro, sem que eles estejam cientes do uso deste método. Uma vez ativado, um *trojan* MitB pode interceptar e manipular qualquer informação que um usuário submeta *online* em tempo real.

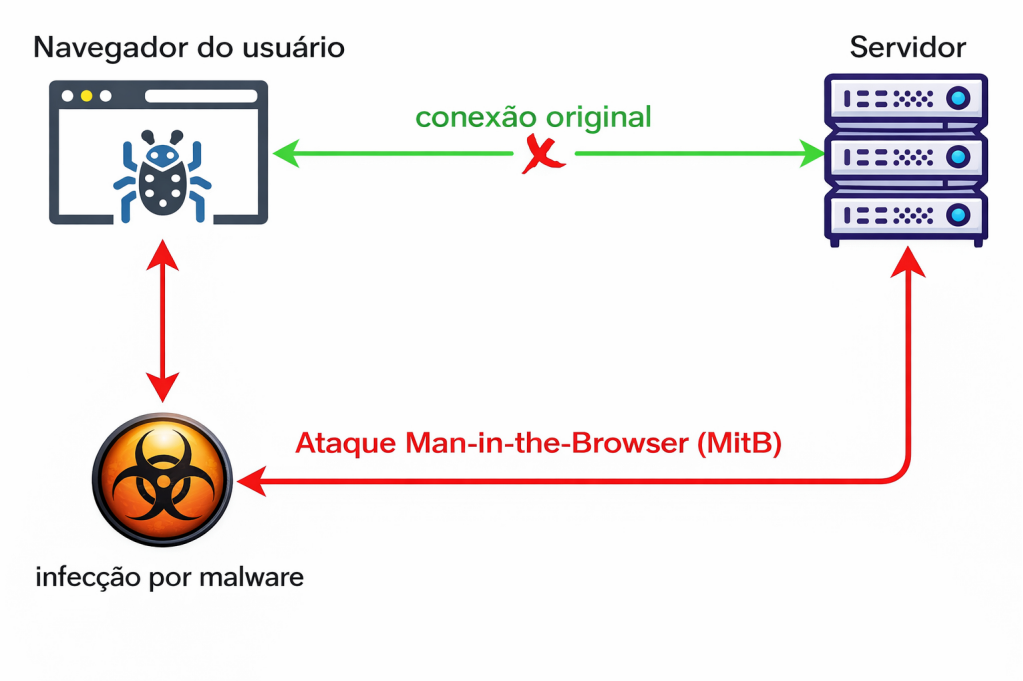
O que torna o ataque MitB uma ameaça tão grande é a sua capacidade de contornar a criptografia *Transport Layer Security* (TLS)/*Secure Sockets Layer* (SSL), que é a principal defesa de comunicação *online*. Conforme destaca Tommasi *et al.* (2021), o MitB é capaz de ver a informação dentro do navegador, uma vez que nenhuma criptografia ocorre dentro dele, contornando facilmente os mecanismos de segurança nos quais todos nós confiamos.

Essa eficiência se dá pela diferença em relação ao ataque MitM, pois enquanto o MitM intercepta o tráfego na rede, o MitB opera diretamente no *endpoint*, injetando seu código no processo do navegador. Por esse motivo, segundo Tommasi *et al.* (2021), o MitB possui mais

vantagens em relação ao MitMem termos de facilidade de execução e capacidade de contornar defesas.

A Figura 1 ilustra o funcionamento do ataque MitB. Nela, observa-se que o *malware* intercepta as credenciais do usuário diretamente no navegador da própria máquina, antes que esses dados sejam acessados pelas instituições.

Figura 1 – Fluxograma do ataque *Man-in-the-Browser*



Fonte: Adaptado de Tommasi *et al.* (2021)

Pode-se observar, portanto, que a técnica MitB é o mecanismo central que possibilita a fraude financeira, contornando defesas robustas ao operar diretamente no *endpoint* da vítima. Contudo, a eficácia de um motor de ataque tão poderoso depende inteiramente do sucesso da infecção inicial e da subsequente execução do *malware*. Nesse sentido, diversos *trojans* bancários, incluindo o *Banbra*, utilizam como tática característica o abuso de arquivos do Painel de Controle do Windows (arquivos *.cpl*) como vetor de evasão, isto é, um método projetado para contornar as defesas do sistema e ocultar a execução inicial do *malware*. A seção seguinte, portanto, visa detalhar o funcionamento desta técnica e sua importância na arquitetura dos *trojans* bancários modernos.

2.3 Arquivos *.cpl*

Uma das maneiras dos *trojans* bancários serem executados é por meio de arquivos do Painel de Controle do Windows, que possuem a extensão *.cpl*. Apesar de parecerem apenas itens de configuração, na verdade, tratam-se de bibliotecas de vínculo dinâmico ou *Dynamic Link Library (DLL)*. Tal fato é explorado por atacantes para executar seu código malicioso por meio de processos legítimos do sistema operacional.

Conforme mencionado por The MITRE Corporation (2025), o processo do Painel de Controle do Windows (*control.exe*) executa itens do Painel de Controle, que são DLLs registradas. Adversários podem abusar do *control.exe* para intermediar a execução de código malicioso. Os Arquivos *.cpl* maliciosos podem ser criados exportando uma função (*CPlApplet*), enquanto isso, estes mesmos arquivos podem ser executados a partir da linha de comando ou por um duplo clique no arquivo.

Esses arquivos maliciosos *.cpl* funcionam como encapsuladores do motor de ataque MitB. Eles ocultam a ferramenta maliciosa real, fazendo o arquivo parecer inofensivo e legítimo. Essa técnica contribui para dificultar a detecção por soluções de segurança, ao mesmo tempo em que aproveita mecanismos nativos do sistema.

Com a compreensão dos motivos que tornam os arquivos *.cpl* um vetor de ataque tão eficaz, a próxima etapa é entender como os atacantes fazem com que esses arquivos cheguem ao sistema da vítima, especialmente considerando que sua execução depende da interação do usuário (ação de dar um duplo clique com o mouse, por exemplo).

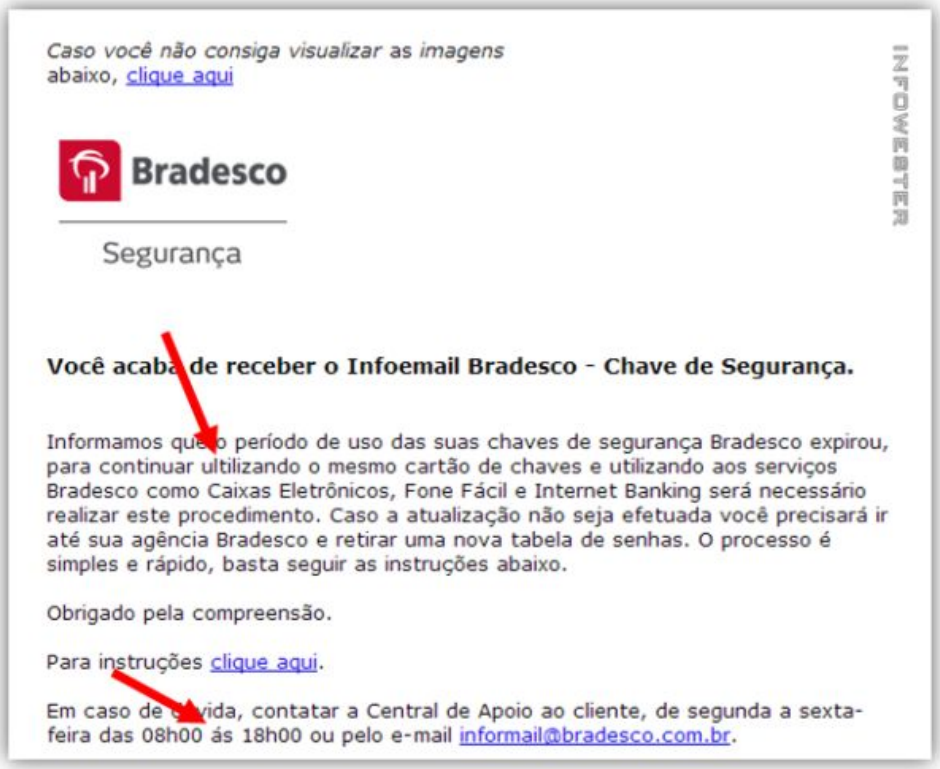
Para persuadir suas vítimas a executar os arquivos *.cpl* maliciosos e se infectarem, os cibercriminosos enviam *e-mails* falsos, que são seu principal meio de propagação do *malware*. Portanto, eles fazem uso de técnicas de **engenharia social** para enganar os usuários, levando-os a acreditar que o arquivo *.cpl* anexado à mensagem é um documento com informações úteis (POROLLI; RAMOS, 2015).

O conteúdo desses *e-mails* é altamente variado. Conforme o detalhamento feito na análise de Porolli e Ramos (2015), as campanhas de *phishing* que distribuem *malware* *.cpl* no Brasil utilizam iscas altamente contextualizadas com a realidade local. Os *e-mails* fraudulentos frequentemente se passam por documentos financeiros como cotações, faturas ou recibos, informações sobre dívidas ou situação bancária e, de forma notória, instrumentos de pagamento digitais como o boleto bancário ou a nota fiscal eletrônica.

A Figura 2 a seguir demonstra um exemplo de um *e-mail* de *phishing* que é utilizado

como forma de convencimento para as vítimas, contendo um teor apelativo.

Figura 2 – Exemplo de *site* falso do Bradesco utilizado em ataque de *phishing*



Fonte: InfoWester (2019)

A combinação da persuasão por meio da engenharia social com um vetor de execução que aparenta ser inofensivo, como é o caso dos arquivos *.cpl*, torna o cenário cada vez mais desafiador para os mecanismos de defesa do sistema, como os antivírus. Diante disso, torna-se importante compreender como as ferramentas de segurança tradicionais operam para detectar esses tipos específicos de ameaças. A seção seguinte visa detalhar os métodos de análise e captura desses tipos de *malwares*.

2.4 Mecanismos de detecção de antivírus tradicionais

Os *softwares* responsáveis pela detecção de ameaças (como os antivírus tradicionais) têm como característica central a análise de arquivos ou programas, comparando-os com padrões previamente conhecidos. Caso haja correspondência com esses padrões, o objeto analisado é bloqueado. Essa é apenas a premissa mais simples. Com o passar do tempo, essas ferramentas evoluíram e passaram a adotar novas formas de realizar essas análises. Podemos destacar, então, diversos métodos de análise diferentes que visam aumentar significativamente as chances de

captura de entidades maliciosas.

Com base nisso, podemos dividir essas técnicas de análise da mais simples à mais robusta. Segundo Mishra (2025), um dos métodos mais comuns é a detecção baseada em assinaturas, que se baseia em um banco de dados de assinaturas de *malware* conhecidas, sequências de dados exclusivas que correspondem a um *malware* específico. Essa técnica verifica arquivos e compara suas assinaturas com o banco de dados para detectar ameaças. Embora a detecção baseada em assinaturas seja eficaz para identificar ameaças conhecidas, ela apresenta limitações notáveis, principalmente contra *malware* polimórfico e metamórfico. Esse processo de análise por assinatura geralmente pode ser dividido em três formas: análise estática, análise dinâmica e análise híbrida.

A **análise estática** se preocupa com a varredura de códigos sem execução. O código é desmontado usando ferramentas de desmontagem, em que informações de baixo nível são extraídas. Esse processo permite a análise da estrutura de várias partes do código, o que aumenta a probabilidade de detectar quaisquer linhas maliciosas injetadas. No entanto, pode falhar se uma ofuscação de código for aplicada (GHALEB, 2019).

A **análise dinâmica**, ou análise comportamental, funciona executando o arquivo para investigar seu comportamento no sistema operacional e, em seguida, explorar como o sistema operacional interagiria ou reagiria a essa ação. Arquivos infectados com código malicioso também são analisados usando máquinas virtuais ou *sandbox* [1] (GHALEB, 2019).

Já a **análise híbrida** busca realizar a combinação desses dois tipos de análise, obtendo uma maior taxa de detecção de ameaças.

Com o objetivo de contornar esse problema de detecção apenas de ameaças já catalogadas, surgiu outro método, que é a detecção heurística. De acordo com Mishra (2025), a análise heurística avalia o comportamento e as características dos arquivos para descobrir ameaças em potencial, mesmo que suas assinaturas sejam desconhecidas; no entanto, esse método pode gerar falsos positivos, levando a interrupções em processos legítimos. Vale considerar também que o mecanismo de detecção heurística consome uma alta quantidade de recursos do computador, o que pode não ser interessante quando se busca a eficiência e eficácia em todos os sentidos.

Com o intuito de burlar os mecanismos de detecção empregados pelos antivírus tradicionais, os atacantes desenvolvem constantemente novas técnicas de ofuscação de código,

[1] Ambiente isolado usado para executar e observar o comportamento de programas sem afetar o sistema real.

injeção de processos e uso de *packers*, que são ferramentas utilizadas para comprimir e/ou criptografar o código malicioso original, o que dificulta sua detecção por esses métodos convencionais. A próxima seção tem como objetivo apresentar, de forma geral, alguns dos métodos de ofuscação de código utilizados por desenvolvedores de *malwares*.

2.5 Técnicas de ofuscação de código

A ofuscação de código destaca-se por sua versatilidade e eficiência na ocultação de mecanismos maliciosos. Essa técnica é utilizada para reduzir a legibilidade do código e dificultar sua compreensão, sem alterar suas funcionalidades. Ela não depende de vulnerabilidades específicas do sistema, mas sim da manipulação do próprio código para dificultar sua análise. Segundo You e Yim (2010), originalmente, essa técnica visava proteger a propriedade intelectual dos desenvolvedores de *software*, mas tem sido amplamente utilizada por autores de *malware* para burlar a detecção. Ou seja, para escapar dos *scanners* antivírus, os *malwares* evoluem seu corpo para novas gerações por meio da técnica de ofuscação.

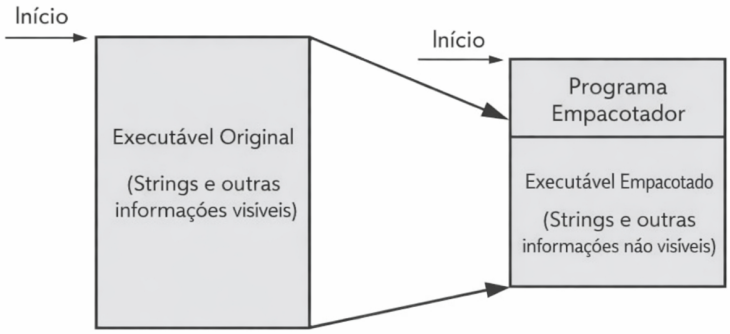
Existem diferentes técnicas de ofuscação empregadas por atacantes. Dentre elas, uma das mais amplamente utilizadas por *malwares* é o *packing* ou empacotamento. Um *packer* é uma ferramenta que comprime e/ou criptografa um executável com o objetivo de mascarar o código malicioso, tornando-o irreconhecível para *softwares* antivírus e dificultando sua análise por especialistas em segurança. Um arquivo malicioso que passou por esse processo é chamado de empacotado (*packed*).

Esse processo de empacotamento impede que o conteúdo real do *malware* seja analisado diretamente. Como explicam Sikorski e Honig (2012), quando o programa empacotado é executado, um pequeno programa *wrapper* também é iniciado para descompactar o arquivo e, em seguida, executar o conteúdo desempacotado. Quando se analisa um programa empacotado de forma estática, apenas o pequeno programa *wrapper* pode ser inspecionado. Isso significa que a maior parte do código malicioso permanece oculta até o momento de sua execução, dificultando a sua identificação por ferramentas de análise estática.

A Figura 3 mostra, à esquerda, um executável original, com todas as *strings* e informações visíveis para análise. Já à direita, exibe um arquivo empacotado (*packed executable*), que contém um *wrapper program* responsável por iniciar a execução, além do executável compactado, cujos dados permanecem invisíveis à análise estática.

No entanto, muitas vezes o desempacotador permanece constante ao longo das

Figura 3 – Funcionamento do empacotamento de um executável (*packing*), ocultando informações do código original



Fonte: Adaptado de Sikorski e Honig (2012)

gerações de *malwares*, o que permite que as ferramentas de detecção identifiquem esses padrões no código e bloqueiem sua execução. Para contornar esse problema, surgiram outras técnicas, conhecidas como *malwares* oligomórficos e polimórficos.

A primeira tentativa foi o *malware* oligomórfico, capaz de alterar ligeiramente seu desempacotador. No entanto, esse *malware* pode gerar no máximo algumas centenas de desempacotadores diferentes, ainda podendo ser detectado com assinaturas. Para superar a limitação, os autores de *malware* desenvolveram o *malware* polimórfico. Esse *malware* polimórfico consegue criar um número incontável de desempacotadores distintos com a ajuda de métodos de ofuscação, incluindo inserção de código morto, reatribuição de registro e assim por diante (YOU; YIM, 2010).

Porém, mesmo com esses novos métodos de ofuscação, com o passar do tempo, os antivírus tradicionais começaram a detectar esses tipos de ameaça, forçando os desenvolvedores de mecanismos maliciosos a se adaptarem novamente. Com o intuito de contornar essas análises realizadas pelos mecanismos de detecção, surgiu uma nova forma de comportamento dos *malwares*, denominada *malware* metamórfico.

Cada vez que um *malware* metamórfico é executado, ele altera o código de operação ou *Operation Code (opcode)* carregado na memória e, em seguida, grava uma nova versão de si mesmo no arquivo infectado do *host*. O *malware* mantém seu comportamento malicioso

sem jamais apresentar a mesma sequência de códigos nativos na memória. Assim, abordagens convencionais baseadas em assinaturas exigiriam a varredura de um programa em busca de milhões de assinaturas diferentes para detectar uma única ameaça (O'KANE *et al.*, 2011).

Além de dificultarem a análise estática, as ameaças modernas, incluindo *trojans* bancários, utilizam estratégias para detectar e reagir à análise dinâmica. Essas técnicas de anti-análise, como *anti-debugging* e *anti-VM/sandbox*, buscam identificar quando os *trojans* estão sendo executados em um ambiente controlado, como uma máquina virtual (*Virtual Machine* (VM)) ou *sandbox*, provavelmente sendo utilizado como objeto de estudo por pesquisadores e *softwares* antivírus.

Para evitar a detecção, o reconhecimento do ambiente d,execução é frequentemente incorporado ao *malware*. Ao fazer isso, esse *malware* pode determinar a presença de ganchos, adaptar-se e possivelmente alterar completamente sua execução em tempo real, impedindo assim sua análise dinâmica em tempo real. Isso pode ser realizado monitorando artefatos de ambiente (APOSTOLOPOULOS *et al.*, 2021).

Compreender essas técnicas de ofuscação é fundamental para decifrar as táticas empregadas por *malwares* modernos. É justamente a combinação e a sofisticação dessas camadas de evasão que determinam a resiliência de uma ameaça. Dentre essas ameaças, as que se destacam no cenário dos *trojans* bancários pertencem à família *Banbra*. A seção seguinte tem como objetivo apresentar as principais informações e características relacionadas a essa família de *malwares*.

2.6 *Trojan Banbra*

Dentro do ecossistema de *malwares* financeiros no Brasil, a família de *trojans* bancários que mais se destaca é a *Banbra*, sendo uma das ameaças mais recorrentes ao longo dos anos. Essa recorrência é evidenciada por detecções registradas em diferentes períodos, como pelo Laboratório de Pesquisa da ESET América Latina (2009), por Porolli e Ramos (2015) e pelo Sindicato dos Bancários de Ponta Grossa e Região (2024). Apesar de já se encontrar atuando a mais de uma década, ainda se destaca como o *trojan* bancário mais detectado segundo um relatório recente realizado pela empresa Karspersky, que é especializada em segurança digital, tal relatório é citado pelo Sindicato dos Trabalhadores em Processamento de Dados e Tecnologia da Informação do Estado de São Paulo (2025).

Essa família de *malware* é projetada para roubar informações pessoais dos clientes

de bancos brasileiros. Os métodos e tecnologias utilizados por este *malware* são geralmente grosseiros. Escrito em *Delphi* ou *.NET*, o *malware* usa formulários fraudulentos para obter as informações necessárias para contornar a autenticação de dois fatores (AO Kaspersky Lab, 2016).

Como mencionado, sua arquitetura possui várias fases, iniciando com a propagação por meio de campanhas de *phishing* altamente contextualizadas, que induzem a vítima a executar seu *downloader* inicial, geralmente um arquivo *.cpl*. Em seguida, ocorre o desempacotamento e a execução de seu motor de ataque, que é o *malware* em si, responsável por realizar a coleta dos dados da vítima.

Devido à sua constante evolução e ao emprego de múltiplas camadas de ofuscação, o *Banbra* serve como um excelente estudo de caso para analisar como os *trojans* bancários modernos dificultam sua detecção por antivírus tradicionais, sendo, portanto, o objeto central de investigação deste trabalho.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados à presente pesquisa, que formam a base para o estudo das técnicas de ofuscação em *trojans* bancários. Esses trabalhos foram encontrados por meio da realização de pesquisas relacionadas a *trojans* bancários nas principais revistas acadêmicas. Dentre eles, podemos destacar os seguintes trabalhos.

3.1 Análise de técnicas baseadas em assinaturas em produtos antivírus

O artigo de Ghaleb (2019) oferece uma revisão detalhada sobre uma técnica fundamental dos *softwares* antivírus que é a detecção baseada em assinaturas. O objetivo principal do estudo é realizar uma revisão e uma avaliação crítica dos algoritmos de detecção baseados em assinaturas, utilizados em produtos antivírus. Este trabalho busca classificar as diferentes abordagens (estática, dinâmica e hibrida), fazer uma análise dos algoritmos de correspondência de *strings* e discutir os *trade-offs* de design, como a complexidade de tempo, custo de memória e taxas de detecção.

A metodologia utilizada é uma revisão de literatura. Ghaleb (2019) analisa pesquisas existentes sobre algoritmos de detecção e *string-matching* (correspondência de *strings*), analisando suas implementações, eficiência e aplicabilidade no contexto de detecção de *malware*. O trabalho não apresenta uma nova análise experimental, mas resume e organiza o conhecimento já consolidado na área.

Ao final do artigo, conclui-se que a detecção baseada em assinaturas, apesar de ser um método estabelecido e eficaz para ameaças conhecidas, possui limitações de desempenho e, de eficácia contra *malwares* novos ou polimórficos. Ghaleb (2019) destaca que, embora algoritmos eficientes possam existir, a necessidade de manter um banco de dados de assinaturas constantemente atualizado e a incapacidade de detectar ameaças desconhecidas são seus principais pontos fracos. É apontado a necessidade de abordagens heurísticas e comportamentais para complementar essa técnica tradicional.

Tal artigo fornece uma ótima base para a compreensão do funcionamento dos antivírus tradicionais. No entanto, seu foco está apenas na perspectiva do defensor, o que não é suficiente para a presente pesquisa, uma vez que não há um aprofundamento nas técnicas de ofuscação, ou seja, na visão do atacante.

3.2 Uma análise abrangente de técnicas de ofuscação de software

Ramanujam (2023) realiza um estudo que tem como objetivo avaliar a eficiência de diversas estratégias de ofuscação na proteção dos softwares contra engenharia reversa e manipulção por atores maliciosos. Este estudo visa consolidar o conhecimento sobre diferentes técnicas, tais como, ofuscação de dados, ofuscação de controle de fluxo e ofuscação de *layout*.

Quanto à metodologia utilizada na confecção do trabalho, Ramanujam (2023) realiza uma revisão de literatura, ou seja, os autores não conduzem uma nova análise experimental de *malware*, mas sim um levantamento e uma síntese das técnicas de ofuscação documentadas em pesquisas e publicações anteriores. O artigo categoriza essas técnicas e discute suas aplicações, limitações e os avanços mais recentes na área.

A conclusão do estudo é de que uma única técnica de ofuscação é insuficiente para garantir uma proteção robusta. Os autores recomendam a diversificação e a aplicação de múltiplas camadas de ofuscação de forma a maximizar a segurança dos dados. Ramanujam (2023) destaca a eficácia de métodos que alteram o fluxo de controle do programa em mais de 90%, como a técnica Code Obfuscation through Overloading and Preservation of Semantics (COOPS). A pesquisa apresentada neste trabalho no entanto, mostra que sua abordagem ampla e generalista apresenta limitações claras quando aplicada ao escopo específico desta pesquisa, onde o foco está na investigação aprofundada das técnicas de ofuscação empenhadas pelas ameaças bancárias.

3.3 Injeção na *web* e *trojans* bancários *malware* - uma revisão sistemática da literatura

O objetivo principal do trabalho de Nelson *et al.* (2023) é analisar de forma sistemática a literatura disponível, com o intuito de descrever o ecossistema dos ataques de injeção *web* em *trojans* bancários. A pesquisa busca responder a perguntas específicas, tais como: quem são os atores por trás desses ataques, quais famílias de *malware* utilizam a técnica, que tipos de injeção *web* são empregados e quem são as vítimas.

A metodologia adotada é uma Revisão Sistemática de Literatura.Nessa pesquisa, os autores realizaram uma busca estruturada em diversas plataformas de dados acadêmicas (como IEEE Xplore e ACM Digital Library) e selecionaram 14 artigos relevantes, nas quais foram analisados com base em um esquema de classificação customizado, baseado no modelo *Diamond Model of Intrusion Analysis*, que estrutura um evento de ataque em quatro pilares (adversário, capacidade, infraestrutura e vítima), e o uso do *framework* MITRE ATT&CK, uma base de

conhecimento global que cataloga as táticas, técnicas e procedimentos dos atacantes.

A análise feita indicou que a literatura acadêmica sobre o tema é focada principalmente na perspectiva do atacante, com poucos estudos sendo voltados para a vítima. Os resultados dizem que a técnica de *web* injection é o principal mecanismo para roubo de credenciais em *trojans* bancários modernos. Além disso, o estudo identifica *Zeus, SpyEye* e *Citadel* como as famílias de *malware* mais citadas e documentadas, servindo de base para outras ameaças.

O trabalho de Nelson *et al.* (2023), no entanto, por se limitar a uma revisão sistemática, falha em se aprofundar nas técnicas de ofuscação em si, concentrando-se apenas na funcionalidade, ou seja, na catalogação dos mecanismos de ataque, como a própria injeção *web*. Além disso, o estudo não realiza uma análise técnica de baixo nível em amostras reais que permita demonstrar, na prática, como as técnicas de ofuscação são implementadas.

3.4 *Malware .cpl* no Brasil: entre *trojans* bancários e *e-mails* maliciosos

Esse é um dos trabalhos mais significativos e diretamente relacionados à presente pesquisa. Este estudo realiza uma análise técnica aprofundada do ecossistema de *malwares* que utilizam arquivos do Painel de Controle *(.cpl)* como principal vetor de ataque no Brasil.

O principal objetivo do estudo da empresa de segurança ESET é examinar a o método das campanhas de *malware* baseadas em*.cpl* no Brasil. Nesta pesquisa, os autores buscam definir o que são arquivos *.cpl*, como são utilizados por cibercriminosos, detalhar suas campanhas de propagação que ocorrem majoritariamente via *phishing*, e realizar uma análise técnica do *payload* malicioso, incluindo as técnicas de ofuscação e os *trojans* bancários que são entregues às vítimas.

A metodologia empregada é de natureza técnica, baseada em engenharia reversa e análise de *malware*. Os pesquisadores analisaram uma ampla coleção de amostras de *malware* no formato *.cpl*, utilizando ferramentas de análise estática e dinâmica para desmontar, depurar e compreender o funcionamento interno dos artefatos. Trata-se, portanto, de um estudo de caso técnico sobre uma ameaça específica, alinhado ao objetivo da presente pesquisa.

O estudo indica que os arquivos *.cpl* funcionam primariamente como *downloaders*, onde sua função é baixar e executar um *trojan* bancário. A propagação é amplamente realizada por meio de *e-mails* de *phishing* com iscas de engenharia social altamente adaptadas para o cenário brasileiro, como por exemplo boletos bancários e notas fiscais eletrônicas. A análise feita identificou que a maioria das amostras é escrita em *Delphi* e emprega técnicas de ofuscação como

algoritmos de criptografia de *strings* que são baseados em operações XOR e subtrações e além disso utilizando truques anti-VM para dificultar a análise em ambientes virtualizados. Por fim, Porolli e Ramos (2015) concluem que o *Win32/TrojanDownloader.Banload* e *o Win32/Spy.Banbra* são as famílias predominantes neste ecossistema.

O relatório de Porolli e Ramos (2015) oferece uma contribuição relevante para a presente pesquisa, apresentando uma abordagem metodológica semelhante. No entanto, o estudo apresenta algumas lacunas que merecem destaque. Por ter sido publicado em 2015, o trabalho encontra-se desatualizado, especialmente no campo da segurança da informação, onde o cenário de ameaças evolui de forma acelerada. Nesse contexto, torna-se necessária uma nova análise que atualize os dados com base na realidade atual. Além disso, o foco principal do estudo de Porolli e Ramos (2015) está no vetor de infecção (*.cpl*), enquanto o presente trabalho concentra-se no *payload* final, especificamente, o *trojan* bancário *Banbra*, o que marca uma diferenciação importante entre os dois focos de pesquisa.

A seguir, a Tabela 1 apresenta um resumo dos trabalhos selecionados que formam a base teórica e contextual desta pesquisa. A tabela compara os objetivos, metodologias e as principais limitações de cada estudo em relação ao escopo do presente trabalho.

Tabela 1 – Resumo dos Trabalhos Relacionados

Trabalho (Autores, Ano)	**Objetivo**	**Metodologia**	**Limitação Principal**
Análise de Detecção por Assinatura (Alasli e Ghaleb, 2019)	Realizar uma revisão e avaliação crítica dos algoritmos de detecção baseados em assinaturas, utilizados em produtos antivírus.	Revisão de literatura, focada na análise de pesquisas existentes sobre algoritmos de detecção e correspondência de strings.	Foco exclusivo no ponto de vista do defensor.
Análise de Técnicas de Ofuscação de Software (Ramujan e Devgude, 2023)	Avaliar a eficiência de diversas estratégias de ofuscação na proteção de softwares contra engenharia reversa e manipulação.	Revisão de literatura, com levantamento e síntese das técnicas de ofuscação documentadas em publicações anteriores.	Abordagem generalista e voltada para a proteção de software legítimo (lado do defensor).
Revisão sobre Injeção Web em Trojans (Nelson et al., 2023)	Analisar de forma sistemática a literatura para descrever o ecossistema dos ataques de injeção web em trojans bancários.	Revisão sistemática da literatura, com busca estruturada em bases de dados acadêmicas e seleção de 14 artigos relevantes para análise.	Aborda superficialmente as técnicas de ofuscação e evasão.
Análise de Malware .cpl no Brasil (Porolli e Ramos, 2015)	Examinar o método das campanhas de malware baseadas em arquivos .cpl no Brasil, frequentemente utilizadas por trojans bancários.	Análise de múltiplas amostras de malware .cpl utilizando uma metodologia híbrida (análise estática e dinâmica).	Defasagem temporal, apresentando técnicas que podem ter evoluído desde 2015.

Fonte: Autoria própria

4 PROCESSO METODOLÓGICO

Este capítulo descreve o processo metodológico adotado na elaboração deste trabalho, com o objetivo de responder à pergunta de pesquisa: *Quais são as técnicas de ofuscação do trojan bancário Banbra que dificultam sua detecção por antivírus tradicionais?*

4.1 Caracterização da pesquisa

Esta seção tem como objetivo apresentar a caracterização da pesquisa sob diferentes perspectivas, classificando-a quanto à sua natureza, aos seus objetivos, à sua abordagem e aos procedimentos técnicos adotados. Essa caracterização é essencial para expor, de forma clara e estruturada, o propósito deste trabalho.

4.1.1 Quanto à natureza

Esta pesquisa é classificada como aplicada, pois seu propósito não é gerar conhecimento apenas teórico, mas sim investigar um problema concreto e específico do mundo real, no caso, as táticas de evasão de um *trojan* bancário ativo. O conhecimento gerado a partir da análise do *Banbra* possui aplicação imediata, contribuindo para o desenvolvimento de estratégias de defesa por parte de empresas de cibersegurança.

4.1.2 Quanto aos objetivos

Esta pesquisa é caracterizada como exploratória e descritiva.

- **Exploratória:** A pesquisa realiza uma investigação aprofundada sobre as técnicas de ofuscação de uma família específica de *malware*, o *Banbra*, explorando um tema ainda pouco documentado no meio acadêmico no que diz respeito às suas especificidades.
- **Descritiva:** O trabalho se propõe a descrever e catalogar minuciosamente as características do objeto de estudo. Ele visa detalhar quais são as técnicas de ofuscação utilizadas pelo *Banbra* e como elas funcionam, sem haver a necessidade de um caráter mais explicativo.

4.1.3 Quanto à abordagem do problema

A abordagem adotada é predominantemente Qualitativa, já que o foco da pesquisa não reside em dados quantitativos, como taxas de detecção ou frequência de ataques, mas sim na

compreensão aprofundada de um determinado caso. Essa escolha permite examinar o *Banbra* de forma mais detalhada, considerando seus aspectos comportamentais e estruturais. A análise qualitativa favorece a identificação de nuances e padrões de ofuscação que poderiam passar despercebidos em uma abordagem puramente estatística. Essa análise é feita com base na interpretação de código, na observação de comportamentos e na decodificação de estratégias. O objetivo é entender a qualidade e a natureza das técnicas de ofuscação, em vez realizar a quantificação da sua ocorrência.

4.1.4 Quanto aos procedimentos técnicos

No que diz respeito aos procedimentos técnicos, a pesquisa se baseia em uma combinação de pesquisa bibliográfica, pesquisa documental e estudo de caso.

A pesquisa bibliográfica e documental foi utilizada para construir a fundamentação teórica, por meio da leitura e análise de livros, artigos científicos e relatórios técnicos de empresas especializadas, estabelecendo os conceitos sobre *malwares, trojans* bancários, mecanismos de detecção e ofuscação.

Já o estudo de caso é o método central da pesquisa, conforme afirmam Goode e Hatt (1973, apud Pereira *et al.* (2009)). O estudo de caso caracteriza-se como a investigação aprofundada de um objeto, permitindo amplo e detalhado conhecimento sobre o mesmo, o que seria praticamente impossível por meio de outros métodos de investigação. No contexto deste trabalho, o "caso" é o *trojan* bancário *Banbra*. Esse método possibilita uma análise aprofundada das camadas de ofuscação e evasão de cada entidade, proporcionando uma visão detalhada do objeto de estudo.

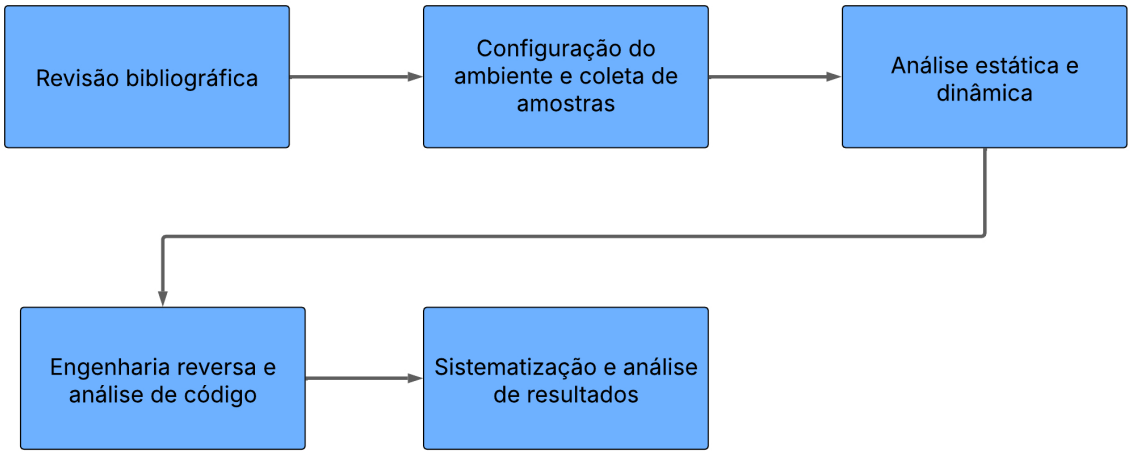
4.2 Fluxo metodológico

O desenvolvimento desta pesquisa seguiu um fluxo de trabalho bem estruturado e dividido em fases sequênciais que fizeram a combinação de uma análise teórica com a investigação do objeto de estudo. Os procedimentos adotados fornecem uma abordagem sistemática para responder à pergunta de pesquisa.

A Figura 4 mostra o fluxo de processos que foram realizados no desenvolvimento da pesquisa. Cada etapa foi estruturada de forma a se apoiar na anterior, assegurando uma investigação coesa, progressiva e aprofundada. Essa estrutura metodológica foi fundamental

para assegurar o rigor científico e a rastreabilidade de todo o processo investigativo.

Figura 4 – Fluxo de processos a serem realizados na pesquisa



Fonte: Autoria própria

O percurso metodológico iniciou-se com uma **revisão bibliográfica**, etapa essencial para estabelecer o referencial teórico e compreender o conhecimento atual e as principais abordagens sobre os *trojans* bancários e técnicas de ofuscação. Esta base de conhecimento fundamenta todas as fases subsequentes. Na sequência, a fase de **configuração do ambiente e coleta de amostras** é executada. Seu objetivo foi preparar um laboratório de análise controlado e isolado, garantindo a segurança durante a manipulação do *malware*, além de obter os artefatos do *Banbra* que foram o foco do estudo.

Com o ambiente devidamente configurado e as amostras (*hashes*) coletadas através das plataforma de detecção de *malware*, deu-se início a **análise estática** e a **análise dinâmica**, estas duas servindo como fase preliminar da investigação. A análise estática examinou as características dos arquivos sem executá-los, enquanto a análise dinâmica observou o comportamento do *malware* em tempo real dentro do ambiente controlado, registrando suas interações com o sistema.

A etapa seguinte consistiu na **engenharia reversa e análise de código**. Nela, o

foco foi desconstruir o funcionamento interno do *malware*, com o objetivo de ultrapassar suas camadas de proteção para examinar seu código e identificar com precisão as técnicas de ofuscação empregadas.

Por fim, o fluxo terminou na **sistematização e análise de resultados**. Nesta fase conclusiva, todos os dados técnicos coletados foram organizados, classificados e interpretados, permitindo transformar as observações em conclusões fundamentadas sobre as estratégias de evasão do *Banbra*.

5 PROCEDIMENTOS DE ANÁLISE E AMBIENTE EXPERIMENTAL

Este capítulo se dedica a detalhar a abordagem prática e o conjunto de procedimentos técnicos empregados para responder à pergunta central deste trabalho. O objetivo é descrever de forma transparente e rigorosa cada etapa da investigação, desde a preparação do ambiente de análise e a obtenção das amostras do *trojan Banbra*, até as fases de análise, engenharia reversa e, por fim, a sistematização dos resultados. Esta descrição é fundamental para garantir a replicabilidade dos experimentos. Ao detalhar o fluxo de trabalho e as ferramentas utilizadas, busca-se ofertar um guia claro sobre como as evidências foram coletadas e interpretadas, fundamentando as conclusões que serão apresentadas posteriormente.

5.1 Revisão bibliográfica

O ponto de partida da investigação prática consistiu em um levantamento da literatura acadêmica e técnica. Foram consultadas fontes diversas, incluindo artigos científicos de bases renomadas como a IEEE Xplore e ScienceDirect, livros especializados em análise de *malware* e engenharia reversa como o Sikorski e Honig (2012) e relatórios técnicos publicados por empresas da indústria de cibersegurança, como os divulgados pela Karspersky e ESET. Esta etapa foi crucial para consolidar o embasamento teórico apresentado no Capítulo 2 e para contextualizar o problema de pesquisa, fornecendo um panorama sobre as características e táticas de *trojans* bancários e mecanismos de ofuscação.

5.2 Configuração do ambiente e obtenção de amostras

A segunda fase da metodologia desta pesquisa, fundamental para a segurança e a validade dos resultados, compreende a configuração de um ambiente de análise controlado e a obtenção criteriosa das amostras do *malware*. Cada um desses processos foi executado com rigor para garantir um ambiente fechado, que impede a propagação do artefato malicioso, e para assegurar a autenticidade e a relevância das amostras coletadas para o estudo.

A preparação do ambiente para a análise segura e isolada do *malware* foi a primeira e mais crítica etapa. Para isso, será estabelecido um laboratório de análise, popularmente conhecido como *sandbox*, utilizando um computador dedicado pertencente à infraestrutura do laboratório Mandacaru.dev, pertencente ao campus da Universidade Federal do Ceará – Crateús[1].

[1] https://site.crateus.ufc.br/

A decisão de usar uma máquina dedicada foi estratégica, garantindo que os recursos de *hardware*, como processador, memória RAM e armazenamento, sejam exclusivamente alocados para a análise, evitando assim qualquer compartilhamento com outros sistemas ou projetos. Isso elimina o risco de degradação de performance por concorrência de recursos e, mais importante, cria uma primeira camada de isolamento. Como medida de segurança primária e mais robusta, o computador foi fisicamente desconectado de todas as redes, incluindo a internet e a rede local da universidade. Essa prática cria um espaço vazio de comunicação, tornando-se a forma mais eficaz de prevenir qualquer risco de propagação acidental do *malware*, protegendo a integridade da infraestrutura de rede da instituição.

Com o ambiente de análise devidamente isolado e seguro, o passo seguinte foi a coleta das amostras do *trojan Banbra*. Essa coleta foi realizada a partir de repositórios públicos de *malware*, que funcionam como grandes bibliotecas colaborativas para a comunidade de cibersegurança. As plataformas escolhidas foram o VirusTotal[2] e o MalwareBazaar[3], selecionadas por sua confiabilidade, vasto acervo e pela riqueza dos metadados que fornecem.

O **VirusTotal** é um serviço do Google e uma das fontes mais confiáveis e abrangentes para a obtenção de amostras. Sua confiabilidade está em uma tecnologia de detecção, mas na agregação dos resultados de mais de 70 motores de antivírus diferentes. Isso fornece um consenso da indústria sobre a natureza de um arquivo. Para obter amostras do *Banbra*, foi necessário criar uma conta de pesquisador, além de entrar em contato com os gerenciadores da plataforma, onde foi disponibilizado uma pasta contendo uma infinidade de amostras. A busca pelas amostras foi realizada utilizando consultas específicas, como hashes SHA-256 conhecidos de variantes anteriores do *Banbra*, ou pesquisando por nomes de detecção comuns atribuídos a ele pelas empresas de segurança (como por exemplo, *Trojan.Win32.Banbra*). Como plataforma permitiu o *download* direto dessas amostras para o ambiente de análise, ela foi uma fonte primária indispensável pelo seu volume e histórico de dados.

Além dessa fonte, foi utilizado o **MalwareBazaar**, ele é um projeto da iniciativa suíça sem fins lucrativos. A escolha desta plataforma se deu pelo seu foco em compartilhar amostras de *malware* recentes e em atividade. Sua confiabilidade é alta na comunidade de inteligência de ameaças, pois os pesquisadores que contribuem frequentemente anexam informações contextuais valiosas, como **tags** (ex: #banbra) e assinaturas de detecção. Para obter as amostras, o processo envolveu o registro de uma conta e a utilização do ferramenta de busca para filtrar por **tags**,

[2] https://www.virustotal.com/gui/home
[3] https://bazaar.abuse.ch/

famílias de malware ou outros indicadores. A capacidade de encontrar variantes recentes e com contexto associado tornou o MalwareBazaar uma fonte estratégica para esta pesquisa.

Para cada artefato coletado nas plataformas, foi adotado um procedimento de catalogação. O hash SHA-256 de cada arquivo foi devidamente registrado. Este passo garante integridade e rastreabilidade. A integridade assegura que o arquivo analisado seja exatamente o mesmo que foi baixado, sem corrupção ou modificação acidental durante a transferência. Já a rastreabilidade cria um vínculo direto entre os resultados da análise e a amostra específica, permitindo que a pesquisa seja transparente, verificável e passível de ser reproduzida por outros pesquisadores.

5.3 Análise estática e dinâmica

Após a preparação do ambiente e a coleta segura das amostras, a investigação seguiu para a fase de análise, que foi dividida em duas abordagens complementares e em sequência: a análise estática e a análise dinâmica. Essas duas técnicas permitem a construção de um perfil detalhado do *malware* analisado.

A análise estática foi a etapa inicial de investigação, efetuada de forma segura e sem a execução do código malicioso. O objetivo foi identificar as características básicas do *Banbra*, como o formato do arquivo, para verificar suas funções importadas e descobrir suas capacidades. Também foram buscadas *strings* de texto embutidas, como nomes de arquivos, que possam dar pistas sobre sua funcionalidade. Um dos focos principais desta fase foi a identificação do uso de *packers*, ferramentas que ocultam o código original e onde sua detecção é um indicativo importante da necessidade de uma análise mais profunda.

Logo após esta fase, a análise dinâmica foi conduzida executando o *trojan* de forma controlada. O propósito foi monitorar e registrar suas interações reais com o sistema. O monitoramento teve como foco o registro de modificações no sistema de arquivos e no registro do Windows, especialmente em chaves de inicialização para detectar mecanismos de persistência. Além disso, foram capturadas as tentativas de comunicação com a rede, mesmo com a máquina offline, para identificar os endereços dos servidores visados pelo trojan.

5.4 Engenharia reversa e análise de código

Esta fase foi uma parte central da investigação, constituindo o núcleo da coleta de dados para esta pesquisa. Enquanto as análises estática e dinâmica revelam as características superficiais e o comportamento observável do *trojan*, a engenharia reversa permite abordar sua lógica interna para compreender profundamente como ele opera e como se protege.

O primeiro passo que foi superado, e que se esperava identificar durante a análise estática, foi a camada de proteção criada pelos *packers*. Sendo assim, o primeiro passo desta etapa foi o processo de desempacotamento do *malware*. O objetivo foi contornar essa proteção para revelar o código original. Isso foi alcançado executando a amostra em um ambiente controlado por um depurador. A ferramenta permitiu pausar a execução em momentos críticos, inspecionar a memória e, no momento exato em que o código original for reconstruído na memória, extraí-lo para um novo arquivo executável, agora sem essa camada de proteção.

Com a extração bem sucedida do *payload*, o código-fonte original se tornou finalmente acessível. Ao ser carregado em ferramentas como o descompilador Ghidra, o código desofuscado se apresenta em um estado que permite a investigação de técnicas de ofuscação mais sutis, que estavam anteriormente ocultas pela camada do *packer*. A ofuscação, afinal, não se restringe apenas ao empacotamento, mas refere-se a qualquer técnica usada para tornar o código mais difícil de ser compreendido.

A partir deste ponto, a análise à ser utilizada é a busca por uma série de táticas utilizadas comumente por *trojans*, que são a **criptografia de *strings***, onde textos importantes como URLs de servidores, nomes de arquivos ou comandos são armazenados de forma cifrada em binário e decifrados apenas em tempo de execução para evitar a detecção estática, a *ofuscação de chamadas de API*, que são técnicas que evitam chamar funções do sistema operacional diretamente pelo nome, utilizando métodos como o cálculo dinâmico de seus endereços de memória de forma a esconder as reais capacidades do *malware* e por fim as **técnicas anti-análise** que são blocos de código projetados para detectar se o *trojan* está sendo executado em um ambiente de análise (como uma máquina virtual ou sob um depurador) e, caso seja verdade, ele alterar seu comportamento ou encerrar sua execução de forma a frustrar o analista.

5.5 Sistematização e análise de resultados

Esta etapa final teve como objetivo transformar os dados brutos obtidos na fases anteriores, em conhecimento estruturado e significativo. Além de apenas identificar as técnicas empregadas, se buscará compreender sua finalidade e avaliar sua eficácia diante de mecanismos de detecção. Para isso, o processo foi conduzido em duas formas: a catalogação sistemática das técnicas observadas e a análise crítica de seu impacto e relevância no contexto da evasão de antivírus.

As técnicas de ofuscação identificadas no código do *trojan Banbra* foram organizadas e catalogadas de forma sistemática. Para cada técnica encontrada, foi criada uma entrada detalhada que incluiu uma definição conceitual, uma descrição de sua implementação específica na amostra analisada e, sempre que possível, evidências concretas, como trechos do código desmontado ou do pseudo-código gerado. Este catálogo não funciona apenas como um inventário, mas como uma base de conhecimento detalhada sobre o modo de operação do *malware*.

Após a catalogação, as técnicas foram submetidas a uma avaliação crítica. O objetivo desta análise foi entender o valor estratégico de cada método de ofuscação. A avaliação se concentrou em relacionar diretamente cada técnica com os mecanismos de detecção que ela visa contornar.

O resultado desta etapa foi a construção de um perfil abrangente das estratégias de evasão do *Banbra*, mapeando não apenas o "o quê", mas principalmente o "porquê" de suas escolhas de ofuscação, fornecendo assim um conhecimento sobre ameaças que é importante no desenvolvimento de contramedidas melhores.

5.6 Ferramentas e ambiente de análise

A execução bem-sucedida das fases de análise descritas neste trabalho dependeu da seleção e configuração de um conjunto de *hardware, software* e ferramentas técnicas. Cada ferramenta foi selecionada por sua capacidade de atender a uma necessidade específica dentro do fluxo de trabalho da análise de *malware*. O que se detalha a seguir é o ambiente de laboratório que foi empregado, formando a base metodológica para a análise do *trojan Banbra*.

A base para este ambiente de análise foi um computador dedicado, fornecido pelo laboratório Mandacaru.dev, com um processador Intel Core i5, 8 GB de RAM e 512 GB de HD. Esta configuração de *hardware* foi escolhida para garantir a capacidade de processamento

necessária para a execução simultânea das múltiplas ferramentas de análise intensiva, sem comprometer o desempenho do sistema.

Uma vez que o ambiente foi preparado, uma série de ferramentas de análise especializadas foram empregadas. A investigação começou com a análise estática, na qual a ferramenta PE-bear foi essencial. Ela foi utilizada para examinar a estrutura do arquivo *Portable Executable* (PE), permitindo uma inspeção detalhada de seus cabeçalhos e seções. O motivo de seu uso foi a capacidade de analisar rapidamente a tabela de importação de endereços, que revela quais funções do sistema operacional o *malware* pretendia usar, oferecendo as primeiras pistas sobre suas intenções sem a necessidade de executá-lo. De forma complementar, o editor hexadecimal HxD foi empregado para visualizar o conteúdo bruto do arquivo, byte a byte. Ele foi escolhido por permitir a identificação de padrões, *strings* ofuscadas ou dados embutidos que não são visíveis em analisadores estruturais, fornecendo uma visão fundamental do conteúdo real do artefato.

Para a análise dinâmica, o Wireshark foi indispensável. Sua função foi capturar e analisar todo o tráfego de rede. Ele será utilizado para registrar todas as tentativas de comunicação que o *malware* fez. A justificativa para seu uso é que, mesmo em um ambiente isolado, ele revela os endereços *Internet Protocol* (IP), domínios dos servidores de comando e controle, os protocolos e os dados que o *Banbra* tenta enviar ou receber. Essas informações são vitais para mapear sua infraestrutura. Para entender o comportamento do *malware* no sistema, o *Process Monitor* (ProcMon) foi a principal ferramenta. Ele foi escolhido por sua capacidade de criar um *log* exaustivo e em tempo real de todas as interações do *malware* com o sistema de arquivos e o registro do Windows, respondendo à pergunta sobre o que o *malware* fez no sistema. Para refinar essa análise, o Regshot foi utilizado. Sua função foi tirar um “antes” e um “depois” do estado do registro, gerando um relatório conciso que destaca apenas as alterações. A razão de seu uso foi a eficiência, pois ele permite identificar rapidamente mecanismos de persistência sem a necessidade de filtrar os milhares de eventos gerados pelo ProcMon.

Por fim, na fase de engenharia reversa, o depurador *x64dbg* foi a ferramenta utilizada. Sua função foipermitir a execução controlada do *malware*, passo a passo, e inspecionar a memória e os registradores da CPU. Ele foi escolhido por ser crucial para o processo de desempacotamento, com ele, é possível pausar a execução no momento exato em que o código original for descompactado na memória, permitindo sua extração para análise. Uma vez desempacotado, este código foi carregado no *framework* Ghidra. A razão para utilizar o Ghidra é seu poderoso

descompilador, que traduz o complexo código em baixo nível em um código C de alto nível. Isso é fundamental porque torna a análise da lógica do *malware*, a identificação de algoritmos e a documentação das técnicas de ofuscação um processo significativamente mais rápido, claro e inteligível.

6 RESULTADOS

Este capítulo apresenta os resultados detalhados obtidos a partir da aplicação da metodologia de análise de *malware* descrita anteriormente. O objetivo é analisar, camada por camada, as técnicas de ofuscação e os comportamentos da amostra selecionada do *trojan* bancário *Banbra*. A análise será apresentada de forma sequencial, iniciando por a obtenção e validação da amostra, seguindo após para uma análise estática para examinar a estrutura do arquivo e identificar ofuscadores e, por fim a descrição das características observadas através da análise dinâmica, ou seja, com a execução do *malware*.

6.1 Obtenção e seleção da amostra

Para este estudo de caso, a amostra do *trojan* bancário *Banbra* foi obtida através da plataforma de inteligência de ameaças VirusTotal. A plataforma foi escolhida por ser um dos maiores e mais conceituados repositórios de *malware* do mundo, agregando dados de vários motores de antivírus e ferramentas de análise, o que garante um alto grau de confiança na identificação inicial da ameaça.

A amostra foi disponibilizada como um arquivo binário, cujo nome correspondia ao seu hash SHA-256 (fb66797227f6f36eebc77e0ea7ebc95ae3ca614985c4b329bdff5724f510a400), seguido por um arquivo de metadados estruturado em formato JSON. Este arquivo continha um relatório detalhado de todas as análises submetidas à plataforma, incluindo informações cruciais como o histórico de submissão, assinaturas digitais, e os resultados de detecção de mais de 60 diferentes soluções de segurança. O critério que ajudou na decisão da seleção desta amostra específica foi a classificação explícita por um dos motores de detecção, o TACHYON, que a identificou como "Banker/W32.Banbra.Gen". Esta tipificação, aliada a outras detecções genéricas de *trojan* bancário e *worm*, forneceu a validação necessária para prosseguir com uma análise aprofundada.

6.2 Análise preliminar e interação com o Sistema de Segurança

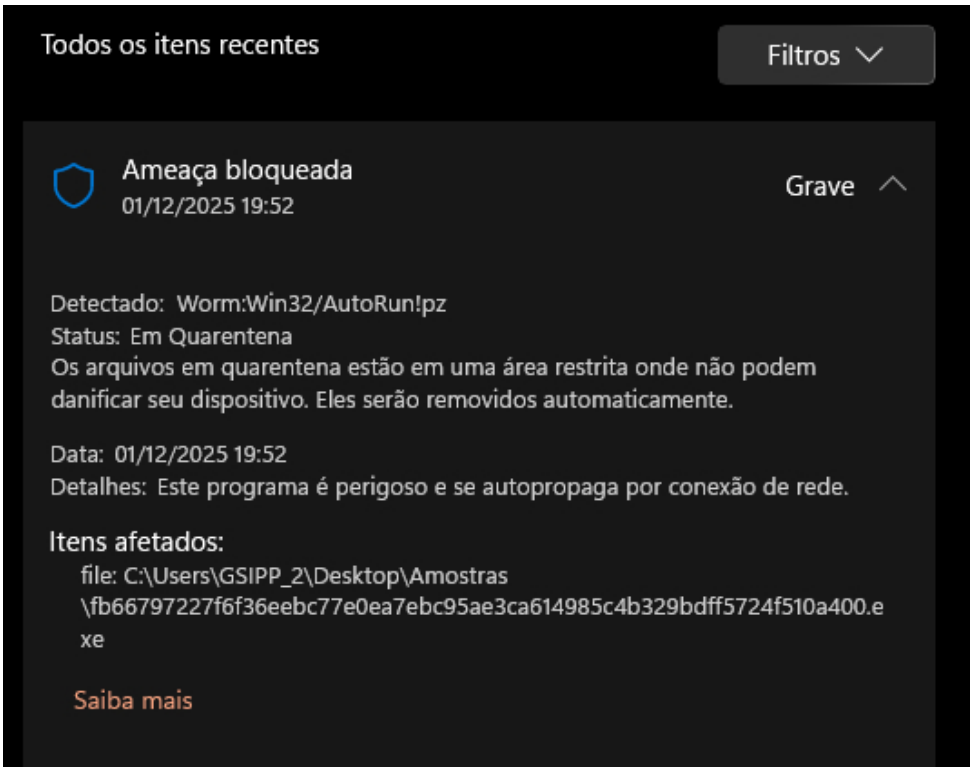
A primeira interação direta com a amostra mostrou imediatamente um comportamento malicioso ativo, na qual validou as classificações observadas no VirusTotal. Após a conversão do hash para um arquivo binário e a atribuição da extensão *.exe*, o sistema de segurança nativo do Windows, o Microsoft Defender, entrou em ação de forma imediata e automática. O

arquivo executável era repetidamente removido da sua pasta de origem e colocado em quarentena. Ao inspecionar o histórico de proteção do programna, a ameaça foi consistentemente classificada como *"Worm:Win32/AutoRun!pz"*.

Esta detecção demonstrou que os mecanismos de detecção nativos dos Sistemas Operacionais já possuem um bom desempenho na identificação de ameaças nocivas ao sistema, no entanto impossibilitam a realização de experimentos sem a alteração dos seus procedimentos de capturas de ameaças. Para viabilizar os procedimentos subsequentes de análise estática e dinâmica, foi necessário configurar um ambiente de laboratório controlado, cuja primeira etapa consistiu em criar uma regra de exceção no Microsoft Defender, instruindo-o a ignorar todas as atividades na pasta designada para a análise do *trojan* bancário. Esta medida, embora perigosa em um sistema normal, é indispensável em ambientes de pesquisa para permitir a manipulação e execução segura do artefato malicioso.

A Figura 5 demonstra de forma detalhada as informações exibidas pelo Microsoft Defender após o bloqueio da amostra do *malware* quando convertido para um arquivo executável.

Figura 5 – Aviso do Microsoft Defender após a captura do *malware*



Fonte: Autoria própria

6.3 Análise estática inicial

Com o ambiente de análise preparado com todas as ferramentas necessárias, a primeira fase da investigação consistiu em uma análise estática, onde o objetivo foi inspecionar a estrutura e o conteúdo do arquivo executável sem realizar sua execução. Nesta etapa foram utilizadas três ferramentas distintas: PE-bear para a análise da estrutura do arquivo PE, HxD para a inspeção do conteúdo hexadecimal e extração de *strings*, e o Ghidra para a tentativa inicial de desmontagem e descompilação.

A análise feita com o PE-bear revelou informações importantes sobre a construção do *malware*. Ao examinar os cabeçalhos das seções *(Section Headers)*, foi identificada uma seção com o nome incomum de **.aspack**. Este nome é um forte indicador da utilização do *packer* ASPack, que é uma ferramenta de software cujo propósito é comprimir e criptografar o código executável original para dificultar a análise e evadir a detecção por antivírus baseados em assinaturas. Embora a versão da ferramenta utilizada não exibisse o cálculo numérico de entropia, que é o cálculo do nível de ofuscação do código, a presença desta seção já gerou uma evidência primária de ofuscação.

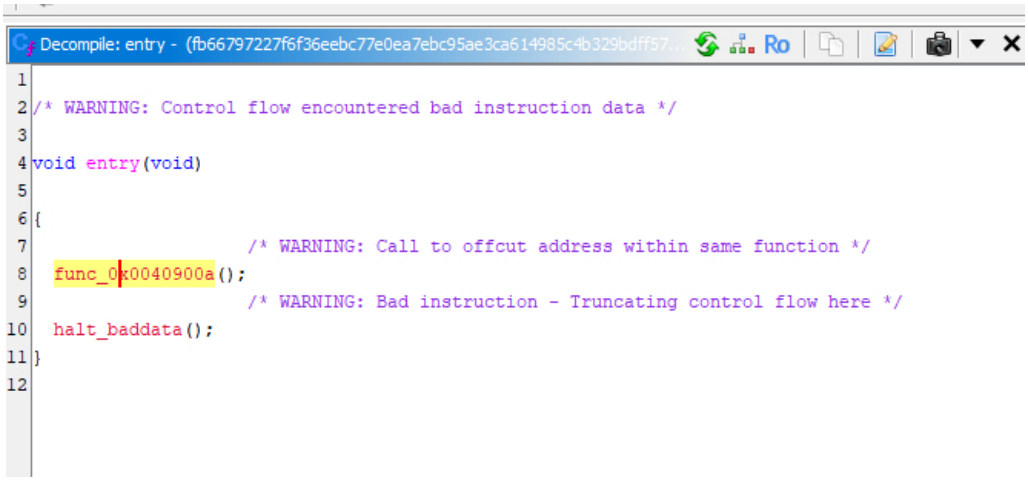
Após isso, foi realizada uma inspeção do binário com o editor hexadecimal HxD, que permitiu a extração de *strings* de texto, mas somente as legíveis que sobreviveram ao processo de empacotamento. Esta análise revelou uma série de artefatos significativos. O primeiro foi a confirmação da dependência da biblioteca **msvbvm60.dll**, indicando que o *payload* foi desenvolvido na linguagem Visual Basic 6. O outro ponto identificado foi a presença de metadados falsificados, onde o *malware* tenta se passar por o programa "Microsoft Firewall", apesar de o mesmo listar a "Xiang Corporation"como a sua empresa de origem, havendo uma grande contradição. Além disso, houve uma descoberta bastante relevante que foi um grande bloco de strings relacionadas à ferramenta **MSInfo32.exe**, sugerindo que o*malware* possui uma capacidade avançada de reconhecimento do sistema para coletar informações detalhadas de *hardware* e **software** da máquina da vítima, podendo ser usado para verificar a sua execução em uma VM ou no roubo de informações do sistema.

Houve ainda a detecção de mais outras duas anormalidades, que foram a presença de *URL*s legitimas relacionadas a validação de certificados digitais como *http://ocsp.thawte.com* e *http://ts-ocsp.ws.symantec.com*, juntamente com a *URLDownloadToFileW urlmon.dll* que é uma função do Windows que permite baixar um arquivo da internet e salvá-lo no disco, provando que o *trojan* pode atuar como um *downloader*.Por fim, foi descoberto *strings* de um programa escrito

em .NET e não somente em Visual Basic 6, mostrando que a amostra se trata de um *malware* híbrido ou de múltiplos estágios, onde a função do primeiro estágio é desempacotar e executar o segundo estágio, que é o *trojan* real.

Para finalizar a análise estática, a amostra foi importada para a ferramenta de engenharia reversa Ghidra. A A tentativa de análise automática e descompilação do ponto de entrada (entry) do programa falhou, resultando em um código fragmentado e ilegível, com múltiplos avisos de *"bad instruction data"* e *"control flow truncation"*, igual é representado pela Figura 6. Este resultado reforçou as descobertas anteriores, confirmando que o código executável estava de fato ofuscado. Os resultados da análise estática inicial realizados por todas as três ferramentas chegaram a conclusão de que a amostra era um *malware* de múltiplos estágios, protegido por um *packer* e que exigiria uma análise dinâmica e depuração para que seu núcleo pudesse ser revelado e estudado.

Figura 6 – Código da função *entry* inicialmente gerado pelo Ghidra



Fonte: Autoria própria

6.4 Análise dinâmica

Diante da impossibilidade de analisar o código ofuscado por completo de forma estática, a pesquisa avançou para a análise dinâmica, onde o objetivo desta fase foi executar

a amostra em um ambiente de laboratório isolado e monitorar suas interações com o sistema operacional em tempo real, utilizando as ferramentas Regshot, Wireshark e ProcMon de forma simultânea. Os resultados mostraram um comportamento agressivo, com táticas de anti-análise e um impacto severo na estabilidade do sistema.

Ao realizar a primeira tentativa de monitoramento, houve uma exposição imediata das capacidades de autodefesa do *malware*. Poucos instantes após a execução da amostra, o processo da ferramenta ProcMon foi finalizado de forma abrupta e sem intervenção do usuário. Este comportamento é característico de *malwares* que buscam ativamente por ferramentas de análise (como ProcMon) para encerrá-las, impedindo assim a observação de suas atividades. Como tentativa de contornar essa tática, foi necessário renomear o executável do ProcMon para um nome genérico antes de uma nova tentativa, de forma que o *malware* não identificasse a ferramenta.

Durante a segunda execução, com as alterações já feitas, a amostra demonstrou ser extremamente nociva ao desempenho do sistema. Após um curto período de atividade intensa, o ProcMon travou, e o Sistema Operacional como um todo se tornou instável de forma que ele congelou, exigindo um desligamento forçado. Esta instabilidade foi atribuída a uma sobrecarga de operações de I/O (leitura/escrita) realizadas pelo *malware* em um curto espaço de tempo. O log do ProcMon, salvo antes do travamento, mostrou a causa dessa atividade, onde o trojan realizou uma varredura exaustiva no sistema de arquivos, enumerando dezenas de executáveis de *softwares* de fabricantes (como HP) e componentes do Microsoft Office e do Internet Explorer.

A ferramenta Wireshark, apesar de ter sido testada antes das análises, não identificou nenhuma tentativa de comunicação com a rede para nenhum programa, não só para a amostra, o que acarretou na desconsideração dessa ferramenta nos estudos das interações da amostra. Já a análise do Regshot não revelou a criação de chaves de persistência convencionais (como nas chaves Run do Registro), mesmo após uma execução bem-sucedida. Esta ausência, combinada com os dados do ProcMon, fortaleceu a hipótese de que a amostra operava em múltiplos estágios. A natureza agressiva e as técnicas de anti-análise do *malware*, no entanto, tornaram a coleta de um log completo de seu ciclo de vida inviável, direcionando a pesquisa para a necessidade de uma análise controlada via depuração para desofuscar e extrair seu núcleo de forma definitiva.

Tendo em vista que tanto as análise estáticas quanto as dinâmicas apontavam para um *payload* ofuscado e de múltiplos estágios, a etapa final da pesquisa se concentrou em utilizar um depurador para executar o *malware* de forma controlada, contornando as proteções do *packer*

e extraindo o código malicioso principal. Para esta tarefa, foi utilizada a ferramenta x32dbg, um depurador de 32 bits.

O arquivo da amostra original foi carregado no x32dbg, permitindo a observação do código assembly em tempo de execução. Conforme esperado, o ponto de entrada inicial continha o código do *packer* ASPack. Utilizando técnicas de depuração exaustivas, como a execução passo a passo e a definição de *breakpoints* de memória na seção **.aspack**, foi possível acompanhar a rotina de descompressão. Com isso, pode-se observar que o processo revelou um comportamento de múltiplos estágios: após a descompactação inicial, a execução não era transferida para o código final do *Banbra*, mas sim para um segundo *"loader"* intermediário. Este segundo estágio foi então depurado de forma sequêncial, fazendo uma interação por vez, até que chegou um momento onde hove um salto na mémória, onde foi transferido o fluxo de execução para uma nova região de memória, revelando o Ponto de Entrada Original (OEP) do *payload* definitivo. O código no OEP iniciava com a instrução PUSHAD, um indicativo clássico do verdadeiro início de um programa executável.

Com a execução pausada exatamente no OEP, foi realizado uma extração da memória no processo do malware utilizando o *plugin* Scylla. Este procedimento acarretou na extração do código completamente desempacotado e reconstruiu sua Tabela de Endereços de Importação (IAT), gerando um novo arquivo executável funcional. Este arquivo, representava o núcleo do trojan *Banbra*, já que não se encontrava mais ofuscado.

De forma a validar o que foi feito, o arquivo extraído foi importado para o Ghidra, onde foi realizado mais uma análise automática. Apesar de ter ocorrido ainda alguns erros, onde o Ghidra não conseguiu ler o arquivo de forma limpa por completo, a seção *Entry* agora revelou uma extensa linha de códigos contendo várias funções onde sua maioria eram nomeadas de forma genérica, pois o Ghidra não conseguiu identifica-las por completo, mas que mostraram que o código já não estava mais ofuscado.

Com a extração e a descompilação bem-sucedida do núcleo do *malware*, a pesquisa atingiu seu objetivo principal de análise das principais técnicas de ofuscação, podendo haver por subsequência o aprofundamento na lógica de negócio especifica contida neste código final.

6.5 Técnicas de evasão identificadas

A análise híbrida (estática e dinâmica) permitiu a identificação de um conjunto diversificado de técnicas de ofuscação e evasão empregadas pela amostra do *trojan Banbra*.

Por mais que essas técnicas tenham sido observadas em diferentes etapas da investigação, elas operam de forma coordenada para atingir um objetivo comum: dificultar a detecção, frustrar a análise e garantir a execução do *payload* malicioso. Esta seção consolida e categoriza as principais táticas identificadas.

As técnicas observadas foram agrupadas em cinco categorias principais:

1. **Empacotamento *(Packing)*:** A camada mais externa de proteção. O código original é comprimido e/ou cifrado, e encapsulado dentro de um *stub* de descompressão.
2. **Arquitetura de Múltiplos Estágios:** Uma estratégia que segmenta o ataque. O vetor de infecção inicial (*dropper*) é diferente do *payload* final, dificultando a correlação das diferentes fases da infecção.
3. **Técnicas de Anti-Análise (Autodefesa):** Mecanismos projetados para detectar e reagir à presença de um ambiente de análise ou de ferramentas de engenharia reversa.
4. **Reconhecimento de Ambiente (Fingerprinting):** A capacidade do *malware* de coletar informações detalhadas sobre o sistema hospedeiro para identificar se está em uma VM ou para adaptar seu ataque.
5. **Ofuscação de Metadados e Fluxo:** Táticas para enganar tanto o usuário quanto ferramentas automatizadas, seja através de informações de arquivo falsas ou pela inserção de código que quebra descompiladores.

A Tabela 2 resume cada técnica específica encontrada, a evidência técnica que comprova seu uso e o objetivo principal por trás de sua implementação.

A combinação de forma coordenada destas técnicas demonstra acomo a amostra analisada é sofisticada. O *malware* não depende de um único método de proteção, mas sim de uma defesa de forma profunda, onde cada camada de ofuscação e evasão serve para proteger a seguinte, tornando a análise completa um desafio técnico significativo.

Tabela 2 – Resumo das Técnicas de Evasão Identificadas na Amostra do Banbra

Técnica Específica	Evidência Técnica	Objetivo da Evasão
Empacotamento com ASPack	Presença da seção `.aspack` no cabeçalho PE (identificada com PE-bear) e alta entropia visual (inspeção com HxD).	Comprimir e cifrar o código para evadir a detecção baseada em assinaturas e dificultar a análise estática.
Loader Intermediário	Necessidade de dois ciclos de depuração no x32dbg para alcançar o OEP final, que era diferente do ponto de entrada do código desempacotado pelo ASPack.	Aumentar a complexidade da análise, forçando o analista a derrotar múltiplas camadas de desofuscação.
Capacidade de Downloader	Presença da *string* `URLDownloadToFileW` e referência à `urlmon.dll` no binário (análise com HxD).	Permitir que o malware baixe estágios subsequentes ou atualizações da internet, distanciando ainda mais o vetor inicial do payload.
Encerramento de Ferramentas de Análise	Finalização abrupta e automática do processo `ProcMon.exe` durante a análise dinâmica.	Impedir o monitoramento de suas interações com o sistema operacional, frustrando a análise de comportamento.
Coleta de Informações do Sistema	Presença de um grande bloco de *strings* relacionadas à ferramenta `MSInfo32.exe` no binário (análise com HxD).	Identificar se está sendo executado em uma máquina real ou em uma VM de análise e coletar dados sobre a vítima.
Uso de Metadados Falsos	O arquivo se identificava como "Microsoft Firewall", mas a empresa listada era "Xiang Corporation"(análise com HxD).	Enganar o usuário e sistemas de reputação, fazendo o arquivo parecer um componente legítimo do sistema operacional.
Ofuscação de Fluxo de Controle	A análise inicial no Ghidra resultou em erros de *"bad instruction data"* e fluxo de controle truncado, impedindo a descompilação.	Quebrar ferramentas de análise automática (desmontadores e descompiladores), escondendo a lógica real do programa.

Fonte: Autoria própria

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho obteve sucesso nos seus objetivos primários de identificar e categorizar as técnicas de ofuscação e evasão empregadas por uma amostra do *trojan* bancário *Banbra*. As táticas identificadas, na qual vão desde o empacotamento externo até mecanismos de autodefesa em tempo de execução, foram devidamente descritas no capítulo de resultados, cumprindo o objetivo de fornecer uma análise estruturada dos métodos defensivos do *malware*.

A investigação identificou que a amostra analisada é um sistema de *software* complexo, que refuta a noção de que *trojans* bancários são artefatos tecnologicamente simples. A arquitetura de múltiplos estágios, a capacidade de reconhecimento de ambiente (*fingerprinting*) e as rotinas de anti-análise ativas demonstram um design extensamente defensivo e na qual foi projetado não apenas para infectar, mas para resistir, detectar e frustrar as tentativas de análise. A forma sofisticada observada evidencia a profissionalização do desenvolvimento de ameaças no cenário brasileiro, tratando-se de um adversário que investe recursos bem significativos para proteger suas operações.

O sucesso desta pesquisa ressalta a importância crítica de uma metodologia de análise híbrida e multifacetada. A utilização de uma única ferramenta ou abordagem se mostrou consistentemente insuficiente. Enquanto a análise estática inicial identificou a presença do *packer*, foi incapaz de se aprofundar na ofuscação de fluxo de controle. Por sua vez, a análise dinâmica de comportamento revelou as táticas de autodefesa, mas falhou em capturar um ciclo de vida completo devido à instabilidade induzida pelo *malware*. Foi somente através da correlação dos artefatos coletados por cada ferramenta e da aplicação da depuração controlada com o x32dbg que a cadeia de ataque pôde ser reconstruída e o *payload* final, extraído. Este estudo, portanto, valida de forma prática que a compreensão de ameaças modernas depende fundamentalmente da capacidade de se integrar e se interpretar os dados de múltiplas fontes de análise.

A desofuscação bem-sucedida do *payload* e a categorização das táticas de evasão, que constituem os resultados centrais deste trabalho, abrem um leque de possibilidades para a continuidade e aprofundamento da pesquisa. As propostas para trabalhos futuros expandem-se a partir das descobertas aqui apresentadas, abrangendo desde a análise aprofundada do artefato extraído até o desenvolvimento de ferramentas e a disseminação do conhecimento para a comunidade de segurança.

Primeiramente, a análise do *payload* desofuscado pode ser significativamente aprofundada. Uma investigação minuciosa de seu código no Ghidra permitiria o mapeamento deta-

lhado de suas rotinas de roubo de dados voltadas a bancos específicos, bem como a decodificação de seu protocolo de comunicação com servidores de Comando e Controle. Adicionalmente, uma análise comparativa com outras famílias de *malwares* brasileiros, como Mekotio ou Grandoreiro, poderia identificar semelhanças no código ou nas técnicas, contribuindo para o entendimento do ecossistema de desenvolvimento de ameaças no Brasil.

Em uma segunda vertente, propõe-se a aplicação prática deste conhecimento no desenvolvimento de contramedidas. O processo manual de identificação das táticas poderia ser automatizado através da criação de *scripts* ou de uma ferramenta de triagem capaz de detectar a presença de *packers*, múltiplos estágios e outras técnicas aqui catalogadas. Com base nisso, seria possível mapear soluções para cada técnica, desenvolvendo assinaturas específicas para o *packer* e para o *loader*, e regras heurísticas para identificar o comportamento de reconhecimento de ambiente.

Finalmente, a disseminação dos resultados é fundamental. Sugere-se a criação de uma plataforma web que sirva como um catálogo dinâmico das técnicas de ofuscação utilizadas por *trojans* bancários, tornando este conhecimento acessível a outros pesquisadores e profissionais. Além disso, os desafios práticos enfrentados durante a análise dinâmica, como a finalização de ferramentas e a instabilidade do sistema, justificam a elaboração de um relato de experiência técnica. Tal documento detalharia os obstáculos e as soluções encontradas, servindo como um guia metodológico valioso para futuros analistas de *malware*.

REFERÊNCIAS

AO Kaspersky Lab. **Trojan-Banker.Win32.Banbra**. 2016. Página de ameaça sobre Trojan-Banker.Win32.Banbra. Disponível em: https://threats.kaspersky.com/br/threat/Trojan-Banker.Win32.Banbra/.

APOSTOLOPOULOS, T.; KATOS, V.; CHOO, K.; PATSAKIS, C. Resurrecting anti-virtualization and anti-debugging: Unhooking your hooks. **Future Gener. Comput. Syst.**, v. 116, p. 393–405, 2021.

BLACK, P.; GONDAL, I.; LAYTON, R. A survey of similarities in banking malware behaviours. **Comput. Secur.**, v. 77, p. 756–772, 2017.

Federação Brasileira de Bancos. **Transações por canais digitais representam 8 em cada 10 operações bancárias no Brasil**. 2024. Acesso em: 23 jul. 2025. Disponível em: https://portal.febraban.org.br/noticia/4310/pt-br/.

GHALEB, M. A. e T. A. Revisão de técnicas baseadas em assinaturas em produtos antivírus. **Conferência Internacional de Ciências da Computação e da Informação de 2019 (ICCIS)**, p. 1–6, 2019.

InfoWester. **Phishing: o que é, como funciona e como se prevenir**. 2019. https://www.infowester.com/phishing.php. Acesso em: 11 jul. 2025.

Kaspersky Lab. **Imortalidade Digital: como o trojan bancário brasileiro evita ser excluído da internet**. 2023. Comunicado de imprensa sobre o trojan bancário brasileiro Guildma. Disponível em: https://www.kaspersky.com.br/about/press-releases/imortalidade-digital-como-o-trojan-bancario-brasileiro-evita-ser-excluido-da-internet.

KUMARI, A.; GUPTA, D.; UPPAL, M. Defending the treasury: Deep learning strategies for banking security against trojan threats. **2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)**, p. 157–162, 2024.

Laboratório de Pesquisa da ESET América Latina. **É possível se infectar com arquivos que não são executáveis?** 2009. Acesso em: 23 jul. 2025. Disponível em: https://www.welivesecurity.com/la-es/2009/02/20/infeccion-archivos-no-ejecutables/.

Microsoft. **Trojan:Win32/Banbra - Threat Encyclopedia**. 2024. Acesso em: 23 jul. 2025. Disponível em: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Trojan:Win32/Banbra.

Ministério Público do Estado de Mato Grosso. **Trojan bancário registra aumento de 87% de ataques no Brasil**. 2024. Acesso em: 24 jul. 2025. Disponível em: https://www.mpmt.mp.br/conteudo/1217/149820/trojan-bancario-registra-aumento-de-87-de-ataques-no-brasil.

MISHRA, M. S. e Piyush Pandey e Aditya Pandey e S. A. Analisando tÉcnicas de evasÃo de malware: Um estudo abrangente de estratÉgias de detecÇÃo em softwares antivÍrus. **International Journal of Engineering Applied Sciences and Technology**, 2025.

NELSON, T.; NANCE, C.; NOTEBOOM, C. Web injection and banking trojan malware -a systematic literature review. **2023 6th International Conference on Information and Computer Technologies (ICICT)**, p. 45–53, 2023.

NIST. **Malware**. 2011. Glossário citando NIST SP 800-137; SP 800-28 Rev2; SP 800-45 Rev2. Disponível em: https://csrc.nist.gov/glossary/term/malware.

Nubank. **66% dos brasileiros usam apps para realizar pagamentos e consultas de saldos, aponta pesquisa Ipsos**. 2024. Notícia no site institucional. Disponível em: https://international.nubank.com.br/pt-br/companhia/summit-de-impacto-financeiro/.

O'KANE, P.; SEZER, S.; MCLAUGHLIN, K. Obfuscation: The hidden malware. **IEEE Security Privacy**, v. 9, p. 41–47, 2011.

PEREIRA, L. d. T. K.; GODOY, D. M. A.; TERçARIOL, D. Estudo de caso como procedimento de pesquisa científica: reflexão a partir da clínica fonoaudiológica. **Psicologia: Reflexão e Crítica**, v. 22, n. 3, p. 422–429, 2009. Disponível em: https://www.scielo.br/j/prc/a/Rjm8bQcZJjSn4MXZCpNzyLj.

PFLEEGER, C. P.; PFLEEGER, S. L. **Security in computing**. 5. ed. Upper Saddle River, NJ: Prentice Hall, 2015. ISBN 978-0134085043.

POROLLI, M.; RAMOS, P. **CPL malware in Brazil: somewhere between banking trojans and malicious emails**. [*S. l.*], 2015. Acesso em: 11 jul. 2025. Disponível em: https://web-assets.esetstatic.com/wls/en/papers/white-papers/CPL-Malware-in-Brasil-zx02m.pdf.

RAMANUJAM, P. K. e Hrishikesh Devgude e S. Uma análise abrangente das técnicas de ofuscação de software. **International Journal of Scientific Research in Computer Science, Engineering and Information Technology**, 2023.

SIKORSKI, M.; HONIG, A. **Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software**. San Francisco: No Starch Press, 2012. Disponível em: http://repo.darmajaya.ac.id/4350/1/Practical%20Malware%20Analysis_%20The%20Hands-On%20Guide%20to%20Dissecting%20Malicious%20Software%20(%20PDFDrive%20).pdf. Acesso em: 15 jul. 2025.

Sindicato dos Bancários de Ponta Grossa e Região. **Brasil é o epicentro de ataques bancários por trojan: veja como se proteger**. 2024. Acesso em: 24 jul. 2025. Disponível em: https://bancariospnr.org.br/posts/brasil-e-o-epicentro-de-ataques-bancarios-por-trojan-veja-como-se-proteger.

Sindicato dos Trabalhadores em Processamento de Dados e Tecnologia da Informação do Estado de São Paulo. **Brasil lidera ataques de trojans na América Latina**. 2025. Acesso em: 23 jul. 2025. Disponível em: https://sindpd.org.br/2025/06/10/brasil-lidera-ataques-trojans-america-latina/.

SOOD ADITYA K.; ENBODY, R. J. Crimeware-as-a-service—a survey of commoditized crimeware and service delivery models. **International Journal of Critical Infrastructure Protection**, v. 4, n. 1, p. 3–17, 2011.

STALLINGS, W.; BROWN, L. **Computer security: principles and practice**. 4. ed. Hoboken, NJ: Pearson, 2018. ISBN 978-0134794105.

The MITRE Corporation. **System Binary Proxy Execution: Control Panel (T1218.002)**. 2025. Sub-técnica T1218.002 para evasão de defesa usando control.exe. Disponível em: https://attack.mitre.org/techniques/T1218/002/.

TOMMASI, F.; CATALANO, C.; TAURINO, I. Browser-in-the-middle (bitm) attack. **International Journal of Information Security**, v. 21, p. 179 – 189, 2021.

YOU, I.; YIM, K. Malware obfuscation techniques: A brief survey. p. 297–300, 2010.