Capítulo 7

Segurança Cibernética 2030: Experiências, Desafios e Oportunidades

Alberto Egon Schaeffer-Filho, Jéferson Campos Nobre, Juliano Araújo Wickboldt, Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary, Weverton Luis da Costa Cordeiro

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

Cybersecurity has assumed an increasingly critical role as a fundamental pillar of a digital society, deeply interconnected and increasingly dependent on services provisioned via consolidated (e.g., 4G/5G) and emerging (such as artificial intelligence) technologies and concepts. In this context, expectations have increased that computing can contribute to solving emerging challenges in cybersecurity, especially those challenges intrinsically influenced by the particularities of Brazilian society. In this sense, there is great expectation about how computational solutions can support cybersecurity professionals and researchers in solving the challenges that plague our digital society, such as fake news, cyber scams, identity theft, data theft, privacy violations, etc. This chapter will address the cybersecurity research landscape, highlighting the opportunities and challenges that are relevant for the next decade: People-centric security, Artificial intelligence and security and Security in the era of programmable networks.

Resumo

A cibersegurança tem assumido um papel cada vez mais crítico como pilar fundamental de uma sociedade digital, profundamente interconectada e cada vez mais dependente de serviços provisionados via tecnologias e conceitos consolidados (por ex., 4G/5G) e emergentes (como inteligência artificial). Neste contexto, aumentaram as expectativas de que a computação possa contribuir na solução dos desafios emergentes em segurança cibernética, em especial àqueles desafios intrinsecamente influenciados pelas

Vídeo com a apresentação do capítulo: `https://youtu.be/uEyOldWYauI`

particularidades da sociedade brasileira. Nesse sentido, há grande expectativa sobre como as soluções computacionais poderão apoiar profissionais e pesquisadores em cibersegurança a resolverem os desafios que afligem nossa sociedade digital, tais como fake news, golpes cibernéticos, usurpação de identidade, roubo de dados, violações de privacidade, etc. O presente capítulo abordará o panorama de pesquisa em cibersegurança, destacando as oportunidades e desafios que se impõem como relevantes para a próxima década: Segurança Centrada nas Pessoas, Inteligência artificial e segurança e Segurança na era de redes programáveis.

7.1. Introdução

A importância da cibersegurança tem crescido significativamente à medida que nossa sociedade se torna cada vez mais interligada e dependente de tecnologias estabelecidas e emergentes. Evidentemente, tal crescimento traz um grande número de oportunidades assim como de riscos. O ciberespaço constitui um cenário promissor pela prática de toda sorte de ações ilícicas, as quais não respeitam fronteiras geopolíticas tradicionais. Dessa forma, ataques cibernéticos exploram as vulnerabilidades das estruturas de Tecnologia da Informação e Comunicação. Como exemplos desses ataques, podem ser citados fake news, golpes cibernéticos, usurpação de identidade, roubo de dados, violações de privacidade, etc.

Os desafios em cibersegurança oferecem uma oportunidade para "repensar"o papel da computação na própria sociedade. Nesse sentido, há grande expectativa sobre como a computação como área desempenhará a abordagem dos desafios emergentes em cibersegurança, produzindo soluções que poderão apoiar profissionais e pesquisadores em cibersegurança. Mesmo sendo na sua maioria globais, é necessária uma atenção especial em aspectos são influenciados diretamente por características específicas da sociedade brasileira.

Considerando o panorama de pesquisa em cibersegurança, alguns pontos podem ser destacados. Primeiro, a emergência da Segurança Centrada nas Pessoas. Assim, vislumbra-se a pesquisa e o projeto de mecanismos de segurança que consideram de forma central aspectos humanos e sociais. Segundo, as relações entre Inteligência Artificial e Cibersegurança. Neste contexto, a IA impacta substancialmente a cibersegurança, tanto de forma positiva quanto negativa. Finalmente, a segurança na era de redes programáveis. Os avanços recentes na internet ampliaram nossa capacidade de modificá-la, sendo necessária a verificação e garantia de propriedades de segurança. Os pontos destacados merecem ser discutidos a fim de se buscar uma compreensão da evolução da cibersegurança nos próximos anos.

O presente capítulo está organizado da seguinte forma. No Capítulo 7.2, será discutida a Segurança Centrada nas Pessoas. No Capítulo 7.4, serão apresentados aspectos sobre a relação entre Inteligência Artificial e Cibersegurança. No Capítulo 7.3, a Segurança na era de redes programáveis será abordada. Finalmente, as oportunidades e desafios que se impõem como relevantes para a próxima década na cibersegurança são discutidos no Capítulo 7.5.

7.2. Segurança Centrada nas Pessoas

A Segurança Centrada nas Pessoas é a cibersegurança projetada com as pessoas em mente. Tradicionalmente, os mecanismos e controles de segurança costumam ser projetados sob suposições ingênuas a respeito dos humanos: que os mesmos sempre agem de forma lógica, racional e fazendo o melhor para si. Infelizmente, isso não é o caso em uma grande quantidade de eventos. Em ambientes organizacionais, as principais motivações de colaboradores são metas, necessidades dos clientes, etc. Assim, se houver medidas de cibersegurança que se oponham em relação a tais motivações, soluções alternativas serão buscadas. Dessa forma, é necessário que a cibersegurança seja desenvolvida considerando que sua abordagem seja funcional e adaptada às pessoas, e não o contrário.

O desenvolvimento de mecanismos de cibersegurança centrados nas pessoas vislumbra a pesquisa e o projeto de técnicas de segurança personalizadas, considerando aspectos humanos e sociais. Isto é necessário porque se a cibersegurança não está implicada para as pessoas, muito riscos estarão associados. Por exemplo, fatores que afetam o comportamento humano e aumentam a suscetibilidade das pessoas à manipulação precisam ser considerados na produção de sistemas computacionais. Excluindo-se tais fatores, a possibilidade de explorar o comportamento humano para obtenção de dados e informações relevantes de potenciais alvos é aumentada, além facilitar a realização de ações pelos colaboradores que colocam as organizações em risco.

Políticas de segurança da informação frequentemente são elaboradas sem um entendimento de como as pessoas realmente trabalham. As pessoas moldam a cibersegurança ao criar soluções alternativas, porque seguir a política muitas vezes dificulta a realização de suas tarefas laborais. Na prática, o fator humano é o elo mais fraco na cadeia de cibersegurança [Mitnick and Simon 2003]. No entanto, a Segurança Centrada nas Pessoas busca inverter esta lógica e colocar o humano como centro dos processos de cibersegurança. Tal inversão frequentemente promove a conscientização e o envolvimento das pessoas de forma colaborativa.

A Segurança Centrada nas Pessoas ajuda os profissionais de segurança a projetarem sistemas e políticas que funcionem considerando as características dos humanos. Como as pessoas não são normalmente motivadas pela segurança, são necessárias estratégias de convencimento (e.g., campanhas de conscientização), além de políticas de segurança da informação claras. Assim, é necessário que os profissionais de cibersegurança estejam dispostos a produzir essas políticas com ênfase nas pessoas por design.

A presente seção está organizada da seguinte forma. Inicialmente, serão discutidos os ataques focados nas pessoas, especialmente no que tange à Engenharia Social. Em seguida, aspectos relacionados com a automatização da engenharia social e os Grande Modelos de Linguagem serão apresentados. Finamente, estratégias para prevenção e mitigação de ataques focados nas pessoas serão comentadas.

7.2.1. Ataques Focados nas Pessoas: uma Introdução à Engenharia Social

Engenharia Social é caracterizada como a prática de aproveitar aspectos humanos com o intuito de obter acesso a dados e informações de possíveis alvos em sistemas de informação, independentemente do uso de tecnologia. Trata-se de uma abordagem de ataque

que se baseia na exploração do comportamento humano, utilizando persuasão e manipulação psicológica. Os ataques de Engenharia Social colocam o atacante em uma posição favorecida no fluxo de informações, tirando proveito de uma relação de confiança. O desenvolvimento de uma relação de confiança faz uso da manipulação psicológica induzindo as pessoas realizarem ações específicas.

O processo de proteção de dados e informações, visando garantir sua confidencialidade, integridade e disponibilidade, está intrinsecamente relacionado à Cibersegurança. A relação entre a Engenharia Social e a Cibersegurança é ainda mais reforçada pelo contínuo desenvolvimento tecnológico, que tem possibilitado a automação e escalabilidade dos ataques de Engenharia Social, tornando-os desafios cada vez mais difíceis de combater [Beal 2005]. Em situações reais, é importante reconhecer que o elemento humano frequentemente representa o ponto mais vulnerável na cadeia de segurança cibernética. [Mitnick and Simon 2003] [Klimburg-Witjes and Wentland 2021].

A ampla disponibilidade de diversos meios de comunicação de grande alcance cria um ambiente propício para os ataques de Engenharia Social. O avanço da tecnologia tem facilitado a automação e escalabilidade desses ataques, permitindo que os invasores alcancem um grande número de possíveis vítimas em um curto espaço de tempo [Pinheiro 2020]. Portanto, compreender e se proteger contra os ataques de Engenharia Social tornase essencial para garantir a segurança de dados e sistemas em ambientes conectados e dependentes da tecnologia.

Os engenheiros sociais podem utilizar a automação para desenvolver ferramentas pré-programadas para realizar tarefas sem a intervenção humana, possibilitando a escalabilidade dos ataques. Tais técnicas podem, por exemplo, considerar o uso de chatbots, tanto como ferramenta para ataques, como para auxílio dos profissionais. Além disso, devem ser considerados aspectos da interação das ferramentas e processos de segurança com o comportamento humano.

Os ataques automatizados podem ser preparados utilizando informações coletadas ou através da influência sobre indivíduos nas redes sociais. Essas redes representam um espaço virtual que pode ser usado para os atacantes explorarem vulnerabilidades técnicas e a falta de conhecimento e conscientização dos usuários sobre ações de Engenharia Social. Por exemplo, uma das vulnerabilidades que são encontradas em redes sociais é a criação de perfis falsos, os quais constituem um percentual significativo dos usuários dessas redes.

7.2.2. Automatização como uma Evolução da Engenharia Social

A crescente conectividade e automação revolucionaram as infraestruturas econômicas e culturais do mundo, ao mesmo tempo em que introduziram riscos em termos de ataques cibernéticos. A Engenharia Social Automatizada representa uma abordagem que combina técnicas de Engenharia Social com a automação, usando ferramentas e scripts para criar ataques eficazes em grande escala. Os ataques de Engenharia Social tradicionais requerem investimento de tempo e recursos para estabelecer uma relação de confiança entre o atacante e o usuário. Portanto, ao automatizar os aspectos repetitivos e monótonos desse processo, os agressores aproveitam para realizar ataques em larga escala de forma mais eficiente [Guzman and Lewis 2020].

A comunicação humana tem sido a base para o desenvolvimento de interfaces homem-máquina. Neste contexto, as redes sociais facilitam a comunicação, a interação social e o compartilhamento de informações pessoais e informações corporativas, aumentando sua popularidade no ambiente cibernético. As conexões formadas nesses ambientes virtuais de socialização permitem um grande troca de informações, reforçando o papel das redes como estruturas comunicativas para as relações sociais [Castells 2002].

As redes representam um espaço virtual atraente para os invasores explorarem vulnerabilidades técnicas. idades e falta de conhecimento e conscientização dos usuários sobre as ações de Engenharia Social [Al-Charchafchi et al. 2019]. O crescimento das redes sociais tem possibilitado a criação de um grande número de perfis falsos, com o uso de bots automatizados para suportar e dimensionar as atividades maliciosas.

Atacantes têm empregado bots para automatizar as etapas necessárias para estabelecer uma conexão de confiança com os usuários [Shafahi et al. 2016]. Essa construção de confiança envolve técnicas de manipulação psicológica, incentivando os usuários a interagir com ferramentas usadas pelos engenheiros sociais no ambiente digital. A combinação de táticas de manipulação psicológica com tecnologia avançada permite que os atacantes alcancem múltiplos alvos com surpreendente eficácia.

Bots são capazes de simular conversas humanas, sendo conhecidos como "ChatBots", e quando atuam nas redes sociais, são chamados de "SocialBots"[Shafahi et al. 2016]. Os ataques de Engenharia Social Automatizada requerem intervenção humana mínima, como um robô automatizado personificando outro humano para es-estabelecer uma conexão com as vítimas e pode atingir vários alvos simultaneamente devido à sua capacidade de escalabilidade [Mitnick and Simon 2003] [Huber et al. 2009].

Ataque de Engenharia Social Automatizada usando recursos como SocialBots e phishing são cada vez mais comuns, aproveitando o uso crescimento para atividades pessoais e profissionais. Os ataques de Engenharia Social requerem tempo e recursos para estabelecer uma relação de confiança. Os ataques de ES demandam tempo e recursos para estabelecer um relacionamento de confiança. No entanto, o desenvolvimento de uma interface homem-máquina permite que tais relacionamentos sejam automatizados.

7.2.3. Engenharia Social Automatizada e os Grande Modelos de Linguagem

O Processamento de Linguagem Natural (*Natural Language Processing* - NLP) tem recebido recentemente ampla atenção na cibersegurança, particularmente na automação cibernética. NLP é uma área da ciência da computação que permite que computadores interajam com a linguagem humana por meio do uso de software específico. Grandes Modelos de Linguagem (*Large Language Models* - LLMs) tornaram-se amplamente utilizados em aplicativos de NLP (e.g., ChatGPT e Google BERT), incluindo chatbots e assistentes virtuais. No entanto, com a utilização crescente destes modelos surge a necessidade de garantir a privacidade dos dados e a conformidade da segurança, especialmente quando estão envolvidas informações sensíveis.

Os usuários devem ter cautela ao enviar informações pessoais para o aplicações que utilizam LLMs, já que esses modelos podem ser treinados com dados que contêm informações sensíveis. Ao enviar uma pergunta, os usuários devem evitar incluir qualquer

informação que possa ser usada para identificá-los ou a outra pessoa, como, por exemplo, nomes, endereços, endereços de e-mail, etc.

Ferramentas genéricas de NLP não funcionam bem com linguagem específica de domínio, pois cada domínio possui características únicas que uma ferramenta genérica não está treinada para lidar. O domínio da cibersegurança apresenta uma variedade de dificuldades únicas, como a necessidade de compreender termos técnicos em constante evolução. Neste contexto, modelos de linguagem de cibersegurança têm sido criados, sendo os mesmos capaz de capturar conotações de texto em textos relacionados à cibersegurança. Um exemplo de tais modelos é o SecureBERT [Liberato 2022].

7.2.4. Prevenção e Mitigação para Ataques Focados nas Pessoas

Os atacantes investem em focar em pessoas dentro de uma organização e seus relacionamentos, a fim de lançar ataques que perpassam os mecanismos de cibersegurança tradicional. Tais atacantes exploram relacionamentos de confiança entre usuários internos e externos. Dessa forma, são necessárias estratégias para prevenção e mitigação de ataques focados nas pessoas. infelizmente, tais estratégias frequentemente falham em capturar o interesse das pessoas e são percebidas como uma tarefa secundária, um obstáculo ou uma distração de suas responsabilidades principais.

O treinamento de conscientização em cibersegurança pode incluir simulações de ataques, fornecendo aos usuários a oportunidade de vivenciar situações reais e aprender a identificar os sinais de manipulação. Ao aumentar a conscientização os usuários passam a ser defensores dos ativos de informação contra a ataques focados nas pessoas (e.g., Engenharia Social Automatizada), reduzindo o impacto desses ataques e fortalecendo a segurança dos sistemas de informações. Uma solução que tem se mostrado eficaz reside na implementação da gamificação, oferecendo uma alternativa envolvente e interativa às sessões de treinamento obrigatórias [Nijland 2022].

A comunicação e a consistência são fundamentais para facilitar uma cultura de segurança positiva[1]. Não apenas quando um incidente ocorreu, mas em qualquer situação em que seja necessário entender exatamente o que está acontecendo. Tal cultura dá aos usuários a confiança de que não apenas podem falar abertamente, mas que quaisquer ações ou decisões serão avaliadas de maneira justa. Isso faz com que se aumente o engajamento nos processos de cibersegurança, permitindo que os usuários concentrem no que é melhor para a organização, em vez de se preocuparem em se proteger.

7.3. Segurança na Era de Redes Programáveis

Os avanços recentes em Redes Definidas por Software (*Software Defined Networking*, SDN) expandiram nossa capacidade de programar a rede em direção ao plano de dados. Através de linguagens específicas de domínio como o P4, os operadores de rede podem rapidamente implementar novos protocolos em dispositivos de encaminhamento, personalizar suas funcionalidades e desenvolver serviços inovadores. Essa flexibilidade vem, no entanto, com um custo: as propriedades de segurança e de corretude em toda a rede (e.g., isolamento e acessibilidade) tornam-se muito mais difíceis de garantir, porque o

[1]A positive security culture - https://www.ncsc.gov.uk/collection/you-shape-security/a-positive-security-culture

comportamento da rede agora é determinado por uma combinação da configuração mantida pelo plano de controle e os programas do plano de dados que residem nos dispositivos de encaminhamento. Neste contexto, as ferramentas existentes para análise de segurança de redes, as quais dependem de um modelo fixo e invariante do plano de dados, são inadequadas para planos de dados programáveis.

Ao mesmo tempo, a capacidade de programar o plano de dados significa que é possível não apenas remodelar o comportamento da rede, mas também tornar a rede mais segura e confiável, melhorando sua confiabilidade, disponibilidade e integridade [Avizienis et al. 2004]. Isso pode ser feito por meio de um fluxo de serviços de segurança e confiabilidade, desenvolvidos a partir de blocos de construção provisionados diretamente nos dispositivos. Exemplos de blocos de construção incluem monitoramento e classificação de fluxo, bem como recursos de plano de dados de aplicação de políticas. Essa abordagem de provisionamento de serviços pode trazer várias vantagens exclusivas. Por exemplo, conformidade com a política pode ser garantida mesmo se o plano de controle e/ou um subconjunto de dispositivos de encaminhamento estiverem com defeito/comprometidos. Sendo assim, a medição da rede e a detecção de anomalias podem ocorrer de maneira verdadeiramente distribuída, com os dispositivos de encaminhamento de dados (*switches*) acionando prontamente ações de contramedidas, se for necessário.

Além dos requisitos de desempenho, as redes modernas podem ter políticas de segurança (explícitas ou implícitas) que definem o fluxo de informação entre *hosts*. Em uma rede *multi-tenant*, por exemplo, o operador pode querer garantir que os *tenants* estejam completamente isolados uns dos outros ou que um *tenant* não possa negar ao outro acesso à rede. Várias classes de propriedades foram consideradas pela comunidade de pesquisa: independentes de contexto (propriedades agnósticas de sessões de fluxo), dependente de contexto (referem-se aos fluxos de dados, por exemplo, iniciação da sessão), quantitativas (que são asseguradas com base em contadores, por exemplo, largura de banda garantida) e híbridas. À medida que os planos de controle e de dados se tornam mais complexos, torna-se mais difícil garantir que eles funcionem sempre corretamente. Para garantir que certas propriedades críticas sejam sempre satisfeitas, é vantajoso ter um mecanismo separado que seja apenas responsável por garantir que essas propriedades sejam respeitadas.

Esta seção visa fomentar discussão sobre a segurança de redes na era de planos de dados programáveis, ao apresentar (i) como o conceito de programabilidade do plano de dados pode ser usado para tornar as redes de computadores mais seguras, e (ii) quais os principais desafios de segurança que emergem juntamente com o conceito.

7.3.1. Modelagem e Análise de Políticas de Segurança

Uma maneira de expressar os requisitos que um sistema em rede deve atingir ou satisfazer é por meio de políticas de rede. A literatura é rica em soluções para especificação de políticas, verificação e aplicação. Boubata e Aib [Boutaba and Aib 2007], apresentam uma perspectiva histórica sobre a gestão de rede baseada em políticas. Os requisitos muitas vezes confiam em protocolos padrão para definir o que pode ser observado e executado (como endereços IP, portas TCP/UDP e outros campos de cabeçalho de protocolos padrão). A agenda de pesquisa de modelagem e análise de políticas para planos de dados programáveis deve se concentrar em três grandes questões: 1) como modelar e expressar

políticas, 2) como traduzir/refinar políticas e 3) como lidar com conflitos entre elas.

7.3.1.1. Propriedade baseadas em políticas específicas

As soluções baseadas em políticas para o plano de dados programável deve considerar classes de propriedades para expressar requisitos de nível superior/inferior que um sistema necessita satisfazer. A questão é, quais são essas classes e propriedades? Trabalhos anteriores consideraram isolamento, acessibilidade e equivalência em SDN, mas sem fornecer uma discussão conceitual de nível superior [Khurshid et al. 2013, Lopes et al. 2015].

Uma propriedade é dita independente de contexto se for agnóstica de fluxo de sessões, ou seja, pode ser definida por pacote, sem recorrer ao estado das informações. Exemplos incluem isolamento e conectividade. Por outro lado, uma propriedade é dita dependente do contexto se aborda o fluxo de pacotes fluxos dependendo de sua semântica na rede. Um exemplo é o início da sessão, que expressa em que direção as conexões podem ser iniciadas na rede (por exemplo, um *host* pode enviar uma consulta de resolução de nomes, mas não receber um). Neste caso, diz-se que algum *host* tem permissão para iniciar uma sessão com outro. Outra classe agrega propriedades quantificáveis. Os exemplos incluem largura de banda garantida, limite de largura de banda e k-redundância. A primeira expressa uma taxa mínima que um *host* tem garantido para enviar pacotes para outro. O segundo expressa uma taxa máxima permitida para o fluxo de informações entre esses *hosts*. A terceira propriedade, k-redundância (k interpretada como uma métrica de redundância), é definida para um determinado *link* lógico e especifica a existência de k outros *links* lógicos conectando o mesmo conjunto de *hosts*. Esta propriedade pode ser útil para expressar canais de *backup* e/ou melhorar a robustez contra Ataques de negação de serviço distribuído (DDoS).

Por fim, as propriedades híbridas apresentam aquelas com características de mais de uma das classes acima. Um exemplo é o *link* equivalência, que expressa que os *links* lógicos conectando quaisquer duas entidades têm o mesmo isolamento, conectividade, largura de banda, configurações, etc. Uma noção estendida da propriedade de equivalência é a redundância k-equivalente. Um *link* é dito k-equivalente redundante se houver k outros *links* conectando o mesmo conjunto de *hosts* e com propriedades equivalentes. A oportunidade de pesquisa envolve a proposta de linguagens políticas expressivas que apoiem o nível de especificação de políticas, e que simultaneamente se aproximem mutuamente de metas conflitantes. Por exemplo, essas linguagens devem ser agnósticas do formato do cabeçalho do pacote ou da semântica de análise, mas também permitem a expressão de políticas de uma maneira que corresponda ao atual comportamento do *switch*.

7.3.1.2. Tradução de políticas de nível superior para nível inferior

Como o *hardware* de rede é personalizado sob demanda e sua semântica de análise de pacotes muda com o tempo, as soluções de especificação de políticas de segurança precisam ter uma dinâmica de revisão e atualização [Udupi et al. 2007, Craven et al. 2011]. Essas políticas em um contexto de planos de dados programáveis despertam oportunida-

des de pesquisas. Por exemplo: 1) Como garantir a consistência entre políticas de nível superior e inferior [Verma 2002, Westerinen et al. 2001] à medida que o comportamento do *switch* muda?; 2) Como pode-se expressar políticas baseadas em propriedades genéricas de segurança e confiabilidade, de uma forma que as torne verificáveis e aplicáveis em qualquer configuração de plano de dados? Neste contexto, é importante definir quais classes de propriedade são de interesse, bem como entender as suas implicações no projeto de mecanismos de tradução de políticas de segurança.

Em uma rede definida por *software*, cabe ao controlador garantir que as políticas de nível superior sejam mantidas [Kreutz et al. 2013]. No entanto, à medida que os aplicativos do plano de controle e os programas de comutação do plano de dados evoluem de forma independente e se tornam mais complexos, torna-se mais difícil garantir a consistência das políticas intra e internível. Esse cenário dinâmico exige soluções que vão além da tradução de políticas e também verificam inconsistências. Um exemplo é uma política declarando que duas redes A e B devem ser isoladas (um cenário de *datacenter* multilocatário) e uma permitindo pacotes do *host* $a_i \in A$ para $b_j \in B$. Outro caso é uma política que expressa que dois *hosts* estão simultaneamente isolados e conectados.

Pesquisas anteriores consideraram casos como conflitos entre diferentes tipos de políticas de nível superior [Lupu and Sloman 1999] e análise de conflito baseada em regras [Hamed and Al-Shaer 2006]. No entanto, eles são limitados, pois consideram linguagens de especificação de políticas de nível mais alto ou são fortemente acoplados a protocolos de rede tradicionais. Sendo assim, a criação de soluções que possam garantir consistência de políticas de nível superior a inferior, considerando a especificação abstrata de programas de comutação, apresenta-se como uma avenida de pesquisa promissora a ser explorada pela comunidade de pesquisa.

7.3.2. Verificação de Políticas de Segurança

A imposição e a verificação são abordagens complementares que podem ser aplicadas como solução para garantir que políticas de segurança sejam respeitadas. Usando a imposição, o plano de dados pode ser monitorado durante a execução para buscar e bloquear ações que resultem em violações das políticas. A verificação (em conjunto com validação) se concentra em encontrar os *bugs* antes que os programas sejam implantados. Ela atua assegurando que o programa atenda às propriedades declaradas por seus requisitos.

Em um mundo onde os gerentes e operadores de rede podem redefinir o comportamento de dispositivos de encaminhamento, escrevendo seus próprios códigos para implementar alguma especificação de protocolo, a verificação e validação adequada (V&V) do código dos dispositivos torna-se crítica para o gerenciamento adequado das operações de rede e, portanto, a continuidade dos negócios. Em 2016, um roteador com defeito forçou a *Southwest Airlines* a cancelar 2.300 voos em quatro dias, resultando em uma perda de US$ 74 milhões [Carey 2017]. Alguns anos depois (julho de 2020), uma configuração de roteador defeituosa na *Cloudflare* causou uma interrupção de rede que durou apenas 27 minutos, mas levou a uma grande interrupção dos serviços de Internet em todo o mundo por mais de uma hora [Winder 2020]. A comunidade de redes tem pesquisado soluções para lidar com defeitos de *software* antes que eles causem tais danos. Abordagens como metadados sintáticos, execução simbólica, asserções e testes funcionais têm sido aplicadas

ao teste de *software* de plano de dados. Nesta seção são abordadas algumas das técnicas utilizadas para verificação e validação para *software* de plano de dados programáveis.

7.3.3. Imposição (*Enforcement*) de Políticas de Segurança

Uma alternativa à verificação é a imposição (enforcement). Em vez de verificar se uma configuração de rede está correta, um *kernel* de segurança logicamente separado evita ações que violem a política de segurança. O *kernel* de segurança deve mediar todas as ações de manipulação de pacotes no plano de dados. Ao contrário do modo de verificação, onde verifica-se as violações da política antes de uma configuração ser enviada para a rede, no modo de imposição, verifica-se as violações da política, uma vez que estão prestes a ocorrer. Tanto a verificação como a imposição têm suas vantagens e desvantagens. Por um lado, a verificação capta problemas precocemente; um verificador pode fornecer informações de diagnóstico detalhadas sobre por que uma configuração viola uma política durante a fase de verificação. No regime de imposição, os problemas são detectados à medida que ocorrem. A imposição pode ser mais atrativa do que a verificação, porque não depende da complexidade do programa de controle ou do plano de dados.

Sendo assim, emergem benefícios para imposição (*enforcement* da política de segurança em plano de dados programáveis. Os planos permitem que os operadores de rede modifiquem o *pipeline* de processamento de pacotes dos dispositivos de rede para implementar novos protocolos, personalizar o comportamento da rede e estabelecer serviços de rede avançados. No que pese a sua simplicidade da programação, os programas P4 demonstraram ser propensos a uma variedade de *bugs* e erros de configuração [Stoenescu et al. 2016, Freire et al. 2018]. Como resultado, os operadores de rede precisam de estruturas para garantir que os programas que produzem tenham um comportamento correto para obter os benefícios de um ecossistema de *software* de plano de dados. Ferramentas de verificação de rede de última geração podem obter um modelo da rede, sua configuração e um conjunto de propriedades específicas usando formalismos tradicionais (por exemplo, lógica temporal ou regras de *Datalog*) e verificar automaticamente se essas propriedades são válidas para qualquer pacote [Beckett et al. 2017, Lopes et al. 2015].

Embora essas ferramentas ajudem os operadores de rede a identificar *bugs* antes que eles se manifestem, deve-se considerar: (i) Primeiro que a maioria dessas ferramentas exige que os programadores modelem manualmente os planos de dados programáveis, atividade complexa e propensa a erros [Lopes et al. 2015]; (ii) Em segundo lugar, essas ferramentas são geralmente restritas em termos de propriedades de acessibilidade para reduzir os tempos de verificação [Lopes et al. 2016]; (iii) Terceiro, ferramentas mais expressivas capazes de verificar múltiplas propriedades frequentemente enfrentam problemas graves de escalabilidade (por exemplo, verificar a conformidade com uma especificação de protocolo pode levar dias, mesmo para um único plano de dados programáveis; e (iv) Por fim, os programadores precisam ter habilidades técnicas formais de verificação para especificar corretamente suas propriedades.

Neves et al. [Neves et al. 2021] apresentam uma nova abordagem baseada na aplicação dinâmica (ou em tempo de execução) em vez de verificação estática. Essa abordagem tem várias vantagens práticas. Já que não é necessário esperar pelo resultado de um longo processo de verificação para enviar uma nova configuração para os *switches*

de rede. Sendo assim, a aplicação do tempo de execução pode intervir prontamente se situações problemáticas realmente ocorrerem, possibilitando: obter informações úteis do código com *bugs* quando ele tem um comportamento correto e reparar problemas sem interferir em qualquer serviço de rede.

Em contraposição com a verificação estática, a aplicação do tempo de execução também permite ao programador expressar a política e o mecanismo usando o mesmo ambiente de programação que o resto do programa. Esse valor deve ser considerado, não só porque facilita a vida do programador, como evita também erros de tradução entre a implementação e as políticas. Sendo assim, para perceber os benefícios de uma aplicação dinâmica, Neves et al. [Neves et al. 2021] desenvolveram o P4box, um sistema para implantação de monitores de tempo de execução em planos de dados programáveis.

Usando P4box os programadores podem anexar monitores antes e depois dos blocos de controle, transições de estado do analisador e chamadas para funções externas de um programa P4. Cada monitor pode modificar a entrada e saída do bloco de código ou função que monitora, permitindo a verificação de pré e pós-condições a serem utilizadas para impor propriedades específicas ou modificar o comportamento do bloco monitorado.

Um monitor de tempo de execução insere-se na interação de um bloco de controle P4 ou analisador com o restante do ambiente de execução, permitindo que o programador do monitor modifique o comportamento do bloco P4 incluso com o restante do ambiente. Um bloco programável P4 faz a *interface* com o restante do ambiente de execução P4 na entrada no bloco, retornar do bloco as chamadas para funções externas fornecidas pela arquitetura. Na programação do modelo P4box, quando um bloco programável é invocado, o controle passa primeiro para um monitor, também escrito em P4, antes de passar para o bloco programável pretendido. Da mesma forma, quando um bloco programável completa o processamento, o controle passa primeiro para o monitor antes de retornar ao dispositivo, permitindo que um monitor modifique o comportamento de blocos programáveis de maneira bem definida.

7.3.4. Explorando Planos de Dados Programáveis para Detectar Ataques DDoS

Ataques de negação de serviço distribuído (DDoS) fazem uso dos limites de capacidade específicos aplicados a todos os recursos da rede. Esses ataques dependem de *botnet* para esgotar recursos computacionais e interromper aplicações na Internet [Hoque et al. 2015]. Buscam encaminhar um grande número de solicitações para o recurso tecnológico invadido, visando exceder a sua capacidade, interrompendo o seu funcionamento.

À medida que os *botnets* aumentam a sua aplicabilidade para explorar os dispositivos IoT (Internet das Coisas) vulneráveis, a frequência, a capacidade e o volume dos ataques DDoS amplia o seu alcance drasticamente. A detecção dessa ameaça é o primeiro passo para minimizar as perdas por meio do desencadeamento das medidas defensivas, no entanto, representa um desafio para a pesquisa em rede [Antonakakis et al. 2017, Anstee et al. 2017, Zargar et al. 2013].

Preferencialmente, a detecção e o bloqueio de ataques DDoS devem ocorrer nas fontes para economizar esforços de deslocamento e processamento sobre o tráfego indesejado [Gil and Poletto 2001, Mirkovic et al. 2002, Peng et al. 2004]. No entanto, isso

é impedido pela disseminação da atividade maliciosa, que é construída a partir da sincronização de solicitações aparentemente legítimas. Além disso, essas fontes normalmente pertencem a diferentes domínios administrativos, nos quais as políticas de segurança são definidas de forma independente. Mais adiante, nas proximidades da vítima, apesar do tráfego de ataque ser mais proeminente para detecção [Kim et al. 2006, Hoque et al. 2015], ele pode já ter saturado recursos *in-path*. A alternativa é implantar medidas defensivas em Provedores de Serviços de Internet (ISPs), que gerenciam a comunicação [Haq et al. 2015, Kang et al. 2016]. Os ISPs se beneficiam de uma visão privilegiada do tráfego e contam com *links* de alta taxa de transferência, permitindo que eles descubram e impeçam as ameaças em tempo hábil.

Ao contrário dos *datacenters*, onde o monitoramento de rede sofisticado pode ser realizado em *hosts* finais [Moshref et al. 2016, Yu et al. 2011], os ISPs dependem de *switch primitive* como amostragem de pacotes [CiscoNetworks 2017, Sflow 2017] e contagem baseada em fluxo [McKeown et al. 2008]. Os dados resultantes são então normalmente montados em servidores fora de banda para inspeção. Enquanto essas primitivas apresentam compensações entre granularidade de visibilidade, utilização de largura de banda, espaço de memória e a comunicação com servidores externos incorre em uma latência adicional para detectar eventos de rede [Moshref et al. 2013]. A fim de manter a utilização razoável da largura de banda e a carga de processamento, a amostragem de pacotes é geralmente empregada em taxas agressivamente baixas [Phaal 2009], apenas transmitindo informações de um conjunto limitado de pacotes. Diferentemente, a contagem baseada em fluxo, como em *switches* OF [McKeown et al. 2008], fornece valores exatos para métricas volumétricas com um alto custo de entradas nas tabelas.

Como alternativa promissora para este problema, o conceito emergente de programabilidade do plano de dados oferece flexibilidade para a execução de algoritmos nos *switches* de rede [Bosshart et al. 2014]. Assumindo um fluxo de pacotes como entrada, esses algoritmos são modelados como um *pipeline* de primitivas elementares, acessos à memória e pesquisas em tabelas. Sendo assim, os operadores podem definir funções de monitoramento e delegá-las a dispositivos de plano de dados em toda rede. Essa arquitetura pode ser explorada para realizar a inspeção em cada pacote sem incorrer em sobrecarga de comunicação. No entanto, buscando executar a taxa de linha com custos razoáveis, o processamento de pacotes é restrito a um pequeno orçamento de tempo e uma quantidade limitada de memória por estágio de *pipeline* [Bosshart et al. 2013].

Lapolli et. al [Lapolli et al. 2019] desenvolveram uma arquitetura de sistema para detecção de DDoS, na qual o plano de dados responsável pela coleta do fluxo das métricas e sua inspeção. Isso é apresentado na forma de uma detecção de ataque DDoS em banda sistema totalmente implementável em uma chave programável através de P4. O trabalho compreende um *pipeline* de processamento para estimar as entropias dos endereços IP de origem e destino. Esses valores são usados para caracterizar o tráfego supostamente legítimo em tempo real. Os resultados desta caracterização servem para calcular a detecção limiares considerando um coeficiente de sensibilidade parametrizável. A fim de respeitar o rigoroso orçamento de tempo e restrições de memória para o cálculo da entropia, a frequência de endereços IP distintos é aproximada por esboços de contagem aprimorados [Charikar et al. 2002]. Outras funções aritméticas de computação intensiva são resolvidas com a ajuda de uma tabela de pesquisa *Longest Match Routing Rule* (LPM) otimizada

para memória.

7.3.5. Depuração e Rastreabilidade de Aplicações em Planos de Dados Programáveis

Planos de dados programáveis permitem que a execução de aplicações cruzem a fronteira entre servidores x86 tradicionais e a rede de computadores, habilitando o descarregamento (ou seja, *offloading*) de partes da computação para PDPs. Esse paradigma tem sido chamado de *in-network computing* [Benson 2019]. À luz desse desenvolvimento, tanto a indústria quanto pesquisadores começaram a investigar ativamente novos projetos para aplicações distribuídas a fim de melhorar o desempenho, a escalabilidade ou a confiabilidade dessas, transferindo parte de sua funcionalidade para a rede. Dessa forma, uma vasta gama de problemas tem explorado essa possibilidade de descarregar parte da computação para a rede: *Caching*: NetCache [Jin et al. 2017] armazena em cache pares de chave-valor em switches, evitando potencialmente longos RTTs para acessar um servidor de armazenamento de chave-valor remoto; *Agregação de Dados*: DAIET [Sapio et al. 2017] realiza agregação de dados na rede para maior escalabilidade; *Machine Learning*: machine learning dentro de switches pode mitigar gargalos existentes durante o treinamento distribuído de modelos [Sanvito et al. 2018, Xiong and Zilberman 2019a]; *Pattern Matching*: a correspondência de padrões eficiente pode ser alcançada através da realização de parte da computação na rede [Jepsen et al. 2019].

À medida que essas abordagens recém-descobertas se aproximam da implantação, surgem preocupações práticas sobre seu gerenciamento em tempo de execução, porque as aplicações distribuídas agora podem executar parcialmente no plano de dados. Especificamente, a incorporação de lógica em PDPs adicionou outra camada de complexidade para rastrear e solucionar problemas dessas aplicações, e esforços tradicionais de rastreabilidade e observabilidade de aplicações em servidores x86 tradicionais não se traduzem diretamente para *in-network computing* [Benson 2019]. Em particular, switches programáveis atuais não fornecem uma abstração rica o suficiente para suportar técnicas de rastreamento tradicionais [Sigelman et al. 2010, Mace and Fonseca 2018, Chow et al. 2014], e essa falta de primitivas de rastreamento força os programadores a criarem suas próprias soluções exclusivas. Isso leva à criação de ferramentas de rastreamento muito específicas e não reutilizáveis para depurar a computação na rede. Mais importante, rastros produzidos por soluções específicas para o PDP provavelmente não serão interoperáveis com estruturas de diagnóstico de rastreamento existentes, por exemplo, Dapper[Sigelman et al. 2010] do Google. Ortogonalmente, as estruturas de rastreamento existentes não fornecem primitivas para gerar ou capturar dados de rastreamento em planos de dados programáveis.

Um desafio de pesquisa atual visa preencher a lacuna entre técnicas tradicionais para telemetria de redes e *frameworks* de rastreamento distribuído. Isso requer abordar execuções que cruzem a fronteira da aplicação distribuída para o plano de dados programável, capturando dados de rastreamento de PDPs e apresentando-os ao plano de aplicação por meio de uma abstração flexível e bem definida. Um dos primeiros esforços nessa direção é o P4-Intel [Castanheira et al. 2019], que (i) aproveita a telemetria de rede para instrumentar PDPs no monitoramento de dados de rastreamento arbitrários definidos pelo usuário e (ii) coordena o armazenamento, coleta e formatação desses dados de rastreamento internamente, fornecendo apenas dados de contexto bem formados para qualquer

ferramenta de depuração do plano de aplicação.

7.4. Inteligência Artificial e Segurança

Para "o bem e para o mal", a área de Inteligência Artificial (IA) vem impactando substancialmente a Segurança Cibernética. Por um lado, *deepfakes* tornam mais fáceis golpes virtuais. Por outro, IA tem potencial para melhorar os processos de segurança, por exemplo, via identificação automatizada de fraudes e ações suspeitas. Além de como usar IA para melhorar a segurança, impõe-se como questão central considerar aspectos-chave como ética, transparência, responsabilidade, explicabilidade e confiabilidade. Considerando-se a experiência do Grupo de Redes de Computadores nesse grande tema, a seguir, aborda-se um específico: a oportunidade de se capitalizar redes programáveis como base para o projeto e o desenvolvimento de mecanismos *in-network* inteligentes voltados à proteção de redes e serviços.

Um marco significativo na evolução de Redes Definidas por Software (SDN) foi o desenvolvimento do OpenFlow como uma implementação real de SDN. No entanto, a operação da rede ainda está limitada ao conjunto de protocolos e cabeçalhos suportados pelo hardware dos dispositivos de encaminhamento (ex: switches e interfaces de rede). Assim, a definição de funções personalizadas para o processamento de pacotes torna-se muito difícil. Recentemente, o conceito de Planos de Dados Programáveis (PDPs) surgiu para superar essas limitações. Os PDPs permitem o controle completo do comportamento da rede, desde as aplicações até o processamento de pacotes dentro dos dispositivos, incluindo a definição e a análise de cabeçalhos personalizados. Tal proporciona uma oportunidade sem precedentes para desenvolver novos recursos nos dispositivos de encaminhamento e revisitar funções existentes para o gerenciamento de redes [Cordeiro et al. 2017]. Atualmente, a linguagem P4 é o padrão de fato para descrever como os pacotes de rede devem ser processados.

Uma das áreas que pode se beneficiar de PDPs/P4 é a de Segurança de Redes. Sistemas de Detecção de Intrusão (IDSs) podem ser aprimorados implementando-os como funções eficientes implantadas no plano de dados, capazes de reagir rapidamente a anomalias de rede que possam representar ameaças. IDSs geralmente dependem da coleta de características de tráfego (*traffic features*), que são posteriormente alimentadas em sistemas sofisticados baseados principalmente em algoritmos de Aprendizado de Máquina (*Machine Learning* – ML). ML tem sido usada com sucesso em segurança de redes devido à sua capacidade de detectar e descobrir padrões e comportamentos não observados anteriormente no tráfego. A maioria das abordagens de segurança desenvolvidas no contexto de SDN e baseadas em ML foram implementadas exclusivamente no plano de controle, apesar dos problemas associados à precisão e à sobrecarga significativa que podem introduzir [Xie et al. 2019].

As funcionalidades introduzidas pelos PDPs tornam possível considerar um novo cenário para soluções de segurança baseadas em ML, aproveitando as capacidades de transferência (*offloading*) de parte dos algoritmos para os dispositivos de encaminhamento. Assim, soluções mais precisas e responsivas podem ser implantadas. A decisão sobre quanto das funções deve ser transferido para o plano de dados não é trivial [Ports and Nelson 2019], pois as capacidades de computação dos dispositivos de rede são limi-

tadas, e o *offloading* excessivo de funcionalidades pode prejudicar a vazão máxima no encaminhamento de pacotes.

A seguir, discute-se alguns desafios enfrentados na interseção entre Planos de Dados Programáveis e IA/ML para detecção de intrusão. Foca-se em como aproveitar as funcionalidades dos PDPs na implementação de IA/ML, especialmente algoritmos de ML. Aborda-se, principalmente, a questão de quanto das operações dos algoritmos é viável ser transferida para dispositivos de encaminhamento. A reflexão tem como base o esforço realizado por Gutiérrez *et al.* [Gutiérrez et al. 2021].

7.4.1. Aprendizado de Máquina em Planos de Dados Programáveis para melhorar a Detecção de Intrusão

O Aprendizado de Máquina tornou-se um marco essencial em vários tipos de soluções de segurança cibernética devido à sua capacidade de extrair anomalias e padrões que podem ser sintomas de ataques internos ou externos contra a infraestrutura. Essas soluções são geralmente integradas por componentes de segurança de rede e *host*, incluindo *firewalls*, antivírus e IDSs [Le and Zincir-Heywood 2020].

Os IDSs estão sendo revisitados para melhor aproveitar as possibilidades habilitadas pelo novo contexto de redes programáveis. A maioria das soluções desenvolvidas para a implementação de IDSs foi implantada como aplicações em execução no plano de controle. No entanto, essa abordagem para a implementação de IDSs tem duas principais desvantagens. Primeiro, o conjunto de características de tráfego derivadas de contadores padrões (por exemplo, aqueles disponíveis nas versões atuais do OpenFlow) ou, em situações extremas, via eventos `PACKET_IN`, é insuficiente para obter precisão razoável nos algoritmos de ML. Segundo, os algoritmos de ML são, normalmente, intensivos em computação. Se não forem adequadamente projetados e implantados, podem introduzir sobrecarga no plano de controle e prejudicar o funcionamento correto da rede [Binbusayyis and Vaiyapuri 2019].

Como introduziu-se anteriormente, o surgimento dos PDPs torna possível o *offloading* de algumas funcionalidades para o plano de dados. A seguir, apresenta-se uma visão geral das etapas de um IDS baseado em ML e delineia-se como os PDPs podem ser aproveitados para melhorar algumas dessas etapas por meio do processamento personalizado de pacotes e do *offloading* de operações específicas [Le and Zincir-Heywood 2020]. A Figura 7.1 apresenta a visão sequencial e as relações entre essas etapas.

Coleta de Dados. Os PDPs estendem as possibilidades de coleta de dados além das estatísticas padrões disponíveis nos dispositivos de encaminhamento. Estatísticas personalizadas podem ser introduzidas, e algum processamento com estado (*stateful*) pode ser incluído nos dispositivos, o que pode se traduzir em indicadores de grande valor para tarefas de detecção de intrusão [Kohler et al. 2018]. Apesar das limitações inerentes ao poder de computação e às primitivas de programação disponíveis em dispositivos de encaminhamento programáveis, duas funcionalidades podem ser aproveitadas para implementar a coleta de dados eficiente para algoritmos de ML: análise personalizada de pacotes e agregação de dados. A análise personalizada de pacotes permite o processamento de cabeçalhos que podem ser usados para calcular estatísticas personalizadas, por

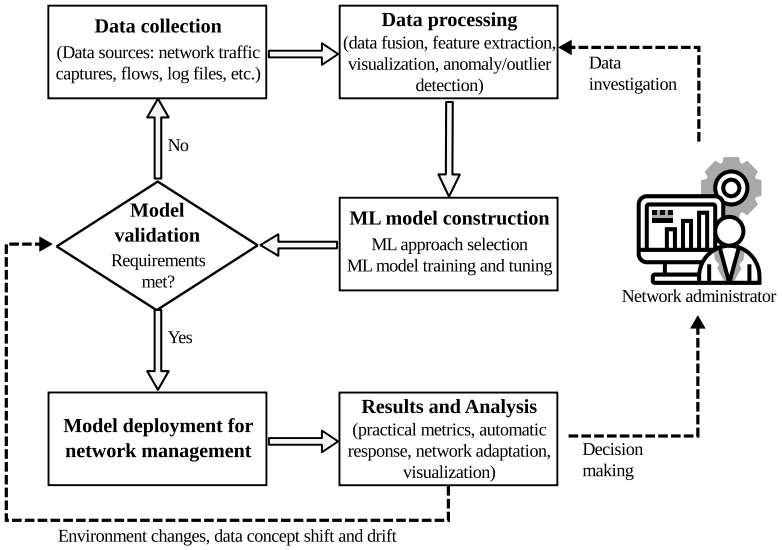

Figura 7.1: Estágios de um sistema baseado em Aprendizado de Máquina aplicado ao domínio de Gerenciamento de Redes (extraído de [Gutiérrez et al. 2021] *apud* [Le and Zincir-Heywood 2020].

exemplo baseadas em campos de novos protocolos [Gupta et al. 2018]. Já a agregação ajuda a reduzir a quantidade de dados que precisa ser transmitida dos dispositivos para o Plano de Controle para a execução de operações complexas [Sapio et al. 2021].

Processamento de Dados. Propostas recentes introduzem o conceito de processamento na rede como um serviço, onde a implementação de operações no plano de dados fica disponível para uso por estruturas de alto nível de ML ou análise de dados [Mustard et al. 2019, Sapio et al. 2017]. Outras propostas introduzem a noção de consultas que acionam a coleta e a análise preliminar de dados para produzir estatísticas que podem ser posteriormente entregues a mecanismos de processamento de alto nível [Gupta et al. 2018]. A ideia central por trás dessas abordagens é que recursos dos PDPs permitem não apenas medições e contagens, mas também a realização de análises em paralelo com a coleta de dados.

Construção do Modelo. A literatura apresenta várias abordagens para aproveitar as funcionalidades disponíveis nos PDPs para implementar diferentes algoritmos, considerando as restrições computacionais dos dispositivos de encaminhamento programáveis [Ports and Nelson 2019]. Essas propostas incluem o uso de registradores de switches (para implementar aritmética e armazenamento de valores), tabelas de correspondência (*match-action*), entre outros construtos, para a implementação de técnicas como Árvores de Decisão, Máquinas de Vetores de Suporte (SVM), classificadores Naive Bayes e Redes Neurais [Qin et al. 2020]. Essa abordagem reduz a quantidade de informações que precisam ser encaminhadas para o plano de controle (por exemplo, eventos `PACKET_IN`), o que contribui para diminuir a sobrecarga do canal de controle [Macías et al. 2020], ao mesmo tempo que aumenta a precisão e a capacidade de resposta [Ports and Nelson

2019, Xiong and Zilberman 2019b]. Além da implementação direta nos dispositivos, outra abordagem a ser seguida é a cooperação na formação de modelos em grande escala por meio da análise de métricas locais. Essa abordagem é chamada de Aprendizado Federado e pode ser usada para treinar modelos complexos, como Redes Neurais Profundas [Qin et al. 2020].

Validação do Modelo. A validação é uma tarefa de alto nível que envolve análise extensa e *feedback* de especialistas humanos. Portanto, os PDPs não têm intervenção direta nas tarefas associadas a essa etapa. No entanto, funcionalidades como Processamento de Eventos Complexos [Kohler et al. 2018] e telemetria baseada em consultas [Gupta et al. 2018] são úteis para fornecer *insights* para depurar situações de baixa precisão e baixo desempenho dos algoritmos de ML.

Implantação. Esta operação deve considerar as particularidades envolvidas no desenvolvimento dos algoritmos. Por exemplo, a disponibilidade, em uma determinada arquitetura de hardware, do tipo de tabelas necessárias ou o número de registradores que podem ser usados para armazenar o estado dos pacotes são aspectos que devem ser validados [Qin et al. 2020]. Para uma discussão detalhada dos problemas associados à implantação de algoritmos de ML em dispositivos de encaminhamento programáveis, consulte [Xiong and Zilberman 2019b].

Análise de Resultados. Funcionalidades como Telemetria de Rede em Banda, que dependem de recursos dos PDPs [Gupta et al. 2018], e Processamento de Eventos Complexos [Kohler et al. 2018] podem fornecer *insights* importantes para essa etapa. Além disso, a definição tanto de limiares para *features* específicas quanto de intervalos de tempo adequados para análise contribuem para avaliar a eficácia dos algoritmos, permitindo algum grau de análise dos dados.

7.4.1.1. BUNGEE-ML: Um Estudo de Caso

BUNGEE-ML é um sistema que combina o processamento rápido do plano de dados e a alta capacidade e inteligência do plano de controle para detecção precisa e mitigação de ataques na rede. Avanço mais recente de toda uma linha de trabalhos [Lapolli et al. 2019, Ilha et al. 2021, González et al. 2021], o sistema implementa uma estratégia de vários níveis [Marnerides et al. 2011] para garantir a operação contínua da rede, promovendo a cooperação vertical e horizontal entre os elementos da rede (Fig. 7.2):

- *Cooperação vertical*: para contornar as limitações de processamento do ASIC (*Application-Specific Integrated Circuit*) dos dispositivos de encaminhamento programáveis, BUNGEE-ML realiza uma análise de tráfego mais sofisticada fora do ASIC, uma abordagem *vertical*, que depende dos recursos da CPU do switch e do controlador SDN. Essa análise *profunda* prioriza a precisão e pode corrigir decisões tomadas no ASIC.

- *Cooperação horizontal*: Aproveitando a topologia programável, BUNGEE-ML "empurra" oportunisticamente o tráfego malicioso o mais longe possível da vítima, uma estratégia de mitigação em *largura* no plano de dados, que permite respostas rápidas a ataques DDoS.

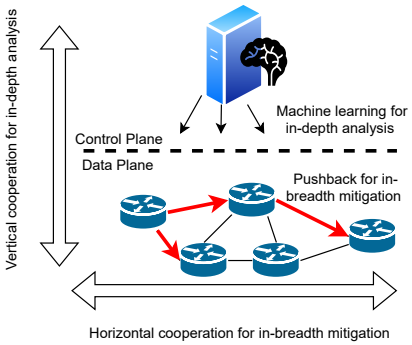

Figura 7.2: Cooperação vertical e horizontal (extraída de [González et al. 2023]).

Essencialmente, BUNGEE-ML permite que ambos os planos cooperem, explorando suas forças individuais. Primeiro, o ASIC do switch executa estratégias leves que possibilitam a detecção precoce de tráfego suspeito na taxa de linha. Isso é combinado com processamento ligeiramente mais sofisticado para comparar estatísticas de tráfego recentes na CPU do switch. No entanto, para realizar uma análise mais profunda e sofisticada das fontes suspeitas, o plano de controle aplica técnicas de Aprendizado de Máquina para decidir se os suspeitos (identificados pelo plano de dados) são atacantes.

Embora as ações de mitigação possam ser implantadas assim que o switch tenha marcado um fluxo como suspeito – por exemplo, o plano de dados pode reduzir seletivamente (*throttling*) o tráfego das fontes suspeitas para lidar rapidamente com o ataque, as contramedidas tornam-se permanentes após o plano de controle confirmar os fluxos de ataque. Nesse caso, o plano de dados implementa uma estratégia de recuo nos suspeitos confirmados, incentivando dispositivos *upstream* a construir uma frente de mitigação colaborativa e parar o ataque o mais longe possível da vítima.

A Fig. 7.3 ilustra o fluxo geral do BUNGEE-ML, mostrando as interações entre seus componentes nos planos de controle e dados:

- A etapa de monitoramento de fluxos (❶) é a base da implementação. O sistema realiza monitoramento contínuo e executa uma estratégia com base na análise de entropia dos pacotes de entrada usando os ASICs nos dispositivos do plano de dados para detectar mudanças no comportamento da rede durante uma "janela de monitoramento".

- No caso de um ataque, o monitoramento de fluxos aciona um alerta para o componente de *Window Inspection* (❷). Nesse componente, os endereços de origem mais recentes da janela de monitoramento são comparados com estatísticas globais para identificar os suspeitos que estão causando a perturbação na rede.

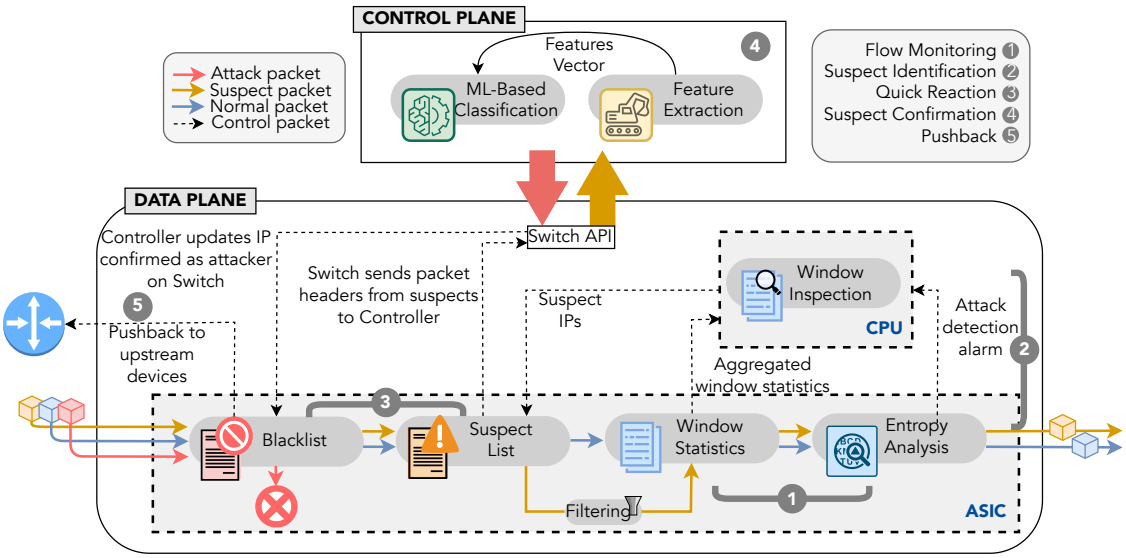

Figura 7.3: Visão geral do BUNGEE-ML (extraída de [González et al. 2023]).

- Uma reação rápida começa depois que as fontes suspeitas são identificadas (❸). Isso inclui a manutenção de uma *Lista de Suspeitos* para que pacotes subsequentes desses suspeitos sejam filtrados. Pacotes suspeitos de entrada são desviados para o plano de controle, onde uma inspeção adicional é realizada para determinar se as fontes são atacantes ou não.

- O plano de controle extrai características dos pacotes para classificar os endereços de origem usando mecanismos de Aprendizado de Máquina (❹). Após a classificação de um endereço, o plano de controle notifica o plano de dados para *(a)* remover os endereços classificados como benignos da *Lista de Suspeitos* ou *(b)* confirmar os endereços classificados como maliciosos, ou seja, incluí-los em uma *Lista Negra*. Pacotes de entrada de fontes incluídas na *Lista Negra* são descartados.

- Por fim, o switch alerta os dispositivos "upstream" sobre o ataque em andamento (❺) enviando a lista de suspeitos confirmados para tomar as medidas de mitigação apropriadas, que se denomina como ação de recuo.

As etapas ❶, ❷, ❸, e ❺ são todas executadas no plano de dados para detectar e mitigar um ataque. Enquanto isso, o plano de controle aprimora a lista de suspeitos formada pelo plano de dados (❹) para melhorar a precisão da classificação e mitigação.

7.5. Considerações Finais

Violações de cibersegurança custam trilhões anualmente às organizações. Assim, são necessários mecanismos que assegurem a proteção dos ativos das ameaças a sua integridade, disponibilidade e confidencialidade. As organizações tradicionalmente implementam esses mecanismos contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. No entanto, os avanços em diversas áreas da computação necessitam ser acompanhados de avanços em cibersegurança.

O presente capítulo discute algumas tendências para cibersegurança nos próximos

anos. Inicialmente, a Segurança Centrada nas Pessoas foi discutida, incluindo Engenharia Social e as evoluções que tem acompanhado a utilização de fatores humanos a cibersegurança. Em seguida, o uso de Inteligência Artificial e Segurança foi relatado, considerando mecanismos de detecção de intrusão e mitigação de ataques. Finalmente, A Segurança na era dos planos de dados programáveis foi apresentada, explorando a relação entre a mecanismos de segurança e a programabilidade do plano de dados.

Apesar da discussão apresentada no capítulo, novos tópicos podem ser trazidos em trabalhos futuros. A ampliação de funcionalidade de computadores quânticos implica em riscos para diversos mecanismos de criptografia usados atualmente. Dessa forma, é necessário o desenvolvimento e a implementação de algoritmos e protocolos de Criptografia Pós-Quântica. Finalmente, a compreensão dos desafios éticos em Computação é fundamental para assegurar uma ambiente digital seguro e protegido. Assim, repercussões filosóficas e sociais precisam ser integrados às discussões técnicas.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

[Al-Charchafchi et al. 2019] Al-Charchafchi, A., Manickam, S., and Alqattan, Z. N. (2019). Threats against information privacy and security in social networks: A review. In *International Conference on Advances in Cyber Security*, pages 358–372. Springer.

[Anstee et al. 2017] Anstee, D., Bussiere, D., Sockrider, G., and Morales, C. (2017). Worldwide infrastructure security report. *Arbor Networks Inc., Westford, MA, USA*.

[Antonakakis et al. 2017] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., et al. (2017). Understanding the mirai botnet. In *26th USENIX security symposium (USENIX Security 17)*, pages 1093–1110.

[Avizienis et al. 2004] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33.

[Beal 2005] Beal, A. (2005). Segurança da informação: Princípios e melhores práticas para a proteção dos ativos de informação nas organizações. *Atlas*.

[Beckett et al. 2017] Beckett, R., Gupta, A., Mahajan, R., and Walker, D. (2017). A general approach to network configuration verification. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 155–168.

[Benson 2019] Benson, T. A. (2019). In-network compute: Considered armed and dangerous. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, pages 216–224, New York, NY, USA. ACM.

[Binbusayyis and Vaiyapuri 2019] Binbusayyis, A. and Vaiyapuri, T. (2019). Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. *IEEE Access*, 7:106495–106513.

[Bosshart et al. 2014] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. 44 (3): 87–95, july 2014.

[Bosshart et al. 2013] Bosshart, P., Gibb, G., Kim, H.-S., Varghese, G., McKeown, N., Izzard, M., Mujica, F., and Horowitz, M. (2013). Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. *ACM SIGCOMM Computer Communication Review*, 43(4):99–110.

[Boutaba and Aib 2007] Boutaba, R. and Aib, I. (2007). Policy-based management: A historical perspective. *Journal of Network and Systems Management*, 15(4):447–480.

[Carey 2017] Carey, S. (2017). Why a single failed router can ground a thousand flights. *The Wall Street Journal*.

[Castanheira et al. 2019] Castanheira, L., Schaeffer-Filho, A., and Benson, T. A. (2019). P4-intel: Bridging the gap between icf diagnosis and functionality. In *Proceedings of the 1st ACM CoNEXT Workshop on Emerging In-Network Computing Paradigms*, ENCP '19, page 21–26, New York, NY, USA. Association for Computing Machinery.

[Castells 2002] Castells, M. (2002). *A sociedade em rede*. Editora Paz e Terra.

[Charikar et al. 2002] Charikar, M., Chen, K., and Farach-Colton, M. (2002). Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer.

[Chow et al. 2014] Chow, M., Meisner, D., Flinn, J., Peek, D., and Wenisch, T. F. (2014). The mystery machine: End-to-end performance analysis of large-scale internet services. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 217–231, Broomfield, CO. USENIX Association.

[CiscoNetworks 2017] CiscoNetworks (2017). Cisco ios netflow. In *Accessed on June, 29 2017*.

[Cordeiro et al. 2017] Cordeiro, W., Marques, J., and Gaspary, L. (2017). Data plane programmability beyond openflow: Opportunities and challenges for network and service operations and management. *J. Netw. Syst. Manage.*, 25(4):784–818.

[Craven et al. 2011] Craven, R., Lobo, J., Lupu, E., Russo, A., and Sloman, M. (2011). Policy refinement: Decomposition and operationalization for dynamic domains. In *2011 7th International Conference on Network and Service Management*, pages 1–9. IEEE.

[Freire et al. 2018] Freire, L., Neves, M., Leal, L., Levchenko, K., Schaeffer-Filho, A., and Barcellos, M. (2018). Uncovering bugs in p4 programs with assertion-based verification. In *SOSR*, page 4. ACM.

[Gil and Poletto 2001] Gil, T. M. and Poletto, M. (2001). {MULTOPS}: A {Data-Structure} for bandwidth attack detection. In *10th USENIX Security Symposium (USENIX Security 01)*.

[González et al. 2023] González, L. A. Q., Castanheira, L., Marques, J. A., Schaeffer-Filho, A. E., and Gaspary, L. P. (2023). Bungee-ml: A cross-plane approach for a collaborative defense against ddos attacks. *J. Netw. Syst. Manag.*, 31(4):77.

[González et al. 2021] González, L. A. Q., Castanheira, L., Marques, J. A., Schaeffer-Filho, A., and Gaspary, L. P. (2021). Bungee: An adaptive pushback mechanism for ddos detection and mitigation in p4 data planes. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 393–401.

[Gupta et al. 2018] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and Willinger, W. (2018). Sonata: Query-driven streaming network telemetry. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 357–371, New York, NY, USA. Association for Computing Machinery.

[Gutiérrez et al. 2021] Gutiérrez, S. A., Branch, J. W., Gaspary, L. P., and Botero, J. F. (2021). Watching smartly from the bottom: Intrusion detection revamped through programmable networks and artificial intelligence. arXiv cs.NI 2106.00239.

[Guzman and Lewis 2020] Guzman, A. L. and Lewis, S. C. (2020). Artificial intelligence and communication: A human–machine communication research agenda. *New Media & Society*, 22(1):70–86.

[Hamed and Al-Shaer 2006] Hamed, H. and Al-Shaer, E. (2006). Taxonomy of conflicts in network security policies. *IEEE Communications Magazine*, 44(3):134–141.

[Haq et al. 2015] Haq, O., Abaid, Z., Bhatti, N., Ahmed, Z., and Syed, A. (2015). Sdn-inspired, real-time botnet detection and flow-blocking at isp and enterprise-level. In *2015 IEEE International Conference on Communications (ICC)*, pages 5278–5283. IEEE.

[Hoque et al. 2015] Hoque, N., Bhattacharyya, D. K., and Kalita, J. K. (2015). Botnet in ddos attacks: trends and challenges. *IEEE Communications Surveys & Tutorials*, 17(4):2242–2270.

[Huber et al. 2009] Huber, M., Kowalski, S., Nohlberg, M., and Tjoa, S. (2009). Towards automating social engineering using social networking sites. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 117–124. IEEE.

[Ilha et al. 2021] Ilha, A. d. S., Lapolli, A. C., Marques, J. A., and Gaspary, L. P. (2021). Euclid: A fully in-network, p4-based approach for real-time ddos attack detection and mitigation. *IEEE Transactions on Network and Service Management*, 18(3):3121–3139.

[Jepsen et al. 2019] Jepsen, T., Alvarez, D., Foster, N., Kim, C., Lee, J., Moshref, M., and Soulé, R. (2019). Fast string searching on pisa. In *Proceedings of the 2019 ACM Symposium on SDN Research*, SOSR '19, pages 21–28, New York, NY, USA. Association for Computing Machinery.

[Jin et al. 2017] Jin, X., Li, X., Zhang, H., Soulé, R., Lee, J., Foster, N., Kim, C., and Stoica, I. (2017). Netcache: Balancing key-value stores with fast in-network caching. SOSP '17.

[Kang et al. 2016] Kang, M. S., Gligor, V. D., Sekar, V., et al. (2016). Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks. In *NDSS*, volume 1, pages 53–55.

[Khurshid et al. 2013] Khurshid, A., Zou, X., Zhou, W., Caesar, M., and Godfrey, P. B. (2013). {VeriFlow}: Verifying {Network-Wide} invariants in real time. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 15–27.

[Kim et al. 2006] Kim, Y., Lau, W. C., Chuah, M. C., and Chao, H. J. (2006). Packetscore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing*, 3(2):141–155.

[Klimburg-Witjes and Wentland 2021] Klimburg-Witjes, N. and Wentland, A. (2021). Hacking humans? social engineering and the construction of the "deficient user" in cybersecurity discourses. *Science, Technology, & Human Values*, 46(6):1316–1339.

[Kohler et al. 2018] Kohler, T., Mayer, R., Dürr, F., Maaß, M., Bhowmik, S., and Rothermel, K. (2018). P4cep: Towards in-network complex event processing. In *Proceedings of the 2018 Morning Workshop on In-Network Computing*, NetCompute '18, page 33–38, New York, NY, USA. Association for Computing Machinery.

[Kreutz et al. 2013] Kreutz, D., Ramos, F. M., and Verissimo, P. (2013). Towards secure and dependable software-defined networks. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, pages 55–60, New York, NY, USA. ACM.

[Lapolli et al. 2019] Lapolli, Â. C., Marques, J. A., and Gaspary, L. P. (2019). Offloading real-time ddos attack detection to programmable data planes. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 19–27. IEEE.

[Le and Zincir-Heywood 2020] Le, D. C. and Zincir-Heywood, A. N. (2020). A frontier: Dependable, reliable and secure machine learning for network/system management. *J. Netw. Syst. Manag.*, 28(4):827–849.

[Liberato 2022] Liberato, M. (2022). Secbert: Analyzing reports using bert-like models. Master's thesis, University of Twente.

[Lopes et al. 2016] Lopes, N., Bjørner, N., McKeown, N., Rybalchenko, A., Talayco, D., and Varghese, G. (2016). Automatically verifying reachability and well-formedness in p4 networks. *Technical Report, Tech. Rep*.

[Lopes et al. 2015] Lopes, N. P., Bjørner, N., Godefroid, P., Jayaraman, K., and Varghese, G. (2015). Checking beliefs in dynamic networks. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 499–512, Oakland, CA. USENIX Association.

[Lupu and Sloman 1999] Lupu, E. C. and Sloman, M. (1999). Conflicts in policy-based distributed systems management. *IEEE Trans. Softw. Eng.*, 25(6):852–869.

[Mace and Fonseca 2018] Mace, J. and Fonseca, R. (2018). Universal context propagation for distributed system instrumentation. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, pages 8:1–8:18, New York, NY, USA. ACM.

[Macías et al. 2020] Macías, S. G., Gaspary, L. P., and Botero, J. F. (2020). Oracle: Collaboration of data and control planes to detect ddos attacks. arXiv cs.NI 2009.10798.

[Marnerides et al. 2011] Marnerides, A., James, C., Schaeffer-Filho, A., Sait, S., Mauthe, A., and Murthy, H. (2011). Multi-level network resilience: Traffic analysis, anomaly detection and simulation. *ICTACT Journal on Communication Technology, Special Issue on Next Generation Wireless Networks and Applications*, 2:345–356.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74.

[Mirkovic et al. 2002] Mirkovic, J., Prier, G., and Reiher, P. (2002). Attacking ddos at the source. In *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, pages 312–321. IEEE.

[Mitnick and Simon 2003] Mitnick, K. D. and Simon, W. L. (2003). *The art of deception: Controlling the human element of security*. John Wiley & Sons.

[Moshref et al. 2013] Moshref, M., Yu, M., and Govindan, R. (2013). Resource/accuracy tradeoffs in software-defined measurement. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 73–78.

[Moshref et al. 2016] Moshref, M., Yu, M., Govindan, R., and Vahdat, A. (2016). Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 129–143.

[Mustard et al. 2019] Mustard, C., Ruffy, F., Gakhokidze, A., Beschastnikh, I., and Fedorova, A. (2019). Jumpgate: In-Network processing as a service for data analytics. In *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, Renton, WA. USENIX Association.

[Neves et al. 2021] Neves, M., Huffaker, B., Levchenko, K., and Barcellos, M. (2021). Dynamic property enforcement in programmable data planes. *IEEE/ACM Transactions on Networking*, 29(4):1540–1552.

[Nijland 2022] Nijland, J. (2022). Gamification of cyber security awareness training for phishing against university students. B.S. thesis, University of Twente.

[Peng et al. 2004] Peng, T., Leckie, C., and Ramamohanarao, K. (2004). Proactively detecting distributed denial of service attacks using source ip address monitoring. In *International conference on research in networking*, pages 771–782. Springer.

[Phaal 2009] Phaal, P. (2009). sflow: Sampling rates. In *June 2009*.

[Pinheiro 2020] Pinheiro, P. P. (2020). Segurança digital: Proteção de dados nas empresas. *1ª edição. São Paulo, SP: Grupo GEN*.

[Ports and Nelson 2019] Ports, D. R. K. and Nelson, J. (2019). When should the network be the computer? In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, page 209–215, New York, NY, USA. Association for Computing Machinery.

[Qin et al. 2020] Qin, Q., Poularakis, K., Leung, K. K., and Tassiulas, L. (2020). Line-speed and scalable intrusion detection at the network edge via federated learning. In *2020 IFIP Networking Conference (Networking)*, pages 352–360.

[Sanvito et al. 2018] Sanvito, D., Siracusano, G., and Bifulco, R. (2018). Can the network be the ai accelerator? In *Proceedings of the 2018 Morning Workshop on In-Network Computing*, NetCompute '18, pages 20–25, New York, NY, USA. Association for Computing Machinery.

[Sapio et al. 2017] Sapio, A., Abdelaziz, I., Aldilaijan, A., Canini, M., and Kalnis, P. (2017). In-network computation is a dumb idea whose time has come. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, page 150–156, New York, NY, USA. Association for Computing Machinery.

[Sapio et al. 2021] Sapio, A., Canini, M., Ho, C.-Y., Nelson, J., Kalnis, P., Kim, C., Krishnamurthy, A., Moshref, M., Ports, D., and Richtarik, P. (2021). Scaling distributed machine learning with In-Network aggregation. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 785–808. USENIX Association.

[Sflow 2017] Sflow (2017). sflow.org - making the network visible. In *Accessed on June, 29 2017*.

[Shafahi et al. 2016] Shafahi, M., Kempers, L., and Afsarmanesh, H. (2016). Phishing through social bots on twitter. In *2016 IEEE International Conference on Big Data*, pages 3703–3712. IEEE.

[Sigelman et al. 2010] Sigelman, B. H., Barroso, L. A., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., Jaspan, S., and Shanbhag, C. (2010). Dapper, a large-scale distributed systems tracing infrastructure. Technical report, Google, Inc.

[Stoenescu et al. 2016] Stoenescu, R., Popovici, M., Negreanu, L., and Raiciu, C. (2016). Symnet: Scalable symbolic execution for modern networks. In *ACM SIGCOMM 2016*, pages 314–327. ACM.

[Udupi et al. 2007] Udupi, Y. B., Sahai, A., and Singhal, S. (2007). A classification-based approach to policy refinement. In *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pages 785–788. IEEE.

[Verma 2002] Verma, D. C. (2002). Simplifying network administration using policy-based management. *IEEE network*, 16(2):20–26.

[Westerinen et al. 2001] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and Waldbusser, S. (2001). Terminology for policy-based management. Technical report.

[Winder 2020] Winder, D. (2020). Much of the internet went down yesterday: Here's the reason why. *Forbes*.

[Xie et al. 2019] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Liu, Y. (2019). A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(1):393–430.

[Xiong and Zilberman 2019a] Xiong, Z. and Zilberman, N. (2019a). Do switches dream of machine learning? toward in-network classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, pages 25–33, New York, NY, USA. Association for Computing Machinery.

[Xiong and Zilberman 2019b] Xiong, Z. and Zilberman, N. (2019b). Do switches dream of machine learning? toward in-network classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, page 25–33, New York, NY, USA. Association for Computing Machinery.

[Yu et al. 2011] Yu, M., Greenberg, A., Maltz, D., Rexford, J., Yuan, L., Kandula, S., and Kim, C. (2011). Profiling network performance for multi-tier data center applications. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*.

[Zargar et al. 2013] Zargar, S. T., Joshi, J., and Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069.