CENTRO UNIVERSITÁRIO DO PARÁ - CESUPA
ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO - ARGO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

NÍCOLAS DE OLIVEIRA ROCHA
PAULA D'NAZARÉ DA SILVA MARTINS
FABIAN VICTOR REIS PFEIFF

REVISÃO SISTEMÁTICA SOBRE A METODOLOGIA DE ATAQUE DOS *WORMS* ILOVEYOU, WANNACRY, MIRAI E LION NOS SISTEMAS OPERACIONAIS WINDOWS E LINUX

BELÉM
2023

NÍCOLAS DE OLIVEIRA ROCHA
PAULA D'NAZARÉ DA SILVA MARTINS
FABIAN VICTOR REIS PFEIFF

REVISÃO SISTEMÁTICA SOBRE A METODOLOGIA DE ATAQUE DOS *WORMS* ILOVEYOU, WANNACRY, MIRAI E LION NOS SISTEMAS OPERACIONAIS WINDOWS E LINUX

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Pará como requisito para obtenção do título de Bacharel em Ciência da Computação na modalidade ARTIGO.

Orientador: Esp. Eudes Danilo da Silva Mendonça.

BELÉM
2023

NÍCOLAS DE OLIVEIRA ROCHA
PAULA D'NAZARÉ DA SILVA MARTINS
FABIAN VICTOR REIS PFEIFF

REVISÃO SISTEMÁTICA SOBRE A METODOLOGIA DE ATAQUE DOS *WORMS* ILOVEYOU, WANNACRY, MIRAI E LION NOS SISTEMAS OPERACIONAIS WINDOWS E LINUX

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Pará como requisito para obtenção do título de Bacharel em Ciência da Computação na modalidade ARTIGO.

Banca examinadora

Data da Defesa: 11 / 12 / 2023

gov.br Documento assinado digitalmente
EUDES DANILO DA SILVA MENDONCA
Data: 19/12/2023 15:18:52-0300
Verifique em https://validar.iti.gov.br

Prof. Esp. Eudes Danilo da Silva Mendonça - CESUPA

Orientador e Presidente da Banca

gov.br Documento assinado digitalmente
VITOR HUGO FREITAS GOMES
Data: 19/12/2023 15:05:09-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Vitor Hugo Freitas Gomes - CESUPA

Examinador Interno

gov.br Documento assinado digitalmente
JOHNNY MARCUS GOMES ROCHA
Data: 21/12/2023 18:13:55-0300
Verifique em https://validar.iti.gov.br

Prof. MSc. Johnny Marcus Gomes Rocha - CESUPA

Examinador Interno

Dados Internacionais de Catalogação-na-publicação (CIP)
Biblioteca do CESUPA, Belém – PA

Rocha, Nícolas de Oliveira.

Revisão sistemática sobre a metodologia de ataque dos *worms* ILOVEYOU, WannaCry, Mirai e Lion nos sistemas operacionais Windows e Linux / Nícolas de Oliveira Rocha, Paula D'Nazaré da Silva Martins, Fabian Victor Reis Pfeiff; orientador Eudes Danilo da Silva Mendonça. — 2023.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário do Estado do Pará, Belém, 2023.

Segurança de dados – Computação. 2. Sistemas operacionais (Computador). 3. Linux (Sistema operacional de computador). 4. Windows (Sistema operacional de computador). I. Martins, Paula D'Nazaré da Silva. II. Pfeiff, Fabian Victor Reis. III. Mendonça, Eudes Danilo da Silva, orient. IV. Título.

CDD 23ª ed. 005.43

Dedicamos este trabalho a todos que nos ajudaram e apoiaram em nossa jornada conturbada.

AGRADECIMENTOS

Desde 2019 até 2023, posso dizer que foram os melhores e mais caóticos anos da minha vida. Parece irônico, mas é a melhor maneira de resumir tudo que aconteceu nesta jornada de cinco anos que passei no CESUPA. Dito isso, não posso deixar de agradecer a todos que me ajudaram a chegar aonde estou.

Primeiramente tenho que agradecer a minha família: minha avó Ana Maria, minha mãe Ana Cláudia, meu tio Emanuel Rocha e meu primo Gilberto Freitas, pela educação e orientação que eles me passaram, contribuindo para a formação da pessoa na qual sou hoje.

Não posso deixar de agradecer a moça que ficou grudada comigo e teve que me aturar desde o início, a minha parceira Paula Martins, que nunca deixou de ficar do meu lado nos diversos momentos e situações, sempre me apoiando e ajudando a não perder a sanidade em todos os sufocos, tensões e nos trabalhos cheios de gambiarras ou que foram necessários perder noites de sono para concluir, só ela sabe o que eu passei. No final, sempre foi uma alegria e conforto passar um tempo juntos em algum ambiente isolado nas unidades do CESUPA.

Devo agradecer também a todos os meus amigos e colegas que conheci durante este tempo, em especial o Breno Reis, por ter feito parte de um dos melhores momentos desta jornada.

Quem também merece meus agradecimentos é o Fabian Pfeiff, alguém que mal esperava reaparecer depois de três anos sumido e aceitou embarcar nessa empreitada no último momento para concluir a nossa jornada.

Agradeço ao meu orientador Eudes Mendonça e o professor Vitor Hugo pelo apoio e orientações que deram durante todo o desenvolvimento do trabalho, sem eles este trabalho não seria possível. Também ao professor Johnny Rocha por ter aceitado participar da banca.

A todos os professores que contribuíram significativamente para minha formação acadêmica.

A todos os funcionários do CESUPA que conheci no período que estagiei no CTIC, em especial os meus supervisores Aureliano Lucas e Felipe Toshio, que passei a maior parte das manhãs realizando várias missões na unidade de direito, e passando os tempos livres de forma descontraída na sala do CTIC.

Por último e não menos importante, agradeço a todas as outras pessoas que contribuíram com a minha jornada, mas que não pude mencionar aqui.

Nícolas de Oliveira Rocha

Agradeço, primeiramente, aos meus pais, Elza e Paulo, pelo amor e apoio incondicional durante toda minha vida, pelo grande incentivo aos estudos e por todo esforço para me garantir uma educação de qualidade. Se eu cheguei até aqui foi devido a vocês.

Agradeço aos meus amigos, que sempre estiveram ao meu lado, me apoiando e me encorajando durante essa jornada. Em especial, Nicolas e Fabian, que estiveram comigo durante a produção deste trabalho de conclusão do curso, não só pela colaboração e dedicação como também pelos momentos de descontração os quais tornaram o processo muito mais leve.

Agradeço ao professor e orientador, Eudes Mendonça, por aceitar conduzir nosso trabalho de pesquisa e pelas valiosas contribuições dadas durante todo o processo.

Agradeço a todos os professores do curso de Ciência da Computação do CESUPA pela alta qualidade técnica de suas aulas.

Agradeço ao CTIC da unidade Argo, pela oportunidade de obter minha primeira experiência profissional, pelo apoio e aprendizado fornecidos por excelentes profissionais.

Agradeço, por fim, ao grupo BTS, que foram meu refúgio durante esses anos, que através de suas músicas, me deram conforto e força para continuar e chegar até aqui.

Paula D'Nazaré da Silva Martins

Esse trabalho foi uma experiência única, e devo agradecer a todos que me acompanharam nessa jornada árdua, principalmente o Nicolas Rocha e a Paula Martins por sempre estarem do meu lado durante esse trabalho, e ao nosso orientador Eudes e o professor Vitor por estarem lá quando precisávamos.

Agradeço a todos os meus amigos e familiares que me acompanharam e me deram o apoio que me fortaleceu nessa jornada. Agradeço ao meus pais Marcelly Pfeiff e Garder Pfeiff por me darem suporte, e aos meus grandes amigos Matheus Castanho e João Vitor por serem uma das poucas pessoas que toleram as minhas gigantes e tediosas conversas dentro dos mais variados assuntos, e por serem uns grandes amigos, principalmente o Mateus por ter me presenteado com um dos melhores cafés que já tomei na vida depois da apresentação do trabalho, ao Paulo José por ser um dos melhores amigos que eu tive, e uns dos que a sua companhia me fizeram uma pessoa melhor.

Agradeço ao André Lima por sempre resolver os problemas dos alunos, e ter me ajudado muito durante o curso, a Alessandra Natasha por ser uma coordenadora de curso atenciosa, ao Ricardo Casseb por ser um dos professores mais exigentes e legais que já conheci, me ajudou a crescer muito como pessoa, e ao Jesus Nazareno por ser um ótimo supervisor de estágio e ter sido uma das pessoas mais maravilhosas que já conheci e ao José Renato por ser um ótimo gestor acadêmico e também ser umas das melhores pessoas que trabalhei junto. O pôr fim, agradeço a todas as outras pessoas que não foram citadas acima e tiveram um peso significante na minha vida, eu sempre tive problema para lembrar nomes, mas nunca esquecerei o que foi feito por mim, eu sou grato a todos vocês.

Fabian Victor Reis Pfeiff

RESUMO

A cada dia, diversos dispositivos são atacados por diferentes ameaças ao redor do mundo, afetando tanto usuários domésticos quanto governos ou grandes instituições, fazendo com que os especialistas de segurança da informação tenham que trabalhar duro para defender destes ataques. Normalmente, os invasores utilizam *malwares* como instrumento de ataque, sendo um dos mais utilizados nos últimos tempos os *worms*. Para terem sucesso nos ataques, os atacantes e cibercriminosos desenvolvem os *worms* com foco nos sistemas operacionais mais utilizados: Windows e Linux, sendo que os dois possuem diferentes arquiteturas e paradigmas. Por isso, há um grande esforço dos especialistas de segurança da informação de trabalharem para proteger esses dois sistemas. Baseando-se neste contexto, o presente trabalho tem como objetivo comparar os *worms*: WannaCry, ILOVEYOU, Mirai e Lion, focando nas suas metodologias de ataque e as metodologias de defesa de seus respectivos sistemas-alvo, a partir de uma revisão sistemática de artigos e literaturas acadêmicas obtidos por uma pesquisa em repositórios e acervos online. Ao final, esperamos que este artigo contribua para pesquisas futuras na área de cibersegurança.

Palavras-chave: *malwares*; *worms*; sistemas operacionais; cibersegurança.

ABSTRACT

Every day, several devices are attacked by different threats around the world, affecting both home users and governments or large institutions, making information security specialists have to work harder to defend against these attacks. Typically, attackers use malwares as an attack instrument, one of the most used in recent times being worms. To be successful in attacks, the attackers and cybercriminals develop worms focused on the most used operating systems: Windows and Linux, both of which have different architectures and paradigms. Therefore, there is a great effort by information security specialists to work to protect these two systems. Based on this context, the present work aims to compare the worms: WannaCry, ILOVEYOU, Mirai and Lion, focusing on their attack methodologies and the defense methodologies of their respective target systems, based on a systematic review of articles and academic literatures obtained by searching through repositories and online collections. In the end, we hope that this article contributes to the future research in the area of cybersecurity.

Keywords: malwares; worms; operating systems; cybersecurity.

LISTA DE ILUSTRAÇÕES

FIGURAS

QUADROS

LISTA DE ABREVIATURAS E SIGLAS

API	Interface de Programação de Aplicações, do inglês *Application Programing Interface*
BIND	Berkeley Internet Name Domain
C&C	Comando e Controle, do inglês *Command and Control*
DHT	*Distributed Hash Table*
DDoS	Ataque Distribuído de Negação de Serviço, do inglês *Distributed Denial of Service Attack*
DNS	Sistemas de Nome de Domínio, do inglês *Domain Name System*
ELF	Formato Executável e de Ligação, do inglês *Executable and Linkable Format*
FTP	Protocolo de Transferência de Arquivos, do inglês *File Transfer Protocol*
GPL	GNU General Public License
GUI	Interface de Usuário Gráfica, do inglês *Graphical User Interface*
I/O	Entrada/Saída, do inglês *Input/Output*
IOC	Indicador de Comprometimento, do inglês *Indicator of Compromise*
IoT	Internet das Coisas, do inglês *Internet of Things*
IPC	Comunicação entre processos, do inglês *Inter Process Communication*
IRC	*Internet Relay Chat*
JPEG	Joint Photographic Experts Group
MS-DOS	Microsoft Disk Operating System

NHS	Serviço Nacional de Saúde, do inglês *National Health Service*
NTFS	NT File System
NTVDM	NT Virtual DOS Machine
OS	Sistema Operacional, do inglês *Operating System*
POSIX	Interface Portável entre Sistemas Operacionais, do inglês *Portable Operating System Interface*
P2P	*Peer-to-peer*
RAS	*Remote Access Server*
SSH	*Secure Shell*
SMB	*Server Message Block*
Telnet	*Teletype Network*
TCP	Protocolo de Controle de Transmissão, do inglês *Transmission Control Protocol*
TF-IDF	Term Frequency-Inverse Document Frequency
UAC	Controle de Conta de Usuário, do inglês *User Account Control*
UDP	Protocolo de Datagrama do Usuário, do inglês *User Datagram Protocol*
µTP	*Micro Transport Protocol*
VBScript	Visual Basic *Script*
WScript	Windows Script Host
WSL	Windows Subsystem for Linux

SUMÁRIO

PARTE I - CONTEXTUALIZAÇÃO

1.1 Revisão Bibliográfica

1.1.1 Sistemas Operacionais

Segundo Stallings (2018) sistemas operacionais são *softwares* responsáveis por controlar a execução de *softwares*, interagir e gerenciar o *hardware* da máquina, agindo como um intermediário entre os *softwares* e o *hardware* do computador. Sistemas Operacionais tem como objetivo fazer com que os computadores sejam mais amigáveis e convenientes para usá-los, eficientes no uso de recursos do computador e escaláveis para facilitar o desenvolvimento de novas aplicações, recursos e funções.

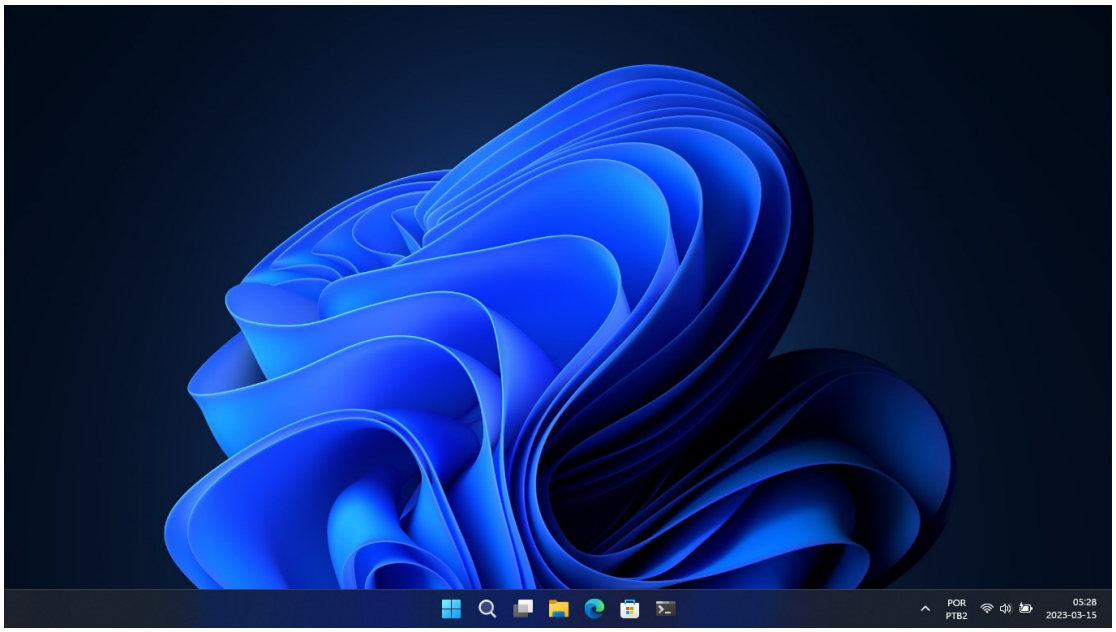
Do final dos anos 1940 até meados dos anos 1950, os computadores exigiam que os usuários interagissem diretamente com o *hardware* (Tanenbaum; Bos, 2015; Stallings, 2018). Mais tarde, foi criado o conceito de *batch OS* para facilitar e melhorar a usabilidade dos computadores. O primeiro *batch OS* foi desenvolvido pela General Motors para ser usado no IBM 701, logo em seguida vieram os sistemas *batch* multiprogramados com intuito de resolver as limitações de performance dos dispositivos I/O (Tanenbaum; Bos, 2015; Stallings, 2018). Um dos primeiros sistemas operacionais modernos foi o UNIX, criado em 1970 por pesquisadores da Bell Labs. O sistema foi desenvolvido para o computador DEC PDP-7 e escrito em linguagem C, sendo considerado um marco histórico, pois os sistemas eram escritos em linguagem *assembly*, o que possibilitava serem executados em vários computadores (Tanenbaum; Bos, 2015; Stallings, 2018). O sistema Unix acabou se tornando bastante popular em academias, agências de governos e outras organizações (Tanenbaum; Bos, 2015; Stallings, 2018).

Os sistemas operacionais possuem dois modos de operações, que são o modo kernel e o modo usuário. No modo kernel onde se encontram os componentes básicos do sistema, o acesso ao *hardware* da máquina é completo. Já no modo usuário, encontra-se a *shell* ou a interface de usuário (podendo ser uma interface baseada em texto ou uma GUI) e as aplicações (como navegadores de internet, compiladores e outros), havendo restrições sobre quais instruções podem ser executadas, instruções que afetam o controle da máquina ou I/O são proibidos de executar no modo usuário (Tanenbaum; Bos, 2015).

1.1.1.1 Windows

Windows é uma família de sistemas operacionais criado e desenvolvido pela Microsoft Corporation. O sistema possui uma grande presença no setor empresarial, industrial e especialmente doméstico, onde é o seu mercado mais popular, com 74,18% de computadores desktop no mundo utilizando o sistema (Figura 1) (Adekotujo *et al.*, 2020; Statcounter, 2023).

Figura 1 - Windows 11



Fonte: Autores (2023)

Originalmente, a primeira versão do Windows, Windows 1.0, foi lançada em 1985 como uma camada gráfica por cima do sistema operacional MS-DOS.

Por alguns anos, as versões domésticas do Windows utilizavam o kernel baseado no MS-DOS, a partir do Windows XP (2001) a frente o sistema utiliza o kernel baseado no Windows NT (Adekotujo *et al.*, 2020; Carpenter, 2011; Microsoft, 2015). O Windows NT foi lançado em julho de 1993, sendo a primeira versão o Windows NT 3.1, originalmente tendo como foco o mercado corporativo e de servidores. Foi a primeira versão do Windows a suportar programação 32-bits em vários microprocessadores da época como o Intel 80386 e o Pentium, onde também foram introduzidas várias funcionalidades como RAS, subsistema de aplicações OS/2[1] e POSIX, e o sistema de arquivos NTFS (Kavas; Feitelson, 2001). O Windows NT foi criado com as seguintes metas: ser um sistema extensível, portátil, robusto, confiável,

[1] Sistema operacional desenvolvido em 1987 pela IBM e Microsoft, foi descontinuado em 2001.

compatível com aplicações de versões anteriores do sistema e, ser responsivo e rápido em cada plataforma de *hardware* suportado pelo mesmo (Custer, 1993).

Windows suporta algumas arquiteturas de hardware como IA-32, x86-64 e ARM (ARM e ARM64), e chegou a suportar em versões anteriores as arquiteturas Alpha, MIPS, PowerPC e Itanium (Yosifovich *et al.*, 2017; Custer, 1993). Por isso, o sistema possui outras versões focadas em vários tipos de dispositivos da Microsoft e seus parceiros, como o console de videogame Xbox, os óculos de realidade aumentada HoloLens, alguns dispositivos colaborativos que são vendidos pela Microsoft ou seus parceiros, e os antigos *smartphones* Windows Phone. Todos esses dispositivos mencionados anteriormente utilizam versões modificadas do sistema (Yosifovich *et al.*, 2017).

O kernel engloba uma série de componentes essenciais que são executados em modo kernel e implementam funcionalidades básicas como o gerenciador de memória virtual, gerenciamento de recursos, I/O, sistema de arquivos e IPC (Custer, 1993). Também possui vários subsistemas de APIs como: Win32, WinRT[2], NTVDM[3] e WSL[4] (Yosifovich *et al.*, 2017).

[2] API do Windows para a execução dos chamados *Windows Apps*, tipo de aplicação que pode ser executada em diversos dispositivos e plataformas diferentes.

[3] Subsistema do Windows para execução de *softwares* de MS-DOS e Windows 3.1, disponível apenas nas versões 32-bits do sistema.

[4] Subsistema do Windows para execução de *softwares* de Linux.

1.1.1.2 Linux

"Linux" é normalmente referido ao grupo de sistemas operacionais que são construídos ao redor do Linux kernel, junto com ferramentas e bibliotecas de outros projetos (Juell, 2017). O kernel é do tipo UNIX, originalmente desenvolvido pelo finlandês Linus Torvalds e lançado em 1991 para os computadores de arquitetura x86. O Linux possui uma licença de código aberto e software livre, fazendo com que o mesmo seja bastante popular na comunidade acadêmica, *hackers* e ativistas (Beck *et al.*, 1996; Guimarães, 2005).

Segundo Guimarães (2005), o Linux foi originalmente criado por causa das frustrações que Linus Torvalds tinha com o sistema operacional MINIX[5], que possuía uma série de limitações técnicas e por não ser gratuito na época. O desenvolvimento do sistema foi anunciado por Linus Torvalds em um grupo na USENET[6] em 1991 (Juell, 2017).

Foi desenvolvido originalmente para a arquitetura x86, e em versões futuras ganhou suporte para inúmeras arquiteturas como x86-64, Itanium, PowerPC, SPARC, ARM (ARM e ARM64), MIPS, DEC Alpha entre outros (Beck *et al.*, 1996; Juell, 2017). Isso fez com que o Linux fosse bastante popular em diversos setores como no mercado de servidores, dispositivos embarcados, computação em nuvem e em *smartphones*, sendo utilizado como base para o sistema operacional mobile Android da Google. Em 2021, smartphones Android possuem 3 bilhões de usuários ativos em todo mundo, também é usado em 97% em servidores *front-end* da Web e empresas provedoras de nuvem como a Amazon utilizam o sistema em mais de 200 mil instâncias (Yoshimura, 2016; Google, 2021).

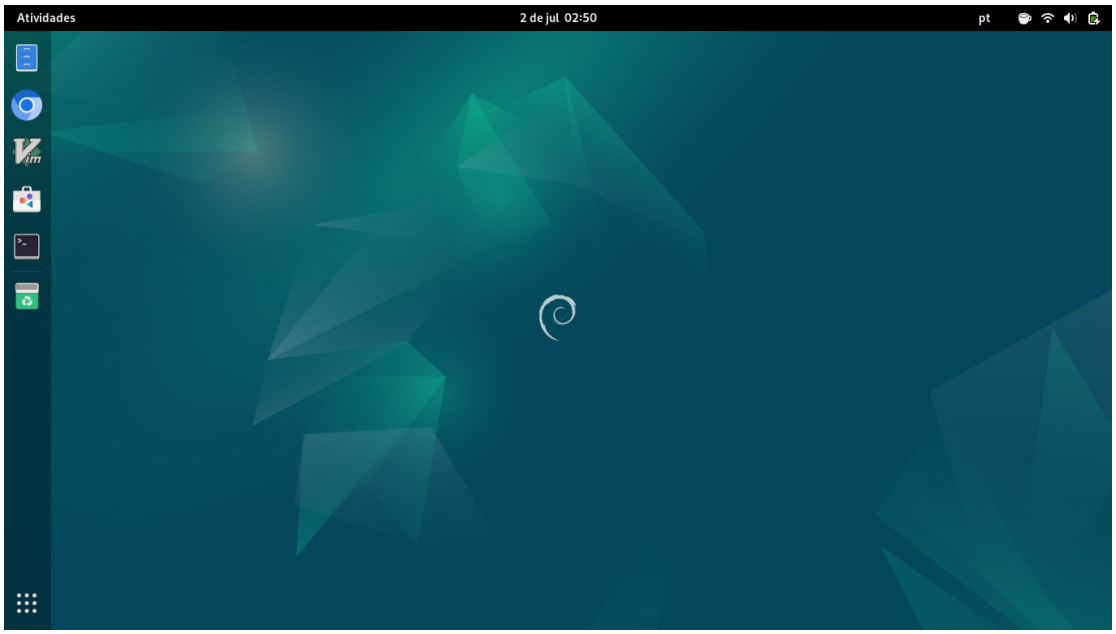
Por não ser um sistema operacional completo e pela sua licença de código aberto, Linux é normalmente empacotado como uma distribuição. Em uma distribuição Linux, estão presentes o kernel, bibliotecas e ferramentas de outros projetos como os utilitários do projeto GNU, Busybox ou musl, aplicações, onde algumas distribuições incluem ambientes desktop como GNOME ou KDE. Há inúmeras distribuições Linux, as mais populares são: Ubuntu, Debian (Figura 2), Kali Linux, Red Hat Enterprise Linux, SUSE Linux e Arch Linux (Juell, 2017).

---

[5] Sistema operacional tipo UNIX que utiliza a arquitetura de microkernel e possui como foco sistemas embarcados e sistemas com recursos limitados, foi criado por Andrew Tanenbaum em 1987.

[6] Canal de comunicação descentralizado criado em 1979 onde os usuários trocam mensagens de texto por grupos, é considerado um antecessor dos fóruns modernos da internet.

Figura 2 - Debian 12 "*Bookworm*" com GNOME desktop



Fonte: Autores (2023)

1.1.2 *Worms*

De acordo com Audelo González *et al*. (2013), os *worms* de computador tiveram seu início em 1979 nos laboratórios XEROX PARC, quando John Shich e Dave Boggs pretendiam analisar padrões de comportamento de tráfego nas redes sob determinadas cargas de trabalho. Os cientistas desenvolveram um programa para simular uma alta carga de trabalho, o que levou ao primeiro protótipo de *worm* de computador (*apud* Guevara López *et al.*, 2015). Em 1988, o estudante da Universidade de Cornell, Robert Tappan Morris, criou um dos primeiros *worm* de computador registrado. Apesar de não ter tido intenções maliciosas, o *worm* se espalhou rapidamente, afetando cerca de 60,000 sistemas UNIX, o que representava cerca de 15% da internet daquele período (Erbschloe, 2005 *apud* Guevara López *et al.*, 2015).

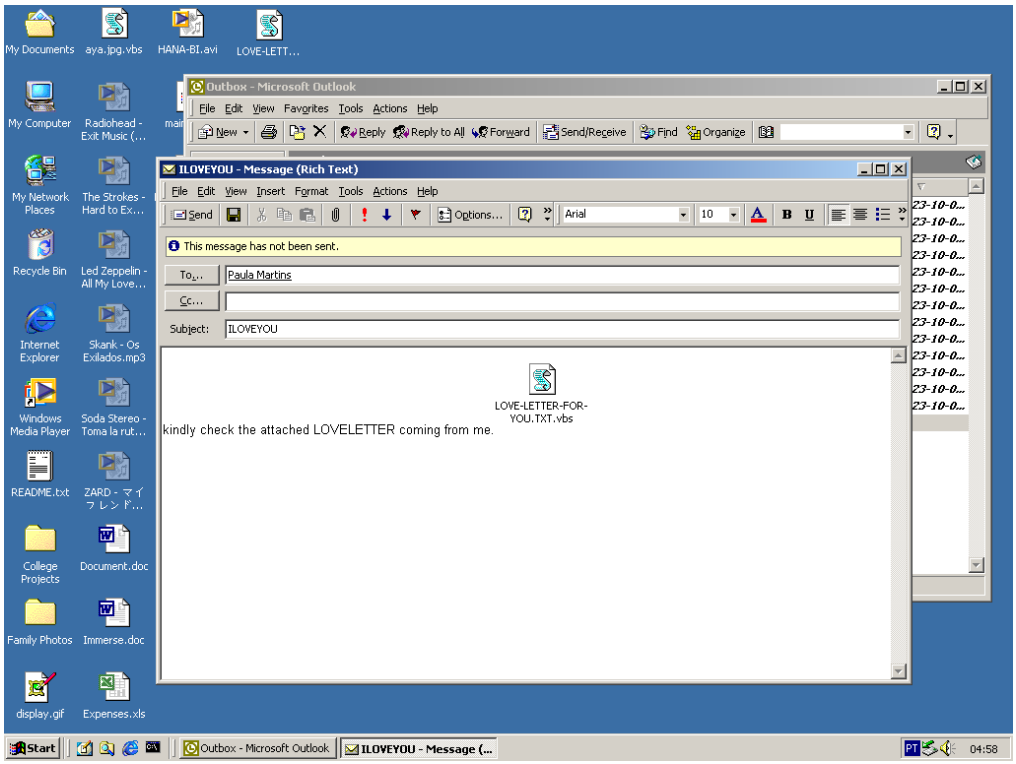
Worm é um tipo de *malware* que tem a capacidade de copiar a si mesmo e se espalhar por uma rede em busca de vulnerabilidades de segurança, sem a necessidade de ativação ou intervenção humana (Aycock, 2006). Essas cópias infectam computadores e servidores vulneráveis que estão conectados ao dispositivo inicialmente infectado. O processo de autorreplicação, execução e propagação permitem que infecções baseadas em *worms* se propaguem rapidamente pelas redes de computadores e pela Internet (Aycock, 2006; Microsoft, 2023; Kaspersky, 2014?).

Os *worms* utilizam diversos métodos para se espalhar pela rede, incluindo anexos de e-mail, sites infectados com código oculto, downloads e servidores FTP, mensagens instantâneas por meio de aplicativos de mensagens, redes de compartilhamento de arquivos P2P e exploração de acesso compartilhado em redes. Essas técnicas permitem que os *worms* se espalhem por vários canais e infectam uma ampla variedade de dispositivos, unidades e arquivos em uma rede (Kaspersky, 2014?).

1.1.2.1 ILOVEYOU

O *worm* ILOVEYOU, também conhecido como "*Love bug*", foi um vírus que se espalhou via e-mail disfarçado de uma carta de amor, infectando milhões de computadores em 4 de maio de 2000. O *malware* capturava o livro de endereços do Microsoft Outlook infectado e envia uma cópia de si mesmo para todos os contatos contidos na lista. Outro aspecto sobre o *worm* ILOVEYOU foi que sua infecção inicial foi feita em um simples programa VBScript (Figura 3) (Root, 2022; Patten, 2017).

Figura 3 - Windows 2000 infectado pelo ILOVEYOU *worm*



Fonte: Autores (2023)

Além do spam de e-mail enviado em nome dos usuários infectados, o *worm* se espalhava através do *IRC*[7]. O *malware* baixava um programa *trojan* para capturar senhas de e-mails e enviar para o criador do *worm*. Também ocultava ou corrompia vários arquivos no disco rígido do sistema infectado, incluindo arquivos de música MP3, imagens JPEG, *scripts* e cópias de páginas da web (Root, 2022).

Umas das principais causas da efetividade do ataque foi pela engenharia social utilizada, que foi suficiente para deixar os usuários curiosos e serem persuadidos a abrirem o e-mail

[7] Sistema de *chat* baseado em texto para mensagens instantâneas criado em 1988, é considerado um antecessor dos mensageiros instantâneos.

(Welivesecurity, 2017). Outro fator que contribuiu para o ataque é o comportamento padrão do sistema operacional Windows de esconder as extensões de arquivos, o nome do vírus engana as vítimas pensando que é um simples arquivo de texto (Ducklin, 2020).

O *malware*, de origem nas Filipinas, se espalhou ao redor do mundo rapidamente, afetando instituições como o parlamento britânico, o congresso dos Estados Unidos, a força aérea dos Estados Unidos entre outras organizações (Bishop, 2000). Segundo Howard e Prince (*apud* Patten, 2017) estima-se que o *worm* ILOVEYOU em nove dias infectou aproximadamente 50 milhões de sistemas. Os autores do *worm*, os estudantes universitários Onel de Guzman e Reonel Ramones, chegaram a ser presos em Manila, mas não foram condenados por falta de legislações contra cibercriminosos nas Filipinas (Welivesecurity, 2017; Ducklin, 2020).

1.1.2.2 WannaCry

O WannaCry (também conhecido como *Wanna Decrypt0r* ou *Wanna Crypt0r*) é um *malware* que causou um dos maiores ataques cibernéticos da história em 12 de maio de 2017, causando um dano de 300 mil sistemas em mais de 150 países, afetando vários setores e organizações como a Sistema Nacional de Saúde do Reino Unido, o Banco da China, prefeituras de várias cidades no Japão, o Ministério Interior da Rússia, FedEx, Renault, Petrobrás e outros (Akbanov *et al.*, 2019; Ghosh, 2017).

Figura 4 - Windows 8.1 infectado pelo WannaCry *malware*

Fonte: Autores (2023)

O *malware* é classificado como um *ransomware* e *worm* de *Intel 80386 PE32*[8] *Executable* para o Windows, sendo conhecido como Ransom:Win32/WannaCrypt. Um ano após o ataque, o Departamento de Justiça dos Estados Unidos acusa o Lazarus Group[9] de ser o criador do *malware* (Tsing, 2019).

O módulo do *worm* se aproveita de dois *exploits* que foram vazados em abril de 2017, *EternalBlue* e *DoublePulsar* (Akbanov *et al.*, 2019; Chen; Bridges, 2017). O primeiro *exploit* que o módulo de *worm* utiliza, *EternalBlue*, se aproveita de uma vulnerabilidade do protocolo SMB que permite que os invasores executem código remoto em máquinas infectadas enviando

[8] Formato de arquivos executáveis utilizado pela família de sistemas operacionais Windows, também pode ser chamado de PE/COFF.

[9] Grupo de cibercriminosos que é acreditado que sejam administrados pelo governo da Coreia do Norte, foram envolvidos em diversos ataques cibernéticos pelo mundo.

mensagens para servidores que utilizam o protocolo SMB na versão 1.0, conectando as portas TCP 139 e 445 em sistemas Windows não atualizados, em especial nas versões a partir do Windows XP ao Windows 8.1. Já o segundo *exploit*, *DoublePulsar*, é um *backdoor* em que permite que os invasores executem e instalem um *malware* adicional no sistema infectado (Akbanov *et al.*, 2019). Segundo Malwarebytes ([.s.d]) a Microsoft já havia lançado uma atualização de segurança anteriormente para corrigir as vulnerabilidades *EternalBlue*, porém vários usuários utilizavam versões não atualizadas do Windows, fazendo com que esses sejam suscetíveis ao ataque.

Ao executar o módulo de *ransomware*, ele é instalado no sistema como um serviço que será executado sempre que o computador for reiniciado, encriptando vários documentos, imagens, áudios, arquivos compactados e outros utilizando uma criptografia RSA com uma chave AES simétrica. Então, é exibido uma janela com instruções de pagamento em vários idiomas para que o usuário pague os invasores (no início, os invasores demandavam 300 dólares em bitcoin, e após certo tempo, o valor aumentava para 600 dólares) e teoricamente, ocorria o resgate dos arquivos (Moreira *et al.*, 2017; Malwarebytes, [.s.d]).

1.1.2.3 Lion

O Linux Lion ou Lion é um *malware* criado por um hacker chines conhecido como Lion com o propósito de advertir ao Departamento de Educação japonês sobre o uso de livros controversos nas escolas do Japão. Segundo o criador do *malware*, os livros legitimam os crimes de guerra cometidos pelo Japão contra a China, Coreia do Norte e Coreia do Sul. O *worm* aproveitou-se de uma vulnerabilidade de transbordamento de dados TSIG no software do servidor BIND utilizado para fornecer instruções ao DNS, para converter endereços Web em endereços IP. O *worm* verifica a porta 53 dos endereços IP da rede classe B e, se encontrar um servidor de nomes em execução no alvo, é lançada sua exploração BIND. Se a exploração for bem-sucedida, uma cópia do *worm* é baixada na forma de um pacote, onde esse pacote é extraído e os *scripts* de inicialização são executados (Arora, 2013).

Os dispositivos infectados enviam as senhas de root para um site hospedado na China, onde as senhas serão descriptografadas, e assim, obtém-se acesso administrativo aos sistemas alvos do ataque. Além disso, foi instalado nos sistemas infectados um *rootkit* para o *worm* não ser detectado. O *worm* não pode ser removido dos sistemas infectados, sendo necessário reformatar o sistema (Arora, 2013).

1.1.2.4 Mirai

Responsável por causar um dos maiores ataques de DDoS da história, o Mirai é um *malware* escrito em linguagem C e classificado como um *worm* de executável ELF[10], sendo conhecido como ELF Linux/Mirai. Foi criado por Anna-Senpai e descoberto primeiramente em agosto de 2016 pelo grupo *hacker* MalwareMustDie[11]. O Mirai teve atenção da mídia meses depois quando vários alvos como o pesquisador de cibersegurança Brian Krebs, o provedor de hospedagem OVH, a Oracle DYN, o país da Libéria e a Deutsche Telekom sofreram ataques de DDoS por uma rede *botnet* criada pelo *malware*. Com a disponibilidade de seu código fonte pelo autor do *malware* no final de setembro de 2016, surgiram inúmeras variações e clone do Mirai, causando diversos ataques ao redor do mundo (Holub, 2020; Bursztein, 2017; Jaramillo, 2018).

O *worm* tem como objetivo infectar dispositivos IoT, que foram negligenciados por pesquisadores de cibersegurança por vários anos, como roteadores domésticos, câmeras IP através de uma varredura contínua das portas TCP 22, 23, 5747, etc (Holub, 2020; Jaramillo, 2018). Apesar da versão original do Mirai ser tecnicamente simples utilizando um conjunto de logins e senhas padrões conhecidos e usualmente usados em dispositivos IoT, acabou se tornando extremamente eficiente infectando mais de 600 mil dispositivos (Bursztein, 2017).

Utiliza uma técnica de força bruta para adivinhar logins e senhas para obter acesso e assim infectar o dispositivo. O dispositivo infectado verifica outras redes à procura de mais dispositivos IoT para aceitar comandos DDoS de um servidor de comando e controle, com intuito de criar um *botnet* (Antonakakis *et al.,* 2017). Segundo Margolis (2017) a razão pela grande influência do Mirai em comparação a outros *worms* é devido a sua eficiência em infectar novos dispositivos.

A identidade dos reais criadores do Mirai, os estudantes da Universidade de Rutgers nos Estados Unidos, Paras Jha, Josiah White e Dalton Norman, inicialmente negaram envolvimento com o Mirai, mas em 2018 o trio se declarou culpado pelos danos causados pelo *worm*. O nome Mirai veio após a série de animação japonesa Mirai Nikki (Shapiro, 2023).

---

[10] Formato de arquivos executáveis utilizados em vários sistemas operacionais baseados em UNIX, em especial Linux e BSD.

[11] Grupo de *hackers white hat* criado em 2012, possui como objetivo reduzir infecções de malware na internet.

1.2 Problema

Durante os anos, pesquisadores da área de segurança documentaram o comportamento dos *worms* e os mecanismos de defesa do sistema-alvo, porém, cada sistema se comporta de uma forma diferente, por isso, os *worms* têm diferentes estratégias e metodologias de ataque. Apesar de haver várias documentações a respeito de *worms* e sistemas operacionais, nesses últimos 20 anos não há trabalhos que mensure ou compare a ação entre os *worms* e os sistemas-alvo, e que também destacam os paradigmas de segurança utilizados ao longo do período.

1.3 Justificativa

Os variados materiais acadêmicos existentes nos últimos 20 anos sobre *worms* apresentam uma escassez de artigos sobre o diferente comportamento de *malwares* em cada sistema-alvo ou como esses sistemas se defendem. Sendo assim, os resultados obtidos nesta pesquisa podem contribuir para estudos sobre novas ameaças na área de segurança digital. Este tipo de estudo têm uma grande relevância na área de segurança digital, pois ajudam os pesquisadores e engenheiros de segurança cibernéticos a entender as metodologias de ataque dos *malwares*, e saber como mitigá-los, contribuindo para identificar novos ataques avançados, visados ou ataques de *Zero-Day*.

1.4 Objetivos

1.4.1 Objetivo Geral

Desenvolver uma revisão sistemática sobre a forma de ataque dos *worms* ILOVEYOU, WannaCry, Lion e Mirai atacam os sistemas Windows e Linux nas últimas duas décadas, comparando o comportamento dos ataques e das defesas ao longo do período.

1.4.2 Objetivos Específicos

- Revisar materiais bibliográficos sobre os *worms* ILOVEYOU, WannaCry, Lion e Mirai, com foco no processo de ataque em seus específicos sistemas operacionais alvo;
- Analisar os métodos de ataque dos *worms* e os métodos de defesa dos sistemas-alvo;
- Comparar a metodologia de ataque dos *worms* e de defesa dos sistemas-alvo, destacando os paradigmas de segurança nas últimas duas décadas.

1.5 Estrutura do Trabalho

O trabalho consiste em um primeiro capítulo contendo a contextualização da temática, onde é apresentado um referencial teórico, abordando os conceitos dos *worms* ILOVEYOU, WannaCry, Lion e Mirai, e dos sistemas operacionais Windows e Linux; o problema de pesquisa; a justificativa e os objetivos que conduziram o estudo. Já no segundo capítulo, é apresentado o artigo contendo uma revisão sistemática de trabalhos comparando os *worms* e suas metodologias de ataque nos sistemas operacionais, uma discussão geral sobre a comparação e resultados analisados, e a conclusão onde é descrito as considerações finais junto com possíveis aperfeiçoamentos, dificuldades na elaboração do artigo e ideias para trabalhos futuros.

PARTE II - ARTIGO

2.1 Introdução

Atualmente, a infraestrutura da nossa sociedade possui uma grande dependência a computadores. Infelizmente os computadores são vulneráveis a ataques e, a cada dia, os ataques realizados por *softwares* maliciosos crescem (Aycock, 2006). *Softwares* maliciosos (ou *malware*) tem como objetivo de prejudicar e criar comportamentos indesejáveis aos usuários como roubar informações, sequestro de arquivos, modificar o comportamento do sistema operacional e outras atividades maliciosas (Tahir, 2018; Ab Razak *et al.*, 2016; Kim; Bu; Cho, 2018).

Em 2020, os *worms* foram um dos tipos de *malwares* mais utilizados em ataques, com 20,15% de incidentes reportados no site CERT.br (Cert.Br, 2022). Os sistemas operacionais alvos mais comuns são Windows com 74,14% em desktops, e Linux com 48,2% em servidores e 38,42% em IoTs (Statcounter, 2023; W3techs, 2023; Embedded, 2019). Windows e Linux são dois sistemas completamente diferentes, Windows utiliza como base o kernel NT nas suas versões mais recentes e possui alguns elementos providos de seu predecessor MS-DOS e do IBM OS/2 (Adekotujo *et al*., 2020; Carpenter, 2011). Linux é um kernel que utiliza vários padrões do UNIX, normalmente é complementado por vários outros *softwares* e componentes para se tornar um sistema operacional completo (Juell, 2017; Adekotujo *et al.*, 2020).

Como Windows e Linux são diferentes em termos de arquitetura, paradigmas, sistemas de arquivos entre outros, os *softwares* maliciosos como *worms* precisam funcionar de formas diferentes, ou seja, terão metodologias de ataque diferentes para conseguir atacá-los (Adekotujo *et al.*, 2020). Na literatura acadêmica há vários artigos e materiais documentando o comportamento e as metodologias de ataque dos *worms* nos sistemas-alvos, porém, não há materiais que fazem uma comparação sobre o comportamento dos *worms* em cada sistema operacional.

Este artigo procura fazer uma revisão sistemática sobre os *worms*: WannaCry, ILOVEYOU, Mirai e Lion, que foram desenvolvidos com intuito de atacar os sistemas operacionais Linux e Windows, e compará-los para avaliar as suas metodologias de ataque dos *worms* utilizadas e os mecanismos de defesa dos sistemas operacionais. Assim, o artigo busca contribuir para o aprofundamento de conhecimentos e desenvolvimento na área de segurança digital no âmbito acadêmico.

2.2 Metodologia de Pesquisa

Este artigo se baseia em uma revisão sistemática acerca de materiais encontrados em bibliografias e repositórios acadêmicos, tendo como foco materiais a respeito das temáticas de *worms* e sistemas operacionais. Essa pesquisa foca nas metodologias de ataque dos *worms* nos seus respectivos sistemas alvo a fim de compará-los com base nos paradigmas de cada período dos ataques. Para a seleção de materiais à revisão foram utilizados os seguintes critérios: correspondência ao tema do artigo; termos de busca; ano de publicação, onde foram revisados materiais publicados entre 2000 e julho de 2023; idioma do texto, onde foram revisados textos nos idiomas inglês e português. Os critérios mencionados são apresentados no Quadro 1:

Quadro 1 - Critérios de pesquisa

Critérios	Parâmetros
Tema do Artigo	Revisão sistemática da metodologia de ataque dos worms ILOVEYOU, WannaCry, Mirai e Lion nos sistemas operacionais Windows e Linux
Termos de Busca	worms, iloveyou worm, wannacry *OR* wannacry worm, linux lion worm *OR* lion worm linux, mirai worm *OR* mirai botnet, windows *OR* microsoft windows *OR* windows nt, linux, windows worms, linux worms
Ano de Publicação	De 2000 a julho/2023
Idioma	Inglês ou português

Fonte: Autores (2023)

A próxima etapa foi o desenvolvimento de um banco utilizando uma seleção de trabalhos mais relevantes acerca do tema proposto para avaliação e obtenção de resultados. Para isso, foram utilizados os repositórios acadêmicos Google Scholar, SciELO, IEEE Xplore, RefSeek e IRDB. Um fluxograma demonstrando a metodologia utilizada para a criação deste artigo está representada na Figura 5:

Figura 5 - Fluxograma da metodologia utilizada

Fonte: Autores (2023)

Detalhando sobre a análise dos trabalhos, devemos enfatizar que dos diversos trabalhos encontrados, vários materiais não foram considerados para a análise por falta de dados ou por não conterem informações relevantes ao desenvolvimento deste artigo.

Os trabalhos selecionados sobre testes e pesquisa do efeito de *worms* em sistemas operacionais foram aqueles que tiveram maior impacto e relevância em suas respectivas épocas. Os critérios de seleção de trabalhos sobre *worms* incluíram o ano, para que fosse possível comparar *worms* de épocas específicas, e o foco em estudos de caso e análises, pois esses trabalhos são ricos em conteúdos relacionados ao funcionamento dos *worms*.

2.2.1 Critérios de Inclusão e Exclusão

Os critérios de inclusão estabelecidos para a seleção dos estudos foram: (1) estudos que conceituam sobre as técnicas de ataque dos e defesa dos sistemas operacionais; (2) estudos que apresentaram análises as técnicas de ataque e defesa; (3) estudos que compararam as diferentes formas de ataque e defesa ao longo dos últimos 20 anos. Foram excluídos os estudos que não estavam relacionados diretamente às técnicas de ataque e defesa relacionadas aos *worms* nos sistemas operacionais Windows e Linux, bem como estudos duplicados e de baixa relevância.

2.2.2 Processo de Seleção e Análise dos Estudos

A seleção dos estudos foi realizada em duas etapas: triagem com base nos títulos e resumos. A seleção inicial de artigos para revisão sistemática foi realizada por todos os integrantes e as discordâncias foram resolvidas por consenso. Após essa análise, os estudos selecionados foram submetidos a uma avaliação detalhada do texto, avaliando a qualidade metodológica, relevância e contribuições para o tema.

2.2.3 Limitações

A escassez de materiais e literaturas acadêmicas sobre *worms* ou outras ameaças no Linux foi a principal limitação encontrada durante o desenvolvimento deste artigo. Para demonstrar este problema, foi realizada uma pesquisa com os termos de busca "windows worms" e "linux worms". Os resultados obtidos em três dos repositórios utilizados neste artigo para criar uma tabela demonstrando a disparidade entre a quantidade de trabalhos encontrados sobre *worms* no Linux e Windows. A tabela é apresentada no Quadro 5:

Quadro 2 - Disparidade do número de resultados obtidos de materiais sobre *worms* no Windows e Linux

	Google Scholar	**RefSeek**	**IEEE Explorer**
Windows	229.000	74.500.000	70
Linux	26.800	11.600.000	25

Fonte: Autores (2023)

2.3 Resultados

Durante a pesquisa de materiais bibliográficos, foram encontrados uma grande quantidade de materiais a respeito de *worms* porém, percebeu-se uma carência de materiais quanto à análise das características dos *worms* trabalhados e como afetam seus sistemas-alvo, principalmente acerca de *worms* no sistema operacional Linux onde percebe-se uma maior carência, sendo que a maioria dos trabalhos encontrados seguindo os parâmetros mencionados tangenciou do tema ou focava em outras aplicações.

Foram selecionados um total de seis trabalhos acadêmicos, observou-se que há uma disparidade entre trabalhos em relação a *worms* nos sistemas Windows e Linux causado pelo problema mencionado anteriormente. Os trabalhos acadêmicos selecionados são apresentados no Quadro 2:

Quadro 3 - Base de trabalhos acadêmicos

Trabalhos	**Repositórios**	**Termos de Busca**
Margolis, Joel et al., 2017	IEEE Xplorer	mirai worm *OR* mirai botnet
Bishop, Matt., 2000	Google Scholar	iloveyou worm
Chen, Qian; Bridges, Robert A., 2017	IEEE Xplorer	wannacry *OR* wannacry worm
Arora, Himanshu., 2013	Google Scholar	linux lion worm *OR* lion worm linux
Kambourakis, Georgios; Kolias, Constantinos; Stavrou, Angelos., 2017	IEEE Xplorer	mirai worm *OR* mirai botnet
Kao, Da-Yu; Hsiao, Shou-Ching., 2018	Google Scholar	wannacry *OR* wannacry worm

Fonte: Autores (2023)

2.3.1 Análise dos Trabalhos

2.3.1.1 An In-Depth Analysis of the Mirai Botnet (Margolis *et al.*, 2017)

Este trabalho apresenta uma análise do Mirai, focando na sua forma de propagação e propostas dispostas a solucionar futuros ataques em dispositivos IoTs. No texto é apontado que pesquisas acerca de segurança em dispositivos IoT não obteve muita visibilidade até aparecer casos recentes, devido a isso, muitas informações lançadas sobre o *worm* Mirai são inconsistentes ou incorretas. Além disso, os autores mencionam trabalhos que relacionam ataques de DDoS em dispositivos IoT, porém não apresentou formas de prevenção para ajudar em ataques similares no futuro. Os autores afirmam que através disso, a possibilidade de ataques pode ser reduzida.

Durante a análise os autores observam como cada parte do *worm* funciona individualmente e em conjunto para ter o entendimento completo de como ele opera e ataca, com intuito de criar métodos de detecção e prevenção do Mirai e seus clones. É apresentado vários tipos de ataques DDoS que o Mirai pode realizar como o ataque de inundação de UDP e VSE, inundação do resolvedor de DNS, ataque de DNS simples, inundação de pacotes SYN e ACK, inundação do TCP STOMP e inundação de HTTP, ressaltando que a versão original do *worm* possui métodos de ataque DDoS já conhecidos anteriormente, porém, com o lançamento de seu código fonte, surgiram diversos clones que implementam métodos de ataques mais perigosos. As capacidades de *worm* do Mirai em harmonia com os métodos de ataques DDoS perigosos de seus clones podem causar danos significativos à rede.

Para o processo de mitigação é proposto quatro soluções de acordo com os comportamentos e padrões comuns de *malware* que atacam dispositivos IoT, com o objetivo de uma detecção e prevenção sistêmica resistente para o futuro. Os autores também afirmam que descobrir um método simples de mitigação contra o Mirai ainda é um desafio devido ao mesmo utilizar as credenciais legítimas para obter acesso e os dispositivos IoT não são monitorados para detectar atividade maliciosa.

É mostrado quatro técnicas de proteção e prevenção, a primeira sendo a mudança de credenciais de serviços como SSH ou Telnet, vale ressaltar que é recomendado que a nova senha seja diferente da que é usada para fazer login no dispositivo. A segunda técnica é o fechamento de portas inutilizadas, as portas que são utilizadas pelo SSH e Telnet não devem ser deixadas publicamente abertos, a não ser que seja explicitamente aberto por um usuário avançado que tem conhecimento dos riscos que podem ocorrer. A terceira técnica é

implementar uma detectação para detectar qualquer atividade suspeita no *watchdog timer*[12] do kernel Linux, Mirai e suas variantes mandam requisições de ioctl para o *watchdog timer* com intuito de desabilitá-lo e nunca o reiniciar, permitindo o *worm* executar no sistema sem o risco de ser interrompido por uma reinicialização inesperada. Também seria possível compilar o kernel com a *flag* CONFIG_WATCHDOG_NOWAYOUT=1 para impedir que o *watchdog timer* seja desabilitado, porém esta opção possui algumas limitações como requerer que o kernel seja compilado, fazendo com que isto seja viável somente pelo fabricador do dispositivo ou por um usuário avançado. A quarta técnica é um *script* de proteção automatizado que pode ser usado para verificação e proteção de dispositivos IoTs em uma rede através de um mapeamento da rede para detectar dispositivos vulneráveis. Caso algum dispositivo seja detectado, tentativas de login semelhantes ao Mirai serão executadas. Se o *script* obtiver acesso ao dispositivo, será feito uma alteração das credenciais do dispositivo, a remoção do *malware* e relatar para o usuário sobre o que foi feito.

Por fim, os autores concluem que as técnicas sugeridas podem ser utilizadas pelos usuários e também pela indústria para detectar e parar o Mirai e sugere que os fabricantes de dispositivos IoT deveriam priorizar a segurança na produção desses dispositivos.

2.3.1.2 Analysis of the ILOVEYOU Worm (Bishop, 2000)

Este artigo possui o propósito de analisar o *worm* ILOVEYOU apresentado as características que um sistema deva ter para ser infectado e como o *worm* os infecta, discutindo sobre os perigos do *worm* e sugerindo técnicas de proteção dos sistemas.

Durante o desenvolvimento do texto, o autor aponta que o ILOVEYOU tem como alvo sistemas operacionais que executam Internet Explorer e o Microsoft Outlook, também que tenha a capacidade de interpretar *scripts* de Visual Basic, com isso o autor levanta três principais perguntas: (1) Como o ILOVEYOU é executado em sistemas que interpretam Visual Basic utilizando a biblioteca WScript[13]? (2) O quão difícil seria modificar o *worm* para ser executado em sistemas diferentes do Windows? (3) Quais são as formas de minimizar os danos?

Para responder a primeira pergunta, o autor descreve a anatomia do *wor*m, destacando que o mesmo é interpretado utilizando a *engine* da biblioteca WScript, que possui quatro rotinas

[12] Sistema de segurança embutido no kernel Linux, sendo composto pelo *timer* do hardware e o módulo de driver do kernel.

[13] Ambiente para executar *scripts* de Visual Basic e JavaScript no Windows.

principais sendo um que inicializa e chama pelos outros. O autor descreve cada uma das rotinas individualmente. Na segunda pergunta o autor aponta várias limitações no Mac OS que impossibilita do *worm* ser executado neste sistema como a falta de um conceito de registro, a preferência do usuário de escolher um navegador diferente do Internet Explorer, o *software* mIRC não suportar o sistema, o modo em que o Mac OS define os tipos de arquivo utilizando as propriedades dele mesmo ao invés da extensão, fazendo com que seja improvável de deletar os arquivos; E a impossibilidade do sistema executar binários do PE32 do Windows, tornando inviável a execução do binário WIN-BUGSFIX.exe (*Trojan* que o *worm* executa para roubar as senhas do usuário). Já nos sistemas baseados em UNIX, haveria as mesmas limitações do Mac OS em adicional de não ter um interpretador de Visual Basic, fazendo com que o *script* seja tratado como um arquivo de texto comum. Mesmo com a limitação mencionadas a respeito dos Mac OS e sistemas baseados em UNIX, o autor ressalta que seria viável criar um *worm* semelhante em outras linguagens. Por fim, para a terceira pergunta, o autor sugere utilizar a técnica de *strong typing* para que verificar caso o arquivo anexado no e-mail seja um executável, o Microsoft Outlook poderia inibir certas funções de serem acessadas pelo executável (como por exemplo, impedir que o executável tenha acesso a lista de contatos do Outlook) e *sandbox* para isolar o *worm* com propósito de impedir que o sistema seja afetado.

O autor conclui discorrendo as suas duas principais contribuições para o estudo do *worm* ILOVEYOU demonstrando que programas similares poderiam causar danos em sistemas diferentes do Windows e sugerindo a combinação das técnicas de *strong typing* e *sandbox* com o objetivo de proteger o sistema. Ademais, o autor também questiona se deve discutir a respeito de vulnerabilidades detalhadamente, já que com o código disponível abertamente possa fazer com que alguém lance uma versão modificada e que seja mais prejudicial, no entanto a descrição e estrutura do *worm* detalhados neste trabalho é capaz de ajudar os analistas a entender o *worm* mais facilmente e também a lidar com os possíveis clones e modificações.

2.3.1.3 Automated Behavioral Analysis of Malware: A Case Study of WannaCry Ransomware (Chen; Bridges, 2017)

Este artigo realiza um estudo de caso do WannaCry com o objetivo de propor um algoritmo para automatizar o processo de análise e identificação do *malware* em dispositivos infectados utilizando os *logs* do sistema e o *software* de análise de *malware* Cuckoo Sandbox, com a justificativa de agilizar o processo de análise manual para a identificação do *malware* e

gerar automaticamente uma análise comportamental do *malware* a partir de *logs* de um *sandbox*.

Para o processo de extração de atributos mais indicativos de *malware* nos *logs* da máquina, os autores realizaram quatro experimentos abordando os seguintes métodos: (1) obter um segundo conjunto de *logs* que são compostos por atividade não-maliciosas, (2) procurar características nos conjuntos de *logs* infectados que são incomumente comuns, (3) aplicar o cálculo de TF-IDF nos conjuntos de *logs* de atividades não-maliciosas e de atividades do WannaCry.

No primeiro experimento é analisado o executável do WannaCry e um *script* Python contendo as atividades normais do usuário no dispositivo pelo *software* Cuckoo Sandbox, as análises obtidas são usadas para calcular o TF-IDF de 74 funções de pré-encriptação específicas do WannaCry e por último é criado uma tabela de *ranking* de 43 ações importantes do *malware*, ressaltando que algumas dessas ações podem ou não ser vinda especificamente dos *logs* de atividade maliciosa do *malware*. O segundo experimento é calcular o TF-IDF de três cenários específicos dos *logs* para certificar se o *ranking* de ações não é influenciado pela variação do número de ações normais dos *logs* de atividades não-maliciosas. Já no terceiro experimento é criado o *script* de Python que executar atividades normais e logo em seguida acionar o *malware*, o *script* é enviado ao Cuckoo Sandbox para análise, o relatório da análise gerado é usado para calcular o TF-IDF, este experimento tem como intuito avaliar a acurácia do método para identificar ações específicas do *malware* quando uma boa parte das ações são provenientes de atividades não-maliciosas. Por último os arquivos submetidos à análise pelo Cuckoo Sandbox são enviados ao website VirusTotal que será verificado por 63 provedores de antivírus para buscar assinaturas detectando os arquivos em análise, o alvo deste experimento é para validar o método utilizados pelos autores com amostras polimórficas ao de fornecedores de antivírus.

Por fim, o trabalho conclui afirmando que os resultados obtidos durante o experimento, comprovam que o método utilizado é capaz de extrair características distintas do *malware* de *logs* contendo a maioria dos eventos não maliciosos e é resistente ao polimorfismo. Foi observado também que em ambientes com atividades de *ransomware*, a extração precisa das características foram automaticamente identificadas. Os autores afirmam que a utilização do cálculo de TF-IDF oferece melhores resultados para analisar o WannaCry comparado a algoritmos discriminatórios baseados em Análise Discriminante Linear de Fisher. Ademais, os autores apontam que trabalhos futuros devem integrar suas contribuições com outros sistemas

de detecção para geração de padrões automáticos ou melhorias em sistemas de autonomia de segurança de sistemas.

2.3.1.4 Worms and linux (Arora, 2013)

A motivação deste artigo é fazer uma visão geral a respeito de *worms* no sistema operacional Linux e realizar um pequeno estudo de caso sobre os dois *worms* mais conhecidos da época: Slapper e Lion, o autor introduz o conceito de *worms* falando tanto na visão de um atacante quanto a de um administrador. É mencionado no início do artigo um debate acerca de que o sistema Linux é frequentemente acreditado ser imune a *malwares* e *worms* por ser pouco usado nos desktops.

No artigo é identificado características que devem ser consideradas caso um ataque queira designar um *worm* de computador como: utilizar vulnerabilidades de sistemas para facilitar a infecção dos dispositivos, ser discreto para não ser detectado facilmente e possuir método de propagação rápido e eficiente para causar maior dano possível em vários dispositivos. Em contrapartida, é também mencionado acerca de métodos para detectar e conter *worms* como: monitorar processos confiáveis para detectar processos com um *worm* ou *malware* anexado, monitorar o escaneamento de IP para detectar se algum *worm* esteja escaneando o endereço de IP para tentar conectar e a implementação de armadilhas ou *honeypots* para alerta o administrador da máquina a bloquear o *worm* que esteja no *honeypot*.

Além disso, é feito uma pergunta sobre por que poucos *worms* obtiveram sucesso para atacar o sistema Linux, para responder a pergunta os autores discorrem as possíveis causas: (1) A maioria dos usuários do sistema Linux são usuários avançados; (2) Contas de usuários normais no Linux possui permissões limitadas; (3) O usuários não são encorajados a permanecer logado como administrador ou root, normalmente o usuário ganha privilégios de root por um tempo limitado quando necessário; (4) A maioria dos softwares utilizados no Linux são de código aberto; (5) Para executar um binário no Linux, é necessário setar permissões de execução no arquivo primeiro; (6) Linux é lançado como diferentes distribuições e com diferentes ferramentas e *softwares*, caso um *worm* seja escrito para infectar uma ferramenta, só conseguiria afetar as distribuições que tenham a ferramenta. Ademais, é falado sobre os *worms* Slapper e Lion, nesta revisão iremos focar no Lion.

O autor faz uma breve explicação de como o *worm* ataca, destacando que o *worm* cria *backdoors* no sistema para os invasores terem completo acesso, se instala no sistema infectado

como um *rootkit* para se esconder, se infiltra em comandos importantes como ls (comando que lista arquivos e diretórios) ou ps (comando que exibe os processos que estão em execução) e apagam certos arquivos de *log* do sistema para limpar seus rastros. É notado que o *worm* não pode ser removido do sistema, fazendo com que seja preciso formatar o dispositivo. O autor conclui chamando atenção para a necessidade de proteger o sistema Linux, mesmo que *malwares* tenham um baixo nível de sucesso no Linux, é dado sugestões para utilizar um antivírus e sistema de detecção de intrusos ou de prevenção de intrusos.

2.3.1.5 The Mirai Botnet and the IoT Zombie Armies (Kambourakis; Kolias; Stavrou, 2017)

Este artigo visa fornecer uma revisão sobre as causas do sucesso do Mirai e suas variantes, oferecendo uma análise compreensiva e abrangente do cenário do *malware*, e sugerir métodos de defesa contra o Mirai.

No texto é destacado que devido ao lançamento do código fonte do Mirai, consequentemente os clones e variantes do Mirai existentes foram de 213 mil para 493 mil nos dois primeiros meses. Algumas variantes utilizam parte da infraestrutura do *malware* original, tentam se defender de competidores fechando portas vulneráveis do dispositivo, encriptam a comunicação entre o *worm* e o servidor C&C e escaneiam diferentes portas TCP. Além disso, os autores apresentam aspectos básicos a respeito de *malwares* que têm como alvo dispositivos IoTs e faz uma comparação entre três *worms* semelhantes: LuaBot, BrickerBot e Hajime.

Os autores fazem uma análise detalhada sobre como o *worm* funciona e fazem uma comparação entre o *worm* semelhante, Hajime, destacando como os dois se infiltram, infectam e operam dentro do dispositivo alvo. No processo de infiltração: Hajime possui uma lista de credenciais registradas assim como o Mirai, porém ao invés de examinar a lista sequencialmente, ele busca aleatoriamente na lista; o Mirai executa um conjunto de comandos no Linux *shell* para verificar se o Busybox está instalado no sistema e se não é um *honeypot*; foram observados que os dois *worms* não tentam resetar a senha do dispositivo infectado. Já no processo de infecção, os dois *malwares* fazem um *fingerprint* no dispositivo de maneiras similares, Mirai utiliza os diretórios /proc/mounts e /proc/cpuinfo enquanto que o Hajime selecionando o seu diretório de trabalho o primeiro diretório que permita modificações. No processo de operação, Mirai escaneia para encontrar outros dispositivos vulneráveis pela rede conectando com o servidor C&C enquanto que o Hajime utiliza os protocolos BitTorrent DHT

e µTP[14] para fazer a comunicação entre os dispositivos, demonstrando que Hajime é mais resistente a ser derrubado por utilizar uma estratégia de rede descentralizada comparado ao Mirai; foi observado que o Hajime tem o comportamento similar aos outros *malwares* que servem para bloquear o Mirai, porém isso não faz com que o mesmo seja um defensor contra o Mirai e outros *worms*, pois instala um *backdoor* sem autorização no dispositivo infectado.

Baseado nas análises feitas, para detectar o mirai e suas variantes, os autores sugerem o monitoramento das portas TCP 23, 2323 e 22 durante o processo de infecção, pois onde ocorrem tentativas de ganhar acesso ao dispositivo, bloquear portas TCP usadas para sondar e que utilizam métodos *brute-force*, aplicar filtros de entrada e saída no roteador, parar e fechar portas e serviços desnecessários, em métodos drásticos, isolar a rede em que os dispositivos IoT estão conectados e permitir que o Busybox seja somente executado por um usuário específico.

O autor conclui que um grande número de dispositivos com propósito de gerenciamento remoto que utilizam sistema Linux e faltam atualizações de *firmware* são vulneráveis a novos *exploits* criados por autores maliciosos para criar *botnets* poderosos, capazes de no futuro poderem se inscrever no armazenamento persistente do dispositivo e ter um número maior de dispositivos IoT. Um dos maiores causadores deste problema são os fabricantes dos dispositivos lançarem os com funções de conexão remota ativadas, mas sem documentação. Os autores adicionam que o sistema Linux pode não ser a melhor opção para estes dispositivos, recomendado os sistemas como Google Brillo, ARM MbedOS e FreeRTOS.

2.3.1.6 The Dynamic Analysis of WannaCry Ransomware (Kao; Hsiao, 2018)

Esta pesquisa é uma análise dinâmica do WannaCry que visa explorar os indicadores de comportamento e danos do *malware*, utilizando o método de compartilhamento de informações de *malware* com a ajuda da plataforma YARA para encontrar famílias de *malwares* similares com mais eficiência. Os autores têm como objetivo gerar uma inteligência de ameaça eficaz através de indicadores de danos coletados para formular formações estruturadas.

Na elaboração do texto, os autores dão uma breve explicação sobre as plataformas de compartilhamento de *malwares* e a ferramenta YARA, destacando que o compartilhamento das assinaturas do WannaCry criados pelo YARA para uma plataforma de compartilhamento de

[14] Variante do protocolo BitTorrent baseado em UDP e criado com intuito de resolver problemas de controles de congestionamento do BitTorrent.

informações de *malwares* formam uma zona de defesa onde a junção das ferramentas possibilita à comunidade a ter respostas instantâneas contra ameaças similares futuras.

Durante o desenvolvimento, é descrita a linha do tempo com os padrões de comportamento do WannaCry, através de quatro perspectivas na análise dinâmica: processos, registro, sistema de arquivos e atividade na rede. Além disso, foi utilizado o software de gerenciamento de máquinas virtuais VMware Workstation Pro para isolar o *malware* do dispositivo real a fim de não causar danos, o sistema operacional Windows 7 x64 SP1. Vale ressaltar que os autores utilizaram uma das várias variantes do WannaCry criadas.

Ademais, é analisado comportamento e as modificações realizadas pelo WannaCry no sistema utilizando os softwares Process Explorer[15] para rastrear as modificações feitas nos processos; o Editor de Registro do Windows e Autoruns[16] para monitorar as alterações no registro; Process Monitor para observar arquivos manipulados e infectados; Process Explorer e Wireshark[17] para observar as mudanças de padrões na rede.

Por fim, os autores afirmam que sua análise contribuirá para a consolidação das plataformas de compartilhamento de informações de *malwares* por adicionar uma quantidade significativa de indicadores do WannaCry, consequentemente auxiliando no combate contra futuras ameaças.

[15] *Software* de gerenciamento de tarefas avançado e monitoramento do sistema, faz parte da suíte SysInternals, desenvolvido pela Microsoft para o Windows.

[16] *Software* para gerenciar programas que se executam automaticamente na inicialização do sistema, faz parte da suíte SysInternals, desenvolvido pela Microsoft para o Windows.

[17] *Software* de código aberto para análise de redes e pacotes, desenvolvido pela Wireshark Foundation e disponível para diversos sistemas.

2.3.2 Síntese dos Trabalhos

Quadro 4 - Síntese dos materiais analisados

Trabalhos/Artigo	Proposta	Conclusão
Margolis, J. *et al.*, 2017	Analisar os métodos do *botnet* Mirai, como ocorre a sua disseminação para novos dispositivos, os danos causados e propor estratégias de mitigação que podem ser usadas para prevenir ataques futuros.	As técnicas de proteção podem ser utilizadas pelos usuários e pela indústria para detectar e parar o Mirai e sugere que os fabricantes de dispositivos IoT devem priorizar a segurança na produção desses dispositivos.
Bishop, M., 2000	Examinar a estrutura e a organização do *worm* ILOVEYOU, bem como as características que um sistema deve possuir para ser infectado.	A combinação das técnicas *strong typing* e *sandboxing* podem proteger o sistema.
Chen, Qian; Bridges, Robert A., 2017	Desenvolver um método para identificar *ransomware* de forma automática em *logs* de sistema.	A utilização do cálculo de TF-IDF para a extração precisa das características do *malware*, oferece melhores resultados para analisar o WannaCry.
Arora, Himanshu., 2013	Apresentar os fatores que contribuem para a raridade de *worms* bem-sucedidos em sistemas Linux.	É necessária a proteção do sistema Linux, mesmo com *malwares* de baixo nível de sucesso.
Kambourakis, Georgios; Kolias, Constantinos; Stavrou, Angelos., 2017	Apresentar uma visão geral do estado atual dos *botnets* de IoT e o motivo do seu grande sucesso, com enfoque no Mirai e outros *worms* semelhantes.	A falta de atualização de *firmware* e de documentação em dispositivos de gerenciamento remoto com Linux, os torna vulneráveis a novos *exploits* criados por autores maliciosos para a criação de *botnets* poderosos.
Kao, Da-Yu; Hsiao, Shou-Ching., 2018	Produzir uma inteligência eficiente sobre ameaças cibernéticas e organizar os IOCs coletados em formações estruturadas.	Contribuição para plataformas de compartilhamento de informações de *malwares*.

Fonte: Autores (2023)

2.3.3 Comparação

Em 2000 os paradigmas de segurança e preocupações eram diferentes de atualmente, pois, os conceitos de segurança viriam a mudar muito com o tempo, principalmente porque as ameaças nunca pararam de evoluir, e com isso, podemos afirmar que um sistema operacional irá se adaptar com tempo, tanto, para atender melhor o consumidor, quanto para se defender das ameaças. A comparação entre os *malwares* ILOVEYOU, WannaCry, Lion e Mirai nos sistemas operacionais Windows e Linux, contribui para contrastar os diferentes paradigmas dentro de cada sistema operacional, as mudanças na segurança e o comportamento dos *worms*.

As diferenças entre as épocas apresentam as várias abordagens de ataque (*worms*) e defesa (sistemas operacionais).

Quadro 5 - Comparação dos *worms*

	ILOVEYOU	**WannaCry**	**Mirai**	**Lion**
Época	2000	2017	2016/2017	2000/2001
Sistemas	Windows 2000	Windows XP / 8.1	Linux	Linux
Falhas	Microsoft Outlook 2000	*EternalBlue* / *DoublePulsar*	Login & Senha Padrão / Escaneamento de Portas	*Buffer Overflow* no BIND
Reparos	Atualização de segurança lançada para Microsoft Outlook 2000	Atualizações de segurança lançadas para versões do Windows (XP ao 8.1)	Dinamização das senhas e login de roteadores na fabricação	Atualização lançada para o *software* BIND
Danos	Danos estimados de 10 bilhões de dólares	Danos monetários estimados de 4 bilhões de dólares. Vários serviços de grande importância foram afetados como bancos e hospitais	Diversos websites e serviços foram derrubados ao redor do mundo, afetando várias organizações e empresas	Desconhecido

Fonte: Autores (2023)

2.3.3.1 Comparação dos *worms* no Windows

Ambos os *worms* podem afetar o mesmo sistema operacional, mas dentre as épocas houve uma grande mudança nas versões e na forma como a segurança digital funciona dentro do Windows, desde aplicações *anti-malwares* até protocolos de rede mais restritos. Criando uma comparação entre os *worms* nas versões do sistema de cada época, é notável que não só o funcionamento da OS mudou, mas também a interação usuário e máquina, pois o *worm* ILOVEYOU utilizava-se dos usuários desatentos e engenharia social para invadir, e era executado em sistemas que interpretam Visual Basic utilizando a biblioteca WScript para executar várias tarefas que dificultavam a experiência do usuário, atualmente o WScript é considerado obsoleto pela Microsoft e será descontinuado nas futuras versões do Windows (Bishop, 2000; Microsoft, 2023). O WannaCry se aproveita de falhas vazadas dentro do próprio sistema operacional, se instalando como um serviço que será executado sempre que o computador for reiniciado e encriptando vários arquivos, tornando impossível acessá-los sem a chave de acesso que o criador do *malware* possui visando extorquir o usuário, que ao contrário do ILOVEYOU que criava várias tarefas maliciosas que apenas prejudicam a experiência do usuário mesmo que o criador do *worm* não tivesse nenhum retorno por elas (Adekotujo *et al.*, 2020; Akbanov *et al.*, 2019; Moreira *et al.*, 2017; Patten, 2017; Root, 2022). Isso expressa um paralelo curioso entre ambos os *worms*, demonstrando o quanto os sistemas operacionais se diferem mesmo sendo apenas versões diferentes e o quanto os *worms* evoluíram com o passar do tempo.

2.3.3.2 Comparação dos *worms* no Linux

O paralelo presente entre a forma de ação dos *worms* no Linux é que o sistema é empacotado como distribuições, apresentando variações entre as diferentes ferramentas apresentadas por cada distribuição. Isso, em grande parte das vezes, contribui para que os *malwares* escritos no Linux estejam limitados às distribuições que possuem uma versão vulnerável de certo software ou biblioteca como o Lion, que atacava distribuições que possuem versões vulneráveis da suíte de *software* BIND. Todavia, há casos em que *malwares* como o Mirai se utilizam de ferramentas que são bem comuns entre as distribuições, e com isso consegue infectar várias das distribuições Linux através desses protocolos e ferramentas (Adekotujo *et al.*, 2020; Kambourakis, Kolias, Stavrou, 2017).

2.3.3.3 Windows *vs.* Linux

O Windows e o Linux se diferem em suas arquiteturas de sistemas. É esperado que eles também apresentem comportamento diferentes quanto ao tratamento de ataques de *worms* e *malwares*. quando se trata dos OS, não seria novidade se eles também fossem diferentes em relação a tratamento de *malwares*. O kernel Linux possui código aberto para uso gratuito, enquanto o Windows possui código fechado para uso comercial. Possuindo código aberto e um rigoroso controle de permissões, o Linux se estende em uma variada gama de distribuições comerciais e gratuitas, definindo de forma independente as ferramentas utilizadas, conferindo variabilidade no funcionamento e estruturas das distribuições, apesar de utilizarem o mesmo kernel como base (Juell, 2017).

Mesmo que as diferentes versões do sistema Windows apresentam mudanças de umas para outras, é notório o esforço para manter arquivos, programas e formatos compatíveis entre as versões do sistema. Esta compatibilidade apresenta prós e contras, como no caso do *malware* WannaCry, que era compatível da versão Windows XP ao Windows 8.1 e possivelmente ao Windows 10 não atualizado. Isto gera um debate sobre problemas de segurança relacionados à compatibilidade entre os sistemas, pois um ataque pode ser bem-sucedido em diferentes versões do sistema operacional. Já no Linux, dificilmente a compatibilidade de *malware* é um problema por conta das diferentes distribuições, e que apenas ferramentas que são fundamentalmente necessárias podem acabar se repetindo nelas, tornando algo muito raro um *malware* ser capaz de infectar todas as distribuições Linux, claro que que há exceções como o Mirai, que contaminou mais de uma distribuição, porém, mesmo atingindo várias distribuições, ainda era limitado as plataformas que usavam ferramentas e bibliotecas específicas que eram necessárias para o funcionamento do *worm*, o que dificulta a infecção em outras máquinas, além de que, as próprias versões das bibliotecas também afetam a compatibilidade dos programas que podem acabar se tornando obsoletos, os *worm* e *malwares* também podem ser afetados por essas atualizações, ou seja, o Mirai teve diversas versões criadas por muitos autores para poder continuar infectando as novas versões das distribuições Linux (Bursztein, 2017). Há antivírus nas distribuições Linux, porém é pouco difundido entre os usuários, já que em teoria não haveria necessidade de utilizá-lo na crença comum, pois o “Linux não teria vírus” (Adekotujo *et al.*, 2020; Kavas; Feitelson, 2001; Arora, 2013).

2.4 Discussão

2.4.1 Mudanças de Paradigmas de Segurança no Windows

Como visto na comparação podemos concluir que os sistemas operacionais Windows 2000, Windows 8.1 mesmo sendo apenas versões diferentes do mesmo OS, expressam um paralelo curioso entre ambos os *worms* que os afetam, demonstrando o quanto eles se diferem e o quanto os *worms* evoluíram com o passar do tempo.

Mesmo se diferindo, não há como negar a evolução da segurança nas versões atuais do Windows, já que os *malwares* tiveram que se adaptar para serem mais discretos, como visto no WannaCry, que gerava um dano controlado para extorquir os usuários em troca de suas informações sequestradas em uma criptografia, enquanto o ILOVEYOU causava vários danos desnecessários durante a sua invasão, como corromper os arquivos, que serviria para alertar o sistema ou o usuário de sua presença. Mostrando que a tendência dos *malwares* é ser cada vez mais silencioso e especializado, para que o sistema operacional e o usuário não os detectem. (Patten, 2017). Com o tempo o Windows foi implementando soluções para detectar e combater malwares maliciosos que poderiam prejudicar tanto a experiência do usuário quanto criar um vazamento de dados, assim surgindo ferramentas como o Windows Firewall, Microsoft Defender, UAC, entre outros… demonstrando o quanto o combate aos *malwares* evoluiu. Mesmo com toda essa evolução, os sistemas mais antigos ainda continuam apresentando fragilidades por falta de atualização, mesmo que continuem tendo vários usuários o utilizando atualmente, assim dando oportunidades de invasão para *worms* como o WannaCry, que a acabou invadido até grandes empresas e instituições públicas, causando danos massivos (Akbanov *et al.*, 2019).

Como Microsoft manteve os arquivos, programas e formatos compatíveis nas suas variadas versões de sistemas, há um debate de segurança sobre se a compatibilidade entre os sistemas, pode ser mais prejudicial do que benéfico, já que a compatibilidade pode se estender para programas maliciosos, mesmo que sejam formulados vários programas e protocolos para a segurança dentro das versões, ainda há *worms* atuais que podem funcionar em versões anteriores do OS.

2.4.2 Mudanças de Paradigmas de Segurança no Linux

Sendo conhecido por suas vasta infinidade de distribuições, desktop, sistemas IoT, servidores e entres outros, é difícil estudar o sistema Linux como uma coisa só e seus *worms*, já que nas várias distribuições, e na escassez de artigos e materiais acadêmicos, há pouco sendo dito sobre o assunto, principalmente porque não é um sistema muito difundido no desktop principalmente, assim o tornando muito nebuloso para uma pesquisa e difícil a comparação entre os seus *worms* e os efeitos nas suas distribuições.

Como visto na comparação, a compatibilidade dos *malwares* no Linux são bem diferentes, gerando uma situação inusitada para se analisar, por isso é difícil ver um malware no Linux ganhar tanta fama como aconteceu no caso do Mirai, já que ele foi responsável pela invasão de vários IoT, tanto que as fabricantes tiveram que mudar como os sistemas de segurança IoT funcionam em padrão de fábrica. Os casos mais comuns são como o do Lion, que são criados para um propósito específico e para uma distribuição específica, tornando bem difícil encontrar estudos sobre esses casos, a correções sempre são em bibliotecas específicas, e normalmente, apenas a atualização da biblioteca para uma nova versão, acaba por inutilizar a ação do *worm*. Por esse motivo, da complexidade dos sistemas Linux, é bem difundido popularmente que não há vírus criados para o Linux, tornando muitos usuários tanto domésticos quanto corporativos, vulneráveis à ameaça.

2.5 Conclusão

Com esse trabalho, podemos observar como o ambiente de sistemas operacionais podem ser diversos e complexos. E com isso concluímos que os *worms* possuem diversas formas de interação com os sistemas operacionais e os usuários, sendo nocivos em grande parte. Diante dessa variabilidade e de *exploits* nas falhas da segurança, é sempre essencial estudar esses casos, para o aprimoramento da segurança digital, e também para prevenção de ameaças futuras, assim evitando grandes perdas que possam ocasionar a sociedade em seu todo.

2.5.1 Dificuldades Encontradas

Além da barreira de linguagem, pela falta de artigos ou materiais escritos em português, tendo grande parte escrito em língua estrangeira o que torna bem mais difícil o acesso e compreensão do tema. Também houve uma dificuldade em encontrar artigos e materiais que mencionam os *worms* no sistema Linux, esta escassez de trabalhos a respeito de *worms* e *malwares* no Linux deve-se ao fato de haver uma crença popular de que o "Linux não possui vírus" ou o "Linux é imune a vírus", como mencionado anteriormente no trabalho de Arora (2013):

> Algumas pessoas acreditam que Linux como um sistema operacional é inerentemente não vulnerável a vírus e worms; outros acreditam que Linux não é vulnerável simplesmente porque tem-se um uso limitado no desktop. O debate continua em aberto (Arora, 2013, tradução nossa).

2.5.2 Sugestões para Trabalhos Futuros

Para trabalhos futuros, objetiva-se que este artigo sirva como incentivo para o aprofundamento de pesquisas na área de segurança digital com enfoque na falta de segurança da utilização de sistemas antigos ou que perderam suporte em empresas e organizações, já que os mesmos são suscetíveis a ataques por vulnerabilidades devido à falta de atualização nos sistemas obsoletos. Outra temática que pode ser mais discutida em futuros trabalhos são os *malwares* em distribuição Linux, a fim de conscientizar sobre ameaças que podem ocorrem no sistema e culminar com a crença de que o Linux não possui vírus. Ademais, espera-se que estes estudos propostos possam agregar para o campo teórico da segurança digital.

REFERÊNCIAS BIBLIOGRÁFICAS

AB RAZAK, Mohd Faizal *et al*. **The rise of "malware": Bibliometric analysis of malware study**. Journal of Network and Computer Applications, 2016. Disponível em: https://www.sciencedirect.com/science/article/pii/S1084804516301904?via%3Dihub. Acesso em: 27 de fevereiro de 2023.

ADEKOTUJO, Akinlolu *et al*. **A Comparative Study of Operating Systems: Case of Windows, UNIX, Linux, Mac, Android and iOS**. International Journal of Computer Applications, 2020. Disponível em: https://www.researchgate.net/profile/Adedoyin-Odumabo/publication/343013056_A_Comparative_Study_of_Operating_Systems_Case_of_Windows_UNIX_Linux_Mac_Android_and_iOS/links/61f2b50a9a753545e2fe8300/A-Comparative-Study-of-Operating-Systems-Case-of-Windows-UNIX-Linux-Mac-Android-and-iOS.pdf . Acesso em: 28 de fevereiro de 2023.

AKBANOV, Maxat *et al*. **WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms**. Journal of Telecommunications and Information Technology, 2019. Disponível em: https://bibliotekanauki.pl/articles/309353.pdf. Acesso em: 24 de fevereiro de 2023.

ANTONAKAKIS, Manos *et al*. **Understanding the mirai botnet**. Usenix. 2017. Disponível em: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf. Acesso em: 19 de março de 2023.

ARORA, Himanshu. **Worms and Linux**. Linux Journal, 2013. Disponível em: https://dl.acm.org/doi/fullHtml/10.5555/2502864.2502868. Acesso em: 19 de março de 2023.

AUDELO GONZALEZ, Jesús; SOLIS, Antonio Castañeda; GUEVARA LOPEZ, Pedro. **Gusanos Informáticos: de los inicios a su primer impacto en los gobiernos**. Compreendemos, 2013. Disponível em: https://www.comprendamos.org/alephzero/64/gusanosinformti.html. Acesso em: 22 de maio de 2023.

AYCOCK, John. **Computer viruses and malware**. Springer Science & Business Media, 2006.

BECK, Michael et al. **Linux Kernel Internals**. Addison-Wesley, 1996.

BISHOP, Matt. **Analysis of the ILOVEYOU Worm**. Internet: http://nob. cs. ucdavis. edu/classes/ecs155-2005-04/handouts/iloveyou.pdf, 2000. Disponível em:

http://wwwusers.di.uniroma1.it/~parisi/Risorse/10.1.1.87.8077.pdf. Acesso em: 23 de fevereiro de 2023.

BURSZTEIN, Elie. **Inside the infamous Mirai IoT Botnet: A Retrospective Analysis**, Cloudflare, 2017. Disponível em: https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/#toc-2. Acesso em 10 de junho de 2023.

CARPENTER, Tom. **Microsoft Windows Operating System Essentials**. John Wiley & Sons, 2011.

CERT.BR. **Incidentes Reportados ao CERT.br -- Janeiro a Dezembro de 2020**, 2022. Disponível em: https://www.cert.br/stats/incidentes/2020-jan-dec/tipos-ataque.html. Acesso em: 20 de fevereiro de 2023.

CHEN, Qian; BRIDGES, Robert A. **Automated Behavioral Analysis of Malware: A Case Study of WannaCry Ransomware**. IEEE, 2017.

CUSTER, Helen. **Inside Windows NT**. Microsoft Press, 1993.

DUCKLIN, Paul. **ILOVEYOU: The Love Bug virus 20 years on – could it happen again?**. Naked Security, 2020. Disponível em: https://nakedsecurity.sophos.com/2020/05/04/iloveyou-the-love-bug-virus-20-years-on-could-it-happen-again/. Acesso em: 24 de maio de 2023.

EMBEDDED. **2019 Embedded Markets Study**, 2019. Disponível em: https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf. Acesso em: 27 de fevereiro de 2023.

ERBSCHLOE, Michael. **Trojans, Worms, and Spyware: A computer security professional's guide to malicious code**. Butterworth-Heinemann, 2005.

GHOSH, Agamoni. **WannaCry: List of major companies and networks hit by ransomware around the globe**. International Business Times, 2017. Disponível em: www.ibtimes.co.uk/wannacry-list-majorcompanies-networks-hit-by-deadlyransomware-around-globe-1621587. Acesso em: 29 de fevereiro de 2023.

GOOGLE. **Google Keynote (Google I/O ‘21) - American Sign Language**. Youtube, 2021. 1 vídeo (115 min). Publicado pelo canal Google. Disponível em: https://www.youtube.com/watch?v=Mlk888FiI8A. Acesso em: 12 de dezembro de 2023.

GUIMARÃES, Antonio Teodoro Ribeiro. **Linux versus Microsoft: as novas tendências no mercado de sistemas operacionais**. Transinformação, 2005. Disponível em: https://www.scielo.br/j/tinf/a/N98kTdvWNJnG6SqGTPdStcP/abstract/?lang=pt. Acesso em: 15 de março de 2023.

GUEVARA LÓPEZ, Pedro. *et al*. **Basic definitions for discrete modeling of computer worms epidemics**. Ingeniería e Investigación, 2015. Disponível em: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-56092015000100014&lang=pt. Acesso em: 22 de maio de 2023.

HOLUB, Artsiom. **The Future is Here – Assaulting the Internet with Mirai**. Cisco Umbrella, 2020. Disponível em: https://umbrella.cisco.com/blog/future-assaulting-internet-mirai. Acesso em: 10 de julho de 2023.

JARAMILLO, Luis Eduardo Suástegui. **Malware detection and mitigation techniques: Lessons learned from Mirai DDOS attack**. Journal of Information Systems Engineering & Management, 2018. Disponível em: https://www.jisem-journal.com/download/malware-detection-and-mitigation-techniques-lessons-learned-from-mirai-ddos-attack.pdf. Acesso em: 19 de março de 2023.

JUELL, Kathleen. **A Brief History of Linux**, DigitalOcean, 2017. Disponível em: https://www.digitalocean.com/community/tutorials/brief-history-of-linux. Acesso em: 28 de fevereiro de 2023.

KAMBOURAKIS, Georgios; KOLIAS, Constantinos; STAVROU, Angelos. **The mirai botnet and the iot zombie armies**. IEEE, 2017.

KAO, Da-Yu; HSIAO, Shou-Ching. **The dynamic analysis of WannaCry ransomware**. In: 2018 20th International conference on advanced communication technology (ICACT). IEEE, 2018. Disponível em: https://icact.org/upload/2018/0411/20180411_finalpaper.pdf. Acesso em: 7 de agosto de 2023.

KASPERSKY. **What's the Difference between a Virus and a Worm?**, 2014?. Disponível em: https://www.kaspersky.com/resource-center/threats/computer-viruses-vs-worms. Acesso em: 22 de maio de 2023.

KAVAS, Avi; FEITELSON, Dror G. **Comparing Windows NT, Linux, and QNX as the basis for cluster systems. Concurrency and Computation: Practice and Experience**, 2001. Disponível em:

https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.613?casa_token=Mk1ptmWvl28AAAAA:zw0zoj7MWTLxlnLB31nL_DSI2j2zUcDGWtteqJY4dSjpaEKdhh7W3So6cafqrGhNSUcDtXblrMSdUtFE. Acesso em: 11 de março de 2023.

KIM, Jin-Young; BU, Seok-Jun; CHO, Sung-Bae. **Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders**. Information Sciences, 2018. Disponível em: https://www.sciencedirect.com/science/article/pii/S0020025518303475. Acesso em: 27 de fevereiro de 2023.

MARGOLIS, Joel *et al*. **An in-depth analysis of the mirai botnet**. IEEE, 2017.

MALWAREBYTES. **What was WannaCry? | WannaCry Ransomware**, [s.d.]. Disponível em: https://www.malwarebytes.com/wannacry. Acesso em: 10 de maio de 2023.

MICROSOFT. **A history of Windows**, 2015. Disponível em: https://web.archive.org/web/20160611182917/http://windows.microsoft.com/en-in/windows/history#T1=era0. Acesso em: 8 de março de 2023.

MICROSOFT. **Worms**, 2023. Disponível em: https://learn.microsoft.com/en-us/microsoft-365/security/intelligence/worms-malware?view=o365-worldwide. Acesso em: 22 de maio de 2023.

MICROSOFT. **Resources for deprecated features**, 2023. Disponível em: https://learn.microsoft.com/en-us/windows/whats-new/deprecated-features-resources. Acesso em: 31 de Outubro de 2023.

MOREIRA, Guilherme Baesso *et al*. **A era dos crypto ransomwares: um estudo de caso sobre o wannacry**. Anais do XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. SBC, 2017. Disponível em: https://sol.sbc.org.br/index.php/sbseg/article/view/19525. Acesso em: 29 de fevereiro de 2023.

PATTEN, David. **The evolution to fileless malware**. Retrieved from, 2017. Disponível em: https://infosecwriters.com/Papers/DPatten_Fileless.pdf. Acesso em: 29 de março de 2023.

ROOT, Enoch. **ILOVEYOU: the virus that loved everyone**. Kaspersky, 2022. Disponível em: https://www.kaspersky.com/blog/cybersecurity-history-iloveyou/45001/. Acesso em: 22 de maio de 2023.

SHAPIRO, Scott J. **The Strange Story of the Teens Behind the Mirai Botnet**. IEEE Spectrum, 2023. Disponível em: https://spectrum.ieee.org/mirai-botnet. Acesso em: 10 de julho de 2023.

STALLINGS, Willian. **Operating Systems: Internals and Design Principles**. Pearson Education Limited, 2018.

STATCOUNTER. **Desktop Operating System Market Share Worldwide**, 2023. Disponível em: https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202204-202204-bar. Acesso em: 27 de fevereiro de 2023.

TAHIR, Rabia. **A study on malware and malware detection techniques**. International Journal of Education and Management Engineering, 2018. Disponível em: https://www.mecs-press.org/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf. Acesso em: 20 de fevereiro de 2023.

TANENBAUM, Andrew S.; BOS, Herbert. **Modern Operating Systems**. Pearson, 2015.

TSING, Willian. **The Advanced Persistent Threat files: Lazarus Group**. Malwarebytes, 2018. Disponível em: https://www.malwarebytes.com/blog/news/2019/03/the-advanced-persistent-threat-files-lazarus-group. Acesso em: 14 maio de 2023.

WELIVESECURITY. **ILOVEYOU: The wrong kind of LoveLetter**, 2017. Disponível em: https://www.welivesecurity.com/2017/02/14/iloveyou-wrong-kind-loveletter/. Acesso em: 24 de maio de 2023.

W3TECHS. **Usage statistics of operating systems for websites**, 2023. Disponível em: https://w3techs.com/technologies/overview/operating_system. Acesso em: 27 de fevereiro de 2023.

YOSHIMURA, Takeshi. **A study on faults and error propagation in the Linux operating system(本文)**. IRDB, 2016. Disponível em: https://irdb.nii.ac.jp/en/01276/0000614419. Acesso em: 15 de março de 2023.

YOSIFOVICH, Pavel *et al*. **Windows Internals - Part 1, 7th Edition**. Microsoft Press, 2017.

GLOSSÁRIO

API	Mecanismos que permitem a comunicação entre dois softwares utilizando um conjunto de definições.
BitTorrent	Protocolo de transferência de arquivos descentralizado em que os usuários mandam e recebem porções do arquivo.
Botnet	Rede de computadores infectados por um malware que pode ser controlado remotamente pelo invasor.
Buffer Overflow	Falha de software onde um programa em execução tenta armazenar dados além do permitido, causando um sobrecarregamento no sistema.
C&C	Computador utilizado por um cibercriminoso para mandar comandos e receber dados de dispositivos comprometidos.
Desktop	Computador designado para uso doméstico ou pessoal.
DDoS	Ataque malicioso em que tenta impedir o funcionamento de um website ou serviço de rede sobrecarregando-o com pacotes maliciosos.
DHT	Sistema de armazenamento descentralizado que providencia um serviço de pesquisa similar a tabela *hash*.
DNS	Sistema de gestão de nomes que traduzem nomes de domínios em endereços IP.
Exploit	Software ou código malicioso que abusam de vulnerabilidades de segurança para atacar o sistema.
Flag	Valor que pode indicar um sinal de uma função ou processo, podendo indicar se algo está habilitado ou não.
Fingerprint	Conjunto de informações coletadas a respeito de um dispositivo ou sistema com intuito de identificação.
Firewall	Dispositivo de segurança de redes que possui a função de controlar o tráfego de entrada e saída, decidindo o que permitir ou bloquear.
FTP	Protocolo de transmissão de arquivos entre computadores pela rede TCP/IP.
GUI	Forma de interface de usuário em que o usuário interage com um dispositivo por elementos gráficos e sonoros.
Honeypot	Ambiente de rede controlado e seguro com intuito de ser uma isca para ataques cibernéticos.
IOC	Informações que apontam que uma intrusão ou violação de segurança ocorrido em um dispositivo.

IoT	Dispositivos que possuem sensores e *softwares* para se conectar com outros dispositivos próximos pela rede.
IPC	Mecanismo de comunicação providenciado pelo sistema operacional para gerenciar dados entre processos.
Kernel	Núcleo de um sistema operacional, tendo a diversas funções garantir o bom funcionamento do sistema.
Log	Registros de eventos relevantes do sistema operacional.
NTFS	Sistema de arquivos padrão do Windows que possui função de *journaling*, foi inspirado no HPFS do OS/2 e lançado junto com Windows NT 3.1 em 1993.
P2P	Arquitetura de rede descentralizado onde cada participante funciona como cliente e servidor.
POSIX	Conjunto de padrões de interfaces de sistemas operacionais criados pela IEEE com o propósito de manter a compatibilidade entre sistemas.
RAS	Serviço de rede do Windows que providencia acesso remoto para outros computadores da rede.
Rootkit	Software malicioso discreto que possui acesso privilegiado a várias áreas do computador.
Sandbox	Ambiente isolado e controlado para execução de programas maliciosos ou suspeitos sem afetar o sistema.
Shell	Interface do sistema que serve para os usuários e programas acessarem serviços do próprio sistema.
SMB	Protocolo de compartilhamento de arquivos entre computadores criado pela Microsoft para o sistema Windows.
SSH	Protocolo de rede para comunicação remota entre dois dispositivos com segurança utilizando criptografia.
Telnet	Protocolo de rede para comunicação remota, atualmente considerado obsoleto por ser bastante inseguro.
TCP	Protocolo de rede padrão que permite a comunicação entre diferentes dispositivos pela rede.
TF-IDF	Medida estatística para avaliar o quão relevante uma palavra é para um documento em uma coleção de documentos.
Trojan	Software malicioso que se disfarça, aparentando ser legítimo e seguro para enganar o usuário.

UAC	Recurso de controle de acesso de usuário do Windows com o intuito de limitar certos *softwares* de ser executado com permissões de administrador, foi lançado em 2007 no Windows Vista e Server 2008.
UDP	Protocolo de comunicação utilizado para estabelecer conexões de baixa latência e baixa tolerância entre software na rede.
White Hat	Também chamado de *hackers* éticos, são *hackers* que só fazem teste de intrusão com permissão para procurar vulnerabilidades.

APÊNDICE

FONTES DOS *MALWARES* UTILIZADOS NA ELABORAÇÃO DAS FIGURAS 3 E 4

ILOVEYOU:

https://pastebin.com/EeU9kp5P

WannaCry:

https://github.com/ytisf/theZoo/tree/master/malware/Binaries/Ransomware.WannaCry