

A Survey on Enterprise Network Security: Asset Behavioral Monitoring and Distributed Attack Detection

Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman

Abstract—Enterprise networks that host valuable assets and services are popular and frequent targets of distributed network attacks. In order to cope with the ever-increasing threats, industrial and research communities develop systems and methods to monitor the behaviors of their assets and protect them from critical attacks. In this paper, we systematically survey related research articles and industrial systems to highlight the current status of this arms race in enterprise network security. First, we discuss the taxonomy of distributed network attacks on enterprise assets, including distributed denial-of-service (DDoS) and reconnaissance attacks. Second, we review existing methods in monitoring and classifying network behavior of enterprise hosts to verify their benign activities and isolate potential anomalies. Third, state-of-the-art detection methods for distributed network attacks sourced from external attackers are elaborated, highlighting their merits and bottlenecks. Fourth, as programmable networks and machine learning (ML) techniques are increasingly becoming adopted by the community, their current applications in network security are discussed. Finally, we highlight several research gaps on enterprise network security to inspire future research.

Index Terms—Enterprise network security, networked asset monitoring, distributed network attack detection

I. INTRODUCTION

Enterprises such as universities and research institutes host critical data and offer publicly accessible services through their networks. Thus, they often become popular targets of distributed network attacks that actively probe asset vulnerabilities and paralyze their services. With practical defense appliances (e.g., firewalls and intrusion detection systems) employed by IT departments of enterprises, network attacks are becoming well distributed in sources and agile in attacking patterns to bypass such static detection and increase their effectiveness. To be more specific, a sophisticated network attack usually employs hundreds and thousands of botnet devices spread across geolocations and diversified in types (e.g., Internet-of-Things, laptops, and compromised servers); each may send malicious traffic with changing patterns and protocols. Some popular and large-scale DDoS attacks [26] include, but are not limited to: Amazon AWS became the target of a massive Terabits-level DDoS attack sourced from hijacked CLDAP servers in 2020; Github suffered from a Memcached protocol-based DDoS attack in 2018; during Rio 2016 Summer

Olympics, critical servers of official Olympics organizations as well as Brazilian banks and telcos [132]) were targeted by sustained distributed network attacks with mixed traffic types such as TCP-SYN, UDP reflection, DNS, CHARGEN (character generator protocol), NTP, and SSDP sourced from millions of compromised devices (e.g., IoTs) across the globe [138]. Successful distributed network attacks lead to service failures, disruptions, and reputation degradation.

Distributed attacks on enterprise networks often consist of two phases, namely reconnaissance attacks (also known as scans) to discover the vulnerability of networked assets and distributed denial-of-service (DDoS) attacks that paralyze the targeted victims that are discovered by malicious actors. To cope with the threats, enterprise IT departments are expected to track the devices within their networks to ensure their expected behaviors and enforce attack defense mechanisms that can effectively detect and mitigate attacks on their networked assets without impacting legitimate communications.

There are many mature products for monitoring the network behaviors of enterprise assets and providing protections against distributed attacks via static configurations, such as next-generation-firewall (NGFW) appliances and intrusion detection systems (IDS). These static solutions are practical to be used in high-throughput enterprise networks. Still, they are ineffective in providing precise results (e.g., differentiating distributed attackers and malicious flows from their benign counterparts). Therefore, it is not surprising that the consequential attack mitigation measures (e.g., randomly dropping packets to the victim) introduce non-negligible collateral damages on benign traffic [31]. For instance, typical next-generation-firewalls (NGFW) require users to configure rules that specify the list of focused enterprise assets and the corresponding defense strategies. Such methods effectively protect certain critical assets by tracking their network activities of several traffic types but fail to capture unknown and complex threats from hosts operated by sub-departments, staff, and visitors. Moreover, the static nature of such methods limits their capabilities in detecting emerging attacks with dynamic and stealthy traffic patterns [165].

Legacy static solutions introduce blind spots likely exploited by malicious actors and agile attackers. Research communities have developed dynamic telemetry methods for network monitoring via flow-level statistics and networked graph structures to address this problem. Those methods can provide fine-grained statistics to track each network flow between enterprise assets and external hosts without leaving any blind spot. How-

M. Lyu, H. Habibi Gharakheili and V. Sivaraman are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mails: minzhao.lyu@unsw.edu.au, h.habibi@unsw.edu.au, vijay@unsw.edu.au).

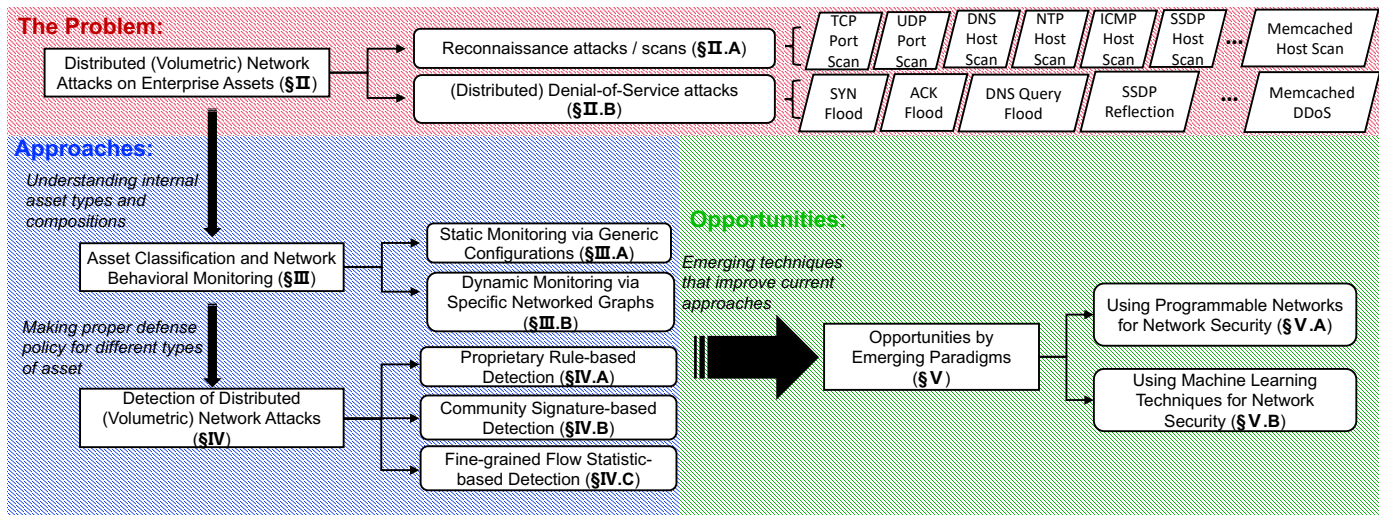


Figure 1: Key topics covered in this survey.

ever, maintaining fine-grained flow-level telemetry unavoidably introduces high computational overheads. Therefore, they are not scalable for large enterprise networks with hundreds and thousands of hosts that exchange millions of concurrent flows.

Recent developments in two emerging paradigms, namely Programmable Networks and Machine Learning (ML), offer promise to improve the flexibility of network monitoring and accuracy of attack detection. Generally speaking, programmable networking covers two main areas: network function virtualization (NFV) and software-defined networking (SDN). It changes the static nature of network traffic processing often carried out by proprietary legacy hardware and middleboxes. Instead, dynamic network functions on generic servers and programmable switches are used to achieve high responsiveness and real-time orchestrations. Researchers have leveraged this technology to overcome the challenges of legacy network monitoring and protection in various use cases, such as real-time defense orchestration for ISP network [41] and elastic control of virtual firewalls [30]. These inspire the development of solutions to the current problems of enterprise network security. On the other hand, recent advances in ML techniques that help obtain data-driven models to make accurate predictions on statistical attributes have proven their supremacy in many disciplines, such as computer vision and language recognition. Despite some of the practical challenges in applying ML methods to network security [133], researchers have successfully employed ML algorithms to make reliable security inferences from various types of network telemetry (e.g., system logs or packet headers) in a variety of scenarios (e.g., IoT attack or SSH brute-forcing). We believe that their trials and efforts provide us with valuable lessons to address issues in asset classification and attack detection accurately and precisely.

This survey systematically reviews related research articles and industry practices, providing comprehensive insight into current developments, challenges, and future directions of asset management and distributed attack detection in enterprise

network security. Unlike prior surveys that broadly studied certain attack types and defense mechanisms, we focus on a narrow aspect of distributed volumetric network attacks and their countermeasures applicable to enterprise networks. In addition, we review the potential and challenges of improving the state-of-the-art in two emerging paradigms (*i.e.*, programmable networks and ML). To this end, we summarize the main topics covered by this survey as follows, which are also visually shown in Fig. 1. **First**, in §II, we highlight the diversity and variety of distributed network attacks including reconnaissance scans and distributed denial-of-service (DDoS) attacks; **second**, in §III, we discuss the current development of enterprise networked asset classification and behavioral monitoring via either static or dynamic methods; **third**, in §IV, enterprise distributed attack detection systems using proprietary rules, community signatures, and fine-grained flow statistics are surveyed; **fourth**, in §V, opportunities introduced by the two emerging paradigms, *i.e.*, flexibility by programmable networks and accuracy by machine learning are discussed as to inspire future researches. Relevant surveys (but on other aspects of network security) are discussed in §VII. We highlight several research gaps as valuable future directions in §VI, and conclude this survey in §VIII.

II. DISTRIBUTED NETWORK ATTACKS ON ENTERPRISE ASSETS

Network attacks that probe, congest, or paralyze enterprise assets such as public-facing servers are becoming distributed in sources, versatile in traffic patterns, and diverse in underlying mechanisms [102], [79], [3]. Such attacks often occur sequentially – an enterprise asset is first examined for its availability and known vulnerabilities through a reconnaissance attack (*i.e.*, host or port/service scans), followed by (distributed) denial-of-service (DoS or DDoS) attacks.

Large-scale scans and denial-of-service are often conducted in a distributed manner from a single source to (a) increase their effectiveness and (b) make it difficult for defense systems to detect and/or mitigate. Distribution is typically achieved by

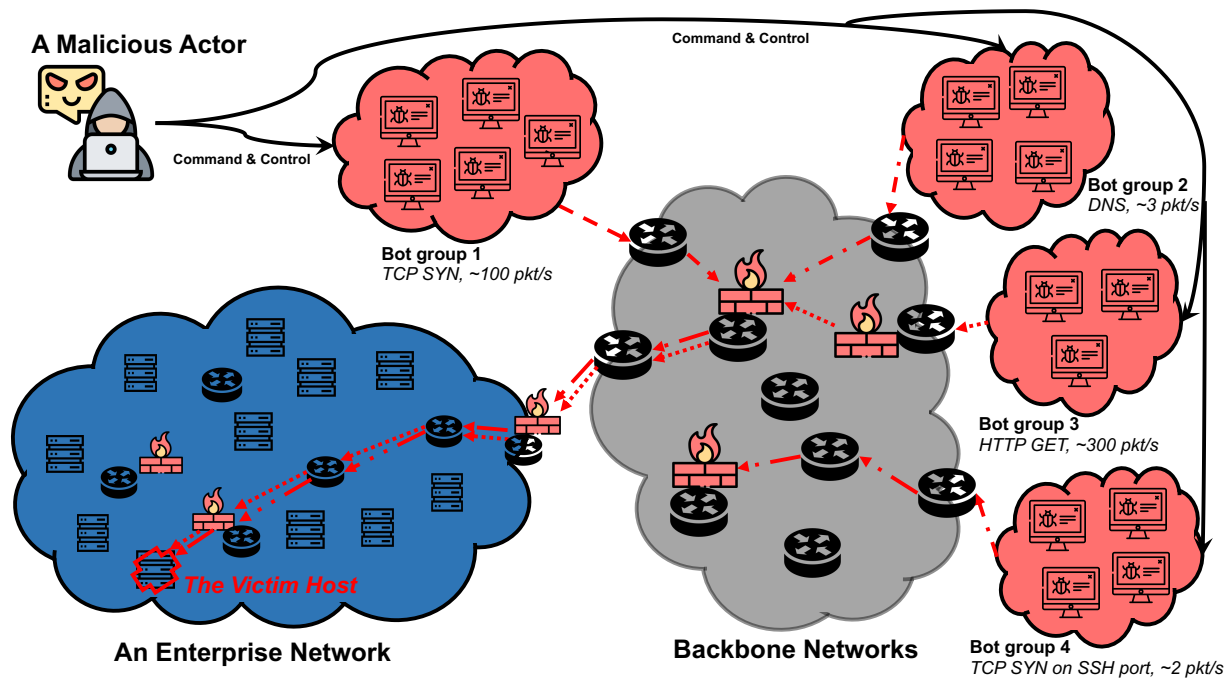


Figure 2: A visual example of distributed network attacks on a victim inside an enterprise.

recruiting botnets, consisting of massive compromised devices like personal computers, powerful workstations, public-facing servers, or compromised IoT appliances [44], [54], [140], [142], [105]. To avoid detection, malicious actors often split an attack into small segments, each performed by a single bot device. For example, in a powerful but stealthy DDoS attack, each bot device only generates low-rate traffic across a variety of protocols [56], making it difficult to be distinguished from benign instances. It is practically challenging to precisely identify all attack sources [1] and block them. We show a visual example of distributed network attacks in Fig. 2, where a malicious actor commands and controls four distributed bot groups to attack a victim residing within an enterprise network. Each group uses a different traffic type and rate so that a reasonable fraction of the entire attack traffic (sent by bot groups 1 and 2 in Fig. 2) can successfully bypass defense appliances. Note that most commercial firewalls operational in backbone and enterprise networks are not optimally tuned to detect stealthy malicious traffic on the path before it hits the victim.

A. Reconnaissance Attacks

Malicious actors use reconnaissance attacks (also known as scans) to construct their knowledge of targeted hosts and services (ports). Those attacks probe the availability of enterprise-connected hosts and discover their potential vulnerabilities [143]. The discovered hosts may not only become victims but may also be exploited as attack amplifiers/reflectors to paralyze other victims. For example, a discovered NTP server with high reflection capability (*i.e.*, generate response packets with a size larger than that of the received requests) can be used to amplify attack volume in reflection-based DDoS attacks [78], [94].

Apart from malicious purposes, security researchers also develop tools to identify potential cyber threats enterprises face, such as open ports and vulnerable services that could be exploited in network attacks. For example, *Nmap* [113] is developed as a comprehensive scanning tool to discover active hosts and ports (*i.e.*, services). To increase the speed and effectiveness of scans, the authors of *Zmap* [35] optimized the scanning process by tuning the probing rate, pre-connection state, and re-transmission, which can probe the entire IPv4 space within 45 minutes. Scanning techniques have evolved to become scalable at 10 Gbps throughput [2] and can perform vulnerability scans towards protocol banners through user queries [33]. Reconnaissance attacks have also been studied for certain scan types, such as critical cyber-physical infrastructures [161] and DNS utilities [59].

To combat reconnaissance attacks, researchers have developed methods, such as tracking port scanners on the IP backbone [136], detecting subtle port scanning via interactive visualization [154], disrupting reconnaissance attacks via address mutation [64], constructing distributed network telescope to capture scanners [121], and optimizing backscatter [45] technique for scan detection in massive IPv6 address space [46]. However, according to [57], [34], only a few enterprises have practically adopted robust defensive measures. Thus, service and host scans are still prevalent on the Internet, exposing service and device vulnerabilities (*e.g.*, Linksys routers, OpenSSL, and NTP). Consequently, exposed hosts may be exploited by malicious actors on the Internet to generate/reflect attacks or become direct victims in the future.

B. Distributed Denial-of-Service Attacks

As already shown in Fig. 2, malicious actors may choose to flood their target victim directly from botnet devices using

various techniques or protocols [56] (*e.g.*, HTTP, ICMP, and TCP-SYN). Also, they may choose to launch a reflection-based DDoS with larger attack volumes. For example, bot devices send packets with the spoofed source IP address (of the ultimate victim) to the discovered reflectors (*e.g.*, DNS and NTP servers); these reflectors will then respond to the victim with larger packet sizes.

DDoS attacks are becoming more complex, distributed, and agile. The existing research literature extensively studied the characteristics of various DDoS attacks. First, according to [53], DDoS attacks are becoming complex in protocols and traffic types. The participant botnets are likely to be independent. Such patterns make it challenging for defenders to isolate malicious traffic and attack sources. Second, the increasing adoption of cyber-physical devices (*e.g.*, IoTs) brings new vulnerabilities and expands attack surfaces yet to be addressed [89]. Therefore, a growing number of IoT devices connected to the Internet are compromised as a botnet, enabling more powerful and frequent DDoS attacks on a global scale [156]. For example, in late 2016, Mirai [95], an IoT malware that hijacked hundreds of thousands of IoT devices, has led to unprecedented DDoS attacks globally. During an attack, each compromised IoT device generates malicious traffic at a low rate, making them hard to be differentiated from benign traffic. Third, DDoS attacks are becoming more dynamic and agile in their activity patterns to evade detection. As pointed out in [150], they are usually launched with changing temporal and spatial patterns to bypass detection, which makes them quite effective against static rule-based and signature-based detection methods. Botnets of different families also work collaboratively. A given bot might adapt its attacking strategy provided by different malware families [22]. Finally, the concept of DDoS-as-a-service is becoming popular as it lowers the barrier to generating a distributed attack effectively [73]. Botnet owners can lease their controlled devices for financial benefits, so malicious actors with fewer resources (*e.g.*, controlled bot) can rent their large botnet to launch powerful attacks.

C. Highlights of Distributed Network Attacks

We now summarize three key highlights in this section.

First, network attacks such as DoS and scans are becoming: (a) “*distributed*” by recruiting botnets to generate attack traffic from different logical sources (*e.g.*, ASes, subnets) and physical geolocations, (b) “*complex*” by leveraging a wide range of protocols and vulnerabilities, and (c) “*dynamic*” by shifting active bot groups or traffic patterns randomly. All the above characteristics increase the difficulties in effectively detecting distributed attacks.

Second, potential vulnerabilities of network-connected hosts (*e.g.*, BYOT devices, enterprise servers, or IoTs) may be identified and exploited by malicious scripts (*e.g.*, URLs contained in phishing emails) or malware. Such compromised devices are used as bots to perform further infections within their local network or participate in attacks on other networked assets. Therefore, continuously monitoring network traffic behaviors and enforcing appropriate security management are essential

Table I: Classifying host types in a large enterprise network by DNS names [92].

Asset type	# hosts
Website server	61
Authoritative name server	15
VPN gateway server	13
Remote computing platform	16
File storage server	14
Mail exchange server	18
DNS recursive resolver	7
Web proxy	4
NAT gateway	256
Personal computer and BYOTs	1,961
Other unclassified (minor) types	18,920

for network operators (IT and cyber departments). We will discuss in §III some of the tools and techniques for asset network behavioral monitoring.

Third, apart from malware infections and misuse, assets such as servers and databases within an enterprise network may be direct targets of distributed attacks. During such attacks, public-facing servers may not be able to respond to benign requests of external clients if their networking and computational resources get exhausted. In addition, network vulnerabilities of internal non-critical enterprise hosts may become exposed to external hackers for further cyber-crimes. Therefore, defending against distributed attacks on enterprise assets is critical for security operations. In §IV, we will elaborate on state-of-the-arts enterprise attack detection systems and mechanisms.

III. ASSET CLASSIFICATION AND NETWORK BEHAVIORAL MONITORING

Obtaining real-time visibility into assets and their behaviors is essential to combat the increasing number of distributed network attacks targeting or utilizing enterprise assets. IT¹, OT², and cybersecurity teams need tools to classify connected assets based on their role (*e.g.*, web server, DSN server, camera, personal computer), ensuring asset activities conform to their role’s patterns (profile). With asset profiles clearly modeled, appropriate security policies (*e.g.*, segmentation, access rules) can be applied to the network, and certain attacks can be prevented or at least detected easier.

However, profiling asset behaviors is a nontrivial task as enterprise hosts come with diverse and complex functionalities and behaviors. For example, an enterprise can have servers of various types that serve internal or external clients; visitors and staff may have their personal devices (*e.g.*, mobile phones and laptops) connected through wireless gateways, and IoTs such as smart cameras and sensors may also be installed in a typical enterprise network [130]. Let us take a look at Table I, which lists popular types of networked hosts (top ten rows) identified by their enterprise DNS names in a large university network, studied by work in [92]. As shown by the last row of Table I, there are many other unclassified and less-popular

¹information technology

²operational technology

host types, such as LDAP server and Redis proxy, which are often hard to enumerate. We note that those identifiable assets (by their domain name) are only accountable for less than 10% of active hosts in the enterprise, and the functionality of other 90% hosts is mostly unknown to the university network managers and administrators.

Connecting many heterogeneous devices will inevitably introduce challenges to network management, operation, and security teams. Devices owned and managed by visitors and/or staff may come infected by malware and hence start conducting malicious activities [37] upon arrival to the network, which may go undetected by security tools and appliances [111], [10]. Also, inaccurate access policies and configurations (*e.g.*, public-facing servers) may give external attackers opportunities to compromise less secure internal hosts for malicious purposes. Organizations like universities and research institutions often have relatively unfettered networks, allowing subdivisions and departments to configure their own IT infrastructures. This makes the problem even more pronounced as asset visibility gets relatively poor.

Many solutions have been developed by industry (*e.g.*, [97], [25], [36]) and academia (*e.g.*, [92], [72], [52], [118]) to classify roles and/or monitor network behaviors of individual assets. Existing methods can be categorized into either static configurations/databases that record high-level characteristics (*e.g.*, role/class/model) of the connected devices or dynamic graphs obtained from passive traffic monitoring that capture communication patterns individual networked hosts display.

A. Static Monitoring via Generic Configurations

Current practical solutions for the management and classification of enterprise networked assets primarily rely on static and relatively generic characteristics (*e.g.*, tables containing a list of device hostnames, their VLAN, Operating System, IP address, or perhaps their role) without capturing their behavioral characteristics like what is shown in Fig. 3. For example, firewall appliances are often configured by access control lists (ACL) and rules that keep static information of internal hosts such as VLAN ids, device categories, and/or user groups [114]; DHCP and DNS servers maintain system logs containing device names and their IP addresses [6]; and, other specialized commercial platforms managing enterprise assets are configured by lists supplied (often manually) by the IT department [50].

Ideally, an IT department equipped with full knowledge of assets connected to their network will be able to segment the network and enforce strict access configurations to prevent unintended communications to/from networked devices on the network [25]. Network traffic not conforming to those configurations will be marked as abnormal, thus, triggering further defense actions such as alerts and mitigation. For example, according to best practices of the Microsoft firewall [99], inbound port or service rules could be enforced so that the border firewall blocks all non-HTTPS traffic towards an enterprise HTTPS server or inbound DNS packets are only allowed if their destinations are enterprise DNS servers. To protect a critical asset operational within an enterprise (say,

a corporate website server), the network administrator may choose to set up an upper bound rate limit via its traffic shaper platform for that asset. Whenever the asset receives traffic rates higher than the allowed limit, the management system (*i.e.*, shaper and/or firewall) will partially or fully drop those inbound packets, preventing a potential volumetric attack on that specific host.

Static configurations enable administrators to manage and monitor their enterprise assets by specifying relatively high-level network profiles. However, with the explosive growth of network applications communicating via a variety of protocols in conjunction with the adoption of IoT/OT devices with heterogeneous behaviors, populating and maintaining generic, high-level configurations and policies become increasingly difficult for enterprise IT departments, especially for those with loosely-federated networks [109]. As highlighted in [153], [137], [148], [48], specifying policies for a large enterprise network with complex host composition is error-prone, and potential misconfigurations can impose high operational costs, such as fixing bugs and resolving conflicts. This problem becomes even worse with the adoption of many and diverse BYOTs and IoTs [63]. Therefore, managing assets by simple (generic) configurations tailored for each device type inevitably leaves many blind spots and becomes impractical in most operational settings [157], [12].

Moreover, the behavior of networked hosts in modern enterprises can change in time [92]. For example, certain divisions or departments (engaged in projects with external stakeholders) may operate multiple services (*e.g.*, DNS and website) on a single machine and expose them to the public Internet, each with a distinct behavioral pattern – some services may get terminated, and/or new services or functionalities may be added on-demand. As a result, static methods fall short of expectations [155], [168], [11].

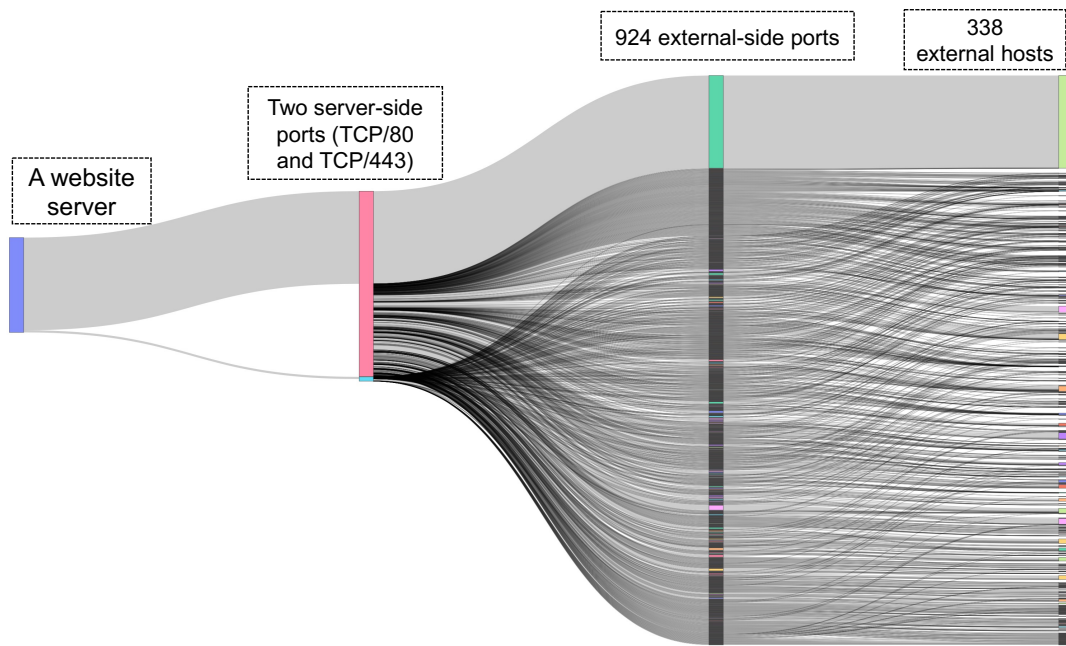
Motivated by some of challenges highlighted above, researchers have developed dynamic methods using specific networked graphs, which will be discussed next.

B. Dynamic Monitoring via Specific Networked Graphs

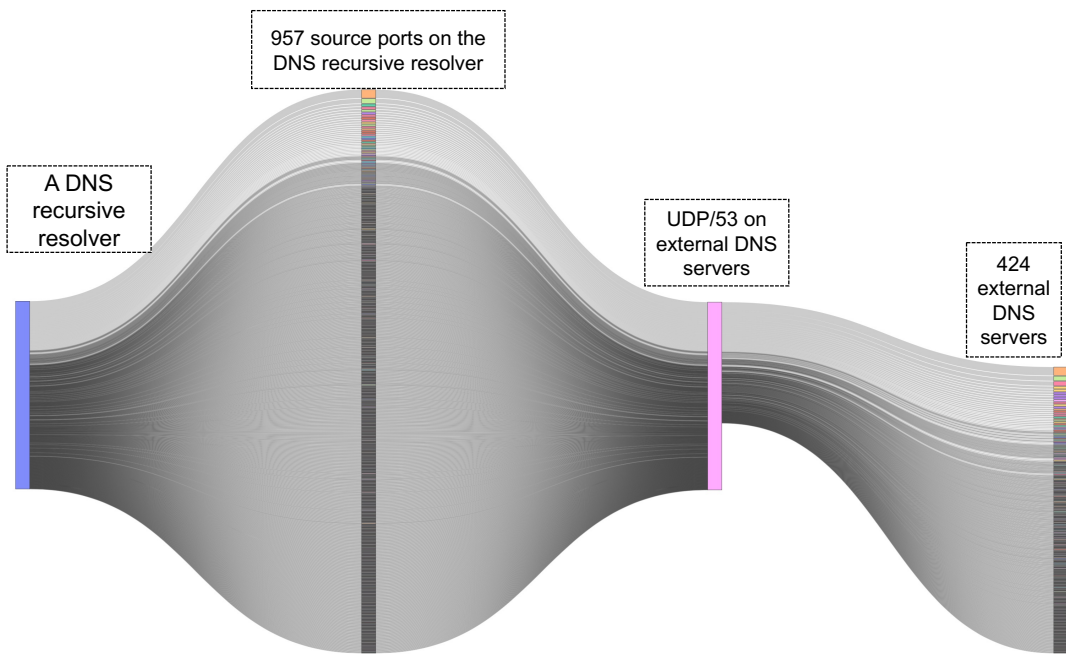
To obtain fine-grained visibility into activities of connected hosts, prior works (will soon be discussed in this section) used networked graphs to characterize (profile) the behavior of various host types. To motivate our discussion, let us consider Fig. 3, which visualizes the flow graphs³ (in the form of Sankey diagrams) of two enterprise hosts connected to a university network (*i.e.*, a website server and a DNS recursive resolver). The website servers often expose only two TCP ports (*i.e.*, TCP/443 and TCP/80) [92] to the public Internet allowing for communication sourced from a wide range of TCP ports by external hosts, while the DNS recursive resolver sends traffic from arbitrary UDP ports targeting only UDP/53 operational on external DNS servers.

1) *Per-Host Classification*: Work in [72] uses graph structures to model network activities of each connected host at the IP address and transport-layer port levels. The authors profiled various types of networked hosts (*e.g.*, HTTP servers, DDOS

³These are constructed from data and models presented in [92].



(a) A website server.



(b) A DNS recursive resolver.

Figure 3: Sankey diagrams illustrating network behavioral profiles of two representative enterprise assets: (a) a website server, and (b) a DNS recursive resolver, using 1000 flows of each networked asset for visualization purpose.

attackers, and P2P clients), each with a unique transport-layer behavioral pattern. For example, the graph pattern of an FTP server consists of a large number edges destined to IP addresses, initiated from a wide range of port numbers connecting to two popular port numbers, namely TCP/20 and TCP/21 on the server. The authors developed a method to classify an unknown host by checking the similarity between its behavioral graph with that of known types. To effectively monitoring communication patterns of hosts, work [52] developed a tool that visualizes communication graphs for network

operators helping them classify networked entities manually. To fill the gap in legacy graphs that only capture flow profiles, researchers have developed relatively advanced graph structures that can model network communications with more descriptive features, such as attributed graph models in [118]. This graph structure is generated to capture both network topological properties (*e.g.*, connections between nodes) and correlated attributes on each graph edge which hold both computational efficiency through sampling techniques used in graph generation and accuracy when classifying host roles in

real-world networks.

2) *Clustering Hosts and Modelling Group Interactions:* Modeling the behavior of individual hosts can be challenging, especially at scale. Therefore, one may choose to reduce the dimension by focusing on clusters/groups of hosts. In such clustered graphs, hosts are often grouped and represented by their common communications behaviors, such as contacting a similar range of external hosts or residing in the same subnet. However, balancing the level of aggregation and visibility into the actual network often requires extensive tuning and optimization. In [15], the authors optimized communication graphs for a large network to achieving the optimal consumption of computational resources while having sufficient information captured in the graph to describe the evolution of a network attack. Authors of [69] used the method named “connection graph analysis” to discover cooperating hosts in P2P networks, which start from a single known P2P node in a network to discover and group other associated hosts progressively. The developed method was demonstrated to have short processing times in grouping all P2P hosts in large networks by processing their NetFlow streams. Besides, statistical methods such as clustering algorithms are quite powerful for grouping and differentiating hosts based on their behavioral profiles in large-scale networked graphs. As an example, in [160], the authors applied clustering algorithms to effectively identify groups of hosts that inherently belong to different application types on bipartite graphs describing communications between hosts. As in [65], the authors clustered hosts within the same enterprise network that have strong inter-IP connectivity (*i.e.*, connecting to a similar range of hosts) for enterprise IT departments so that they could track the behavior of each identified group instead of individual hosts for scalability in network management.

3) *Detecting Host Anomaly from Graphs:* Using networked graphs that describe host interconnectivity can be particularly useful for cybersecurity applications such as to detect malicious hosts in a network or identify clusters of botnet devices launching distributed attacks [119], [152], [7], [103], [10], [39], [71], [62]. For some recent examples, the work in [10] considers a large enterprise network with dynamic compositions and communication patterns of hosts, and hence becomes difficult to manage and secure. Therefore, the authors developed a probabilistic graph model to measure the success rate of an attack on a given network topology so that IT departments could optimize their attack detection policies and fix vulnerable network configurations. *SpotLight* [39] achieved accurate and responsive detection of anomalies in high-density graphs for IP communications near real-time. The anomalies (*e.g.*, port scans and DoS) are identified by the sudden changes in subgraphs consisting of a subset of nodes and edges from the networked graph. The authors leveraged randomized sketching algorithms to make cost-effective inferences with optimal memory consumption. Similarly, *NoraCle* [71] detect anomalous behavioral changes of individual hosts in network graphs using stochastic block models, which could detect hosts with deviated behaviors (*e.g.*, connecting to unusual hosts) compared with other hosts in the same cluster. Whereas *TRACE* [62] builds a distributed enterprise-wide communica-

tions graph tracking information from both network connectivity (*e.g.*, IP address and port number) and involved device system calls (*e.g.*, application name and process ID) between enterprise hosts for advanced persistent threat (APT) detection.

C. Highlights

In summary, configuring static policies on middleboxes like firewalls is the de facto method by the current industry practices for managing (selected) networked assets. Such methods are practical computationally, as they often maintain lightweight data structures for specific groups of managed entities, those with critical values to and/or functions for an enterprise. This method prioritizes practical deployment but makes it difficult to gain fine-grained visibility (*e.g.*, at the flow level) and effectively classify host behaviors that are often dynamic or unknown to IT departments.

On the other hand, dynamic monitoring with specific networked graphs is proven to be effective in providing comprehensive visibility into network traffic so that IT departments can effectively classify connected assets and detect potential anomalies. However, using complex graphs incurs high computational costs that make such methods impractical for deployment in large enterprise networks with many diverse, active hosts and concurrent communication flows.

IV. DETECTING DISTRIBUTED NETWORK ATTACKS ON ENTERPRISE HOSTS

Detecting distributed attacks (*i.e.*, DDoS and reconnaissance) is critical for enterprise network operations. To date, the cybersecurity research community has developed solutions to detect various distributed network attacks. For those attacks aiming to congest the Internet link of an enterprise network by sending Gbps or even Tbps malicious traffic to enterprise hosts, handling the attack at ISP levels (close to source and in-transit) appears to be the most effective option [164], [100], [101], [112], [41]. For distributed attacks targeting certain enterprise assets, which is the focus of this survey, detection mechanisms employed by the target enterprise (close to victim/destination) are proven to be more effective [164]. Therefore, enterprise IT departments usually set up inline security middleboxes near their network edges, sitting in between their internal private network and the public Internet. To this end, monitoring and/or detection policies can be developed and enforced for, say, each of the critical enterprise servers [149] that is attractive to potential attackers. Such detection solutions, typically employed by enterprise IT departments, can be categorized into three types: proprietary rules, community signatures, or (flow-level) statistical models, which are comprehensively reviewed in this section.

A. Proprietary Rule-Based Detection

Rule-based distributed attack detection, which allows users to configure their security policies from a list of rules defined by the appliance manufacturer or developer, is widely used by the enterprise security industry.

Zone Protection Profile

Name: ZP_drop_extra

Description: alert

Active Tab: Flood Protection

Scan	Enable	Action	Interval (sec)	Threshold (events)
Host Sweep	<input checked="" type="checkbox"/>	block	10	100
TCP Port Scan	<input checked="" type="checkbox"/>	block	2	100
UDP Port Scan	<input checked="" type="checkbox"/>	block	2	100

Buttons: OK, Cancel

(a) Reconnaissance attack.

DoS Protection Profile

Name: DdosProtection_RED

Description:

Type: Aggregate Classified

Active Tab: Flood Protection

Sub-Tab: SYN Flood

SYN Flood

Action: Random Early Drop

Alarm Rate (packets/s): 44000

Activate Rate (packets/s): 48500

Max Rate (packets/s): 63000

Block Duration (s): 10

Buttons: OK, Cancel

(b) DDoS – SYN flood.

DoS Protection Profile

Name: dosP_drop

Description: v

Type: Aggregate Classified

Active Tab: Flood Protection

Sub-Tab: UDP Flood

UDP Flood

Alarm Rate (packets/s): 10000

Activate Rate (packets/s): 10000

Max Rate (packets/s): 40000

Block Duration (s): 300

Buttons: OK, Cancel

(c) DDoS – UDP flood.

Figure 4: Firewall configurations available for distributed network attack protection (*i.e.*, detection and mitigation): (a) reconnaissance/scan protection, (b) SYN flood DDoS protection, and (c) UDP flood DDoS protection.

1) *Thresholds in Commercial Appliances:* Proprietary appliances such as next-generation-firewall (NGFW), typically deployed at the border of enterprise networks, use threshold-based mechanisms for detecting attacks. Network administrators configure rules to govern access policies of certain networked hosts. Each rule may specify thresholds on traffic volume (*e.g.*, packet rates to specified IP addresses) to distinguish normal and/or abnormal communications. In Fig. 4, we show three screenshots of configuration pages on a commercial firewall appliance, often used in large-scale networks. It can be seen how defensive (default) rules against reconnaissance attacks (Fig. 4(a)), DDoS attacks via SYN flood (Fig. 4(b)), and DDoS attacks via UDP flood (Fig. 4(c)), are configured. For the reconnaissance protection, shown in Fig. 4(a), the network administrator who wants to protect their assets from

host reconnaissance or port scans may set up a security rule to block all external IP addresses that send more than 100 packets to intended hosts within a specified interval (say, 2 or 10 seconds). For protecting against DDoS via SYN flood in Fig. 4(b), the administrator is able to configure thresholds on the packet rate of inbound TCP-SYN toward certain IP zones – exceeding thresholds indicates volumetric anomalies, thereby triggering alerts or actions. Likewise, Fig. 4(c) shows similar detection and mitigation thresholds configured for UDP-based DDoS attacks.

Legacy proprietary middleboxes, enabling admin-configured rules, have been widely deployed by the industry for distributed attack detection/mitigation. Through, these methods are quite simple for adoption and relatively effective for certain attacks types, they are insufficiently flexible to fulfil emerging

security needs like detecting distributed attack sources with versatile traffic patterns. The absent standard way of configuring rules and policies across security appliances sourced from diverse manufacturers and a lack of full compatibility between commercial vendors will introduce practical challenges to operators of multi-vendor networks. It becomes difficult for them to effectively apply their detection logic across appliances, each protecting parts of their network [16].

2) *Experimenting with Expressive Queries*: Given some problems discussed above, particularly a lack of flexibility in configuring rules, researchers [106], [47], [55] employed programmable networking techniques to prototype an expressive query-based middlebox that allows for configuring reactive rules (essentially thresholds-based) at run-time. For instance, to realize a sustainable and versatile attack detection mechanism, particularly in fast-changing environments, the authors of *Marple* [106] designed a query language to perform monitoring tasks via key-value store primitives on programmable P4 switches. To make a rule-based security mechanism effective in combating sophisticated cyber-attacks, involving various logical steps and targeting a large number of network entities, *SAQL* [47] was developed as a stream-based query system that provides an anomaly query engine that allows users to specify their complex detection logic using domain-specific languages. By leveraging both programmable P4 switches and software stream processors, *Sonata* [55] was proposed as a network telemetry system that is scalable and expressive in performing security tasks (e.g., detection of SSH brute force, port scan, DDoS, or Slowloris attacks) with fewer configurations compared to prior relevant systems. Although those research ideas still have a long way to go before being fully adopted by the industry, they are valuable steps toward realizing a low-cost, easy-to-upgrade, and expressive rule-based detection system.

3) *Performance Evaluation and Rule Optimization*: While rule-based security systems are relatively prevalent across the computer networking industry, configuring effective and error-free specifications requires expert administrators with sufficient domain knowledge, as well as complete visibility into connected assets on their networks, without which they can hardly set up effective thresholds, queries, or take appropriate actions. Moreover, manually managing configurations can be challenging for medium to large enterprise networks with complex host compositions and behaviors, particularly in handling rule redundancies, logical conflicts, and configuration errors.

Optimizing the placements of security rules and identifying potential redundancies have received extensive attention from researchers. The work in [90] conducted experiments (e.g., with the number of rules and their complexity) to evaluate the performance degradation in latency and bandwidth that may be caused by placing firewall policies at various security levels. The authors concluded that the placement of firewall rules can have significant impacts on metrics such as latency and throughput, thus, optimization of firewall technologies is critical in reducing performance losses. The authors of [158] conducted a quantitative analysis of rule sets and configuration errors available on a commercial firewall in produc-

tion, highlighting that corporate firewalls are often improperly configured, which prevents them from providing sufficient security protection. Although vendors supply templates and guidelines, network administrators often face challenges in manually selecting and efficiently adopting those templates for their networks. To better understand the performance impacts of rule-based firewalls, the authors of [123] developed a queuing model with a Markov chain to model key performance metrics of firewalls when handling normal or DoS traffic flows. Work in [151], [5] extensively studied performance bottlenecks such as CPU and memory usage under network conditions such as varying traffic rates, packet sizes, and the number of communication flows.

With operational challenges and performance bottlenecks associated with rule-based solutions discussed above, various optimization methods have been employed. Legacy firewalls check each received packet against individual existing rules. Therefore, increasing the number of firewall rules will unavoidably lead to larger processing time. The work in [104] proposed a data mining approach to predict hit probabilities of mutually exclusive rules so that they could be ordered based on their popularities, significantly reducing the processing time up to 40%. To tackle rule redundancies, the authors of [146] proposed an optimization algorithm to locate and reduce redundant rules configured on an enterprise firewall. Work in [141] designed a stateful firewall architecture that can classify network traffic according to their application types; each is mapped to a customized processing pipeline to achieve better performance in terms of CPU utilization, throughput, and queuing delay. Work in [82] developed a hash-based packet classification algorithm to significantly reduce the delays caused by the rule-matching process on a typical firewall appliance. Although a handful of prior research works exist on optimally managing errors and performance degradation introduced by redundant firewall rules (manually configured), rule-based firewall performance issues are still key concerns yet to be solved [38].

B. Community Signature-Based Detection

With the increasing complexity of attack vectors, enforcing effective security rules by administrators has become more challenging than ever. To ease this pain point, the security community developed various software intrusion detection systems (e.g., Bro [117] and Snort [122]) that do not require complex configurations. Instead, users could simply import security signature files containing fingerprints of malicious traffic characterized and made publicly available by security experts and/or researchers.

1) *Merits*: Unlike rule-based detection via proprietary systems, signature-based attack detection typically leverages software engines (CPU-based computing) that support highly flexible traffic processing functions. In addition, software-based intrusion detection systems (IDS) are relatively attractive because various functionalities can be customized by network/security admins without tedious negotiations with vendors to upgrade hardware appliances. As highlighted in [165], hardware appliances are designed for high performance

(e.g., sustained Tbps traffic) and thus sacrifice operational flexibility in dynamic network environments. At the same time, software-based systems can overcome those limitations by elastically scaling or replacing detection functions based on operational needs and traffic composition.

2) *Current Issues*: Despite certain advantages software signature-based IDSes offer, there are practical challenges in using these tools in operation. First, generating quality signatures for diverse attacks can be nontrivial and time-consuming, requiring expert (and expensive) man powers. Second, given the attack surface of various networks can be quite different, signatures developed by third parties may not be readily (and directly) applicable to every enterprise network – yielding poor efficiency. Third, such software tools hardly scale cost-effectively to process high traffic rates. To address three problems, researchers developed methods for the automatic generation of signatures, increasing the efficacy of detection, and improving the scalability of software IDSes.

Automatic Generation of Signatures: Many research works attempted to develop automatic methods for generating reliable attack signatures. Work in [77] presented a system to automatically generate signatures needed for pattern matching and protocol conformance checks. The authors set up honeypots to passively capture malicious network traffic. To evade getting matched against known signatures, attackers may try to craft the payload contents of their malicious packets. To defend against those sophisticated attacks, *Polygraph* [108] was proposed to automatically generate signatures that contain multiple disjoint content sub-strings for polymorphic worms (i.e., an example attack that varies its packet payloads frequently). *AutoRE* [159] focuses on detecting those botnets that send spam emails. The authors aimed at avoiding allowlists which can be tedious to populate. They instead check whether email payloads contain identifiable malicious patterns URLs and look for distributed destination and/or bursty patterns in the email traffic sent.

Detection Effectiveness: In terms of the prediction power of signature-based detection systems, researchers have identified various problems and proposed corresponding solutions. In [116], *S. Patton et al.* highlighted the “Squealing” vulnerability of a signature-based IDS. Given known signatures employed by the IDS in charge, attackers can craft malicious packets that result in high rates of false positives making the alerting system almost unreliable (useless). Authors of [134] observed that the legacy signatures using byte sequences suffered from a high false-positive rate due to the dynamics of attacks. To address this issue, the authors developed a signature engine on the Bro IDS [117] that can generate richer signatures by incorporating factors like the dependency of networking events (e.g., requests and replies). Works in [28] and [8] compared the accuracy and performance of IDS designed for computing environments: single-threaded tools (i.e., Snort) versus multi-threaded tools (i.e., Suricata). They concluded that Suricata gives higher accuracy under a multi-core setup, while Snort achieved fewer false negative alarms within a single-core networking system. Besides, according to [21], [13], the adoption of emerging assets such as IoT and sensors makes legacy security signatures less effective in

flagging malicious activities, as they exhibit different traffic patterns compared with typical IT networked hosts and assets.

Scalability: Software-based IDSes incur high computational costs and often do not scale well (unlike specialized hardware appliances) to handle high traffic rates cost-effectively [165]. Ineffective design of software components can make this problem even worse [88]. Therefore, signature-based IDSes running software platforms are mostly used by relatively smaller enterprises with low traffic rates. Researchers incorporated various techniques to improve the scalability of software IDSes. First, many prior works have exploited the concept of distributed computing. *The NIDS cluster* described in [145] used distributed computational nodes with optimized coordination approaches to achieve decent performance with software-based stateful intrusion detection. The authors of [29] proposed a domain-specific model that distributes traffic analysis across different processing units with specific functions to achieve scalability and efficient detection on multi-core hardware. Also, there exist works that developed methods to reduce the overheads by signature-matching. For example, work in [76] developed an alphabet compression table that combines distinct input signature symbols with identical behavior into one symbol, thereby reducing memory usage. *O3FA* [163] was proposed to achieve packet ordering and flow reassembly during pattern-matching phases with low buffer consumption, which is particularly useful in reducing computational overheads when handling attack traffic with long sequences of out-of-order packets. Moreover, with the increasing popularity of virtualization technologies, network intrusion detection on virtualized platforms is proven to be useful in reducing overheads, as it supports dynamic scaling of computational resources and flexible deployment of detection functions. For example, in [30], *J. Deng et al.* built a virtualized IDS regulated by a virtualized controller for semantic consistency, correct flow update, buffer overflow avoidance, and optimal scaling in real time. *vNIDS* [85] employed a detection state-sharing mechanism to reduce the virtualization overhead of IDS. Therefore, it achieves elasticity in detecting attacks of various profiles and also guarantees acceptable scalability.

C. Fine-Grained Detection using Flow Statistics

Distributed attacks sourced from external botnets are often mixed with benign traffic flows from legitimate sources to the victim enterprise server [93]. However, both proprietary rule-based and community signature-based detection systems (discussed in §IV-A and §IV-B) barely maintain fine-grained traffic statistics for individual flows. Instead, they focus more on aggregate statistics (destination IP/subnet-level). Therefore, they face challenges in providing the necessary visibility for precisely differentiating malicious flows from benign ones destined for the victim, particularly when attack sources are distributed. Many researchers have proposed methods for anomaly detection in network traffic using flow-level statistics, which enables them to achieve precise attack detection/mitigation without causing collateral damage [31], i.e., dropping only packets in malicious flows without affecting packets in benign flows.

1) *Scalability Issues*: It is important to note that collecting and analysing fine-grained flow statistics across a large and fairly active network may not always be practical. Therefore, many research efforts have been made to develop lightweight data structures to maintain flow statistics. *Kronecker graph* [84] was designed to model network flows using graphs generated by a non-standard matrix operation called Kronecker product, which is both descriptive and practical. The authors of [147] leveraged distributed computing nodes that collectively maintain in-memory graphs containing flow statistics to detect DDoS attacks cost-effectively. Many prior works employed streaming (online) algorithms to realize attack detection with relatively lower computational costs. *STONE* [19] maintains traffic attributes pertaining to the volume of activities (e.g., TCP SYN counts) for target asset groups. The authors employed streaming techniques that can scale and are more conducive to real-time monitoring. Work in [98] systematically reviewed the processing methods, such as “insert-only graphs”, “graph sketches”, and “sliding window”, for streaming graphs that help to reduce the computational costs when processing flow statistics. Work in [61] developed an anomaly detection scheme, looking for malicious flows such as DDoS and reconnaissance attacks. The proposed scheme aggregates flow alerts based on their similarities/correlations in five-tuple metadata to address the scalability.

2) *Identifying Important Features*: Identifying key predictive features from flow statistics for attack detection is another popular direction. Principal Component Analysis (PCA) is a method, widely used, to determine which features are more influential in classification tasks. To exclude (or reduce the impact of) redundant and less relevant attributes, the authors of [60] proposed a multi-stage feature selection method. They utilized lightweight filters and heavy regression models to extensively examine the importance of features in a progressive manner. Commonly used features for network anomaly detection were examined, and less than 40% of them were found to be effective in attack detection. The work in [96] introduced five groups of descriptive features (e.g., flow metadata features, sequence packet features, and general statistical features) of network flows. The authors demonstrated the efficacy of those attributes in detecting seven types of network attacks, including SSH patator, DDoS, and port scan.

3) *Statistical Learning Methods*: Developing statistical learning methods using flow characteristics for better attack detection has been explored by researchers. For example, S. Jin *et al.* discussed their work in detecting SYN flooding attacks using a covariance analysis model in [68]. They showed that the model could effectively distinguish benign flows and malicious flows by profiling their TCP headers. K. Lee *et al.* [81] applied clustering algorithms to a set of traffic features (e.g., randomness of source and destination IP addresses) to differentiate DDoS traffic from normal communications. The authors of [120] employ a statistical metric called “total variance distance” that quantifies the similarity between flows, achieving better performance in detecting attack traffic than legacy methods.

D. Highlights

In this section, we categorized attack detection methods into three types including proprietary rule-based, community signature-based, and fine-grained flow statistic-based detection.

Currently, the industry (at least large enterprises) widely adopts proprietary rule-based detection appliances for their ease of deployment and scalable operation. However, such an approach becomes less effective in combating dynamic attack vectors applied to expanded attack surfaces. It falls short of expectation, particularly at scale, when the enterprise network serves diverse asset classes and functionalities.

Signature-based detection is often realized as software products, relying on knowledge (i.e., signatures) supplied by open-source communities. Optimally selecting appropriate signatures, developed by security experts, could help (to a great extent) network operators (of medium/smaller enterprises) to quickly respond to emerging cyber threats, (e.g., “Log4Shell exploit” [40], [131], [80]). However, appropriately setting up the software environments, routinely updating signatures, and, importantly, trusting the open-source community may not always be feasible for administrators of large organizations. Besides, they are often packaged as software tools on commodity servers that make them expensive to scale for a large network.

Network attack detection methods, leveraging fine-grained flow statistics, have proven their superiority in precisely identifying victims, attackers, and malicious flows of a distributed attack. However, real-time maintenance and processing of fine-grained traffic statistics for many concurrent flows traversing an enterprise network can hardly scale. Therefore, achieving scalability while not compromising the quality of visibility for flow statistic-based methods is a crucial challenge to address before they can be widely adopted.

V. OPPORTUNITIES OF EMERGING PARADIGMS FOR NETWORK SECURITY

The advancements in programmable networking and machine learning (ML) techniques have opened up new possibilities for addressing current challenges in enterprise network security. Researchers have recently leveraged these two specific technologies in various network security domains. For instance, they have developed orchestration systems that offer flexible attack detection capabilities in ISP networks (§V-A) and proposed accurate algorithms specifically designed to detect certain types of attacks (§V-B).

These seminal prior works serve the research community with foundational lessons in developing practical and effective security systems for large enterprise networks. In the subsequent sections, we will delve into the existing research in the areas of network security that utilize programmable networks and ML techniques, respectively.

A. Programmable Networking for Network Security

The concept of programmable networking, broadly speaking, stems from technologies like Network Function Virtualization (NFV) [110] and Software-Defined Networking (SDN)

[43] and enables flexible network monitoring and controls. These technologies empower IT and cyber teams to dynamically configure and update flow rules and/or network functions, allowing for custom security measures and defense utilities in response to their ever-changing attack surfaces.

1) *Practical Challenges*: While programmable networking sounds promising in enhancing defense capabilities, its adoption faces several practical challenges [126], including the performance bottleneck of software controllers, specific vulnerabilities associated with controllers and switches, scalability limits of software-based network functions, and concerns about compatibility with existing systems and middleboxes.

To address these challenges, researchers have made efforts to develop practical solutions. For example, R. Sommer *et al.* [135] proposed a specialized NFV architecture that effectively utilizes multi-core processors to achieve scalable network intrusion detection. *O3FA* [163] is developed as a lightweight packet inspection engine using deterministic finite automaton (*i.e.*, a finite-state machine) that processes out-of-order packet streams without reassembling flows. Thus, the system requires less memory than other packet inspection engines. *StateAlyzr* [74] identifies and reduces the unnecessary operational processes for state clones in security middleboxes to achieve low computational overheads. *NetBricks* [115] employs a zero-copy software isolation mechanism that significantly reduces the computational overheads in CPU and RAM usage on typical NFV platforms. The NFV framework *OpenNetVM* [166] is designed with high-level abstractions, allowing users to quickly build and deploy customized network functions without the need to handle complex optimization of computing resource allocations. The hybrid packet processing pipeline *ParaBox* [167] is specifically designed to incorporate parallel network functions, resulting in superior performance compared to traditional serial function chaining mechanisms. *StatelessNF* [70] breaks down virtual network functions into two components: a state management component to store stateful traffic information and a stateless packet processing component to extract packet information at high speeds. They are well separated and orchestrated by SDN utilities so that the traffic is processed in a more scalable manner. *vNIDS* [85] tackles the challenges of inefficient (and costly) detection of SDN/VNF-based systems by developing techniques such as state sharing among detection modules and dynamic slicing of detection logic programs.

2) *Prototypes for Attack Detection*: In addition to the research efforts to develop practical methods, prototypes have been built for certain attack detection problems that utilize programmable networks. These prototypes showcase the potential of programmable networking in enhancing security measures.

For example, R. Braga *et al.* [17] developed a system that utilizes programmable switches to extract flow statistics for detecting flooding attacks. S. Lim *et al.* [86] utilized OpenFlow-based switches to achieve flexible isolation of bots in DDoS attacks. In a study by K. Giotis *et al.* [51], the authors developed a system that combines OpenFlow and sFlow utilities to collect and process network statistics for scalable anomaly detection. The *FlowTags* system [42] employs an SDN architecture to achieve flexible security enforcement through middleboxes at

a network level with relatively low computational overheads. The *Bohatei* system [41] utilizes SDN proactive and reactive flow rules to dynamically orchestrate network traffic forwarding through backbone networks, diverting attack traffic to be handled by specialized security middleboxes with appropriate computational resources. C. Yoon *et al.* [162] demonstrated the feasibility of utilizing programmable networks for cybersecurity by developing representative security functions of in-line firewalls, passive IDS, and network anomaly detectors with SDN technology. The *Atlantic* system [27] leverages the flexibility of SDN to detect, classify, and mitigate malicious flows in relatively small networks (*e.g.*, consisting of 100 hosts and two switches). In [87], the authors utilize SDN reactive routing to selectively forward only the initial packets of each network flow for deep packet inspection. J. Deng *et al.* [30] constructed a virtual firewall architecture using SDN and NFV, enabling elastic rule placement and flexible detection functionalities. *Sonata* [55] achieves scalable traffic processing by offloading resource-intensive and repetitive network functions from software processors to hardware programmable switches. *ACC-Turbo* [9] implements an in-network mechanism on a programmable P4 switch to detect DDoS attacks with short and high-rate pulse patterns. Lastly, *PEDDA* [93] utilizes a programmable control-plane switch (*i.e.*, OpenFlow) and virtual network functions to dynamically apply DDoS detection modules, each with specific capabilities and costs, enabling fine-grained detection and scalable real-time operation.

B. Machine Learning for Network Security

Machine learning techniques have proven their efficacy in accurate inference (classification and/or anomaly detection) in domains like computer vision and speech recognition. Although the application of machine learning in cybersecurity faces practical challenges [49], [133], researchers have made significant progress in developing machine learning-based methods and systems to enhance the security of various networks [18]. These relevant prior works are wide in scope and objectives, providing valuable insights to the research community.

For example, the *MADAM ID* framework, proposed in [83], utilizes machine learning-based data mining techniques to process network telemetry data (*e.g.*, packet and flow events and connection status) for intrusion detection. In [128], J. Shum *et al.* employed simple neural networks trained with back propagation algorithms for detecting network attacks such as DDoS, spam, and exfiltration. The *BotMiner* system [53] applied unsupervised clustering algorithms to characterize the behavior of botnet groups that exhibit similar patterns in their command-and-control activities. Such similarity can be determined in traffic attributes like the number of flows generated per hour. M. Lyu *et al.* utilized clustering algorithms in [91] to classify enterprise DNS assets and health metrics based on their DNS traffic profiles for anomaly detection. In [75], L. Koc *et al.* introduced the Hidden Naive Bayes (HNB) method for network intrusion detection, outperforming other machine learning models in handling high-dimensional data, identifying dependent features, and reducing computational overheads.

Table II: Prior relevant surveys with different objectives and focuses.

Survey	Key objectives	Focused aspects	The latest year of article reviewed
[102]	DDoS	Attack and defense mechanism	2003
[49]	Anomaly-based Network Intrusion Detection	Attack and defense mechanism	2005
[164]	DDoS	Attack and defense mechanism	2012
[66]	Typical Protection Methods for IT Infrastructure	Attack and defense mechanism	2013
[18]	Intrusion Detection via Data Mining and Machine Learning	Dataset and defense mechanism	2015
[144]	Flow-based Intrusion Detection	Dataset and defense mechanism	2016
[148]	Network Firewall	Configuration method	2016
[143]	Network Vulnerability Scanning	Attack and defense mechanism	2017
[107]	Next-Generation-Firewall	Defense mechanism	2015
[23]	Network Situational Awareness	Defense mechanism	2018
[125]	Moving Target Defense	Defense mechanism	2019
Our survey	Asset Monitoring and Distributed Attack Detection	Attack and defense mechanism	2023

Table III: Relevant surveys on different network types.

Survey	Network type	The latest year of article
[24]	Wireless sensor network	2007
[124]	Software-defined network	2014
[4]	Software-defined network	2015
[21]	IoT network	2018
[13]	IoT network	2017
[3]	Cloud networks	2018
[20]	3GPP 5G network	2019
[14]	Cloud networks	2021
Our survey	Enterprise network	2023

A scheme designed by M. Javed *et al.* in [67] specifically focuses on detecting SSH brute-forcing attacks using a beta-binomial distribution model. The authors in [127] developed an ensemble model that combines Bayesian Network with Gain Ratio for feature selection and Artificial Neural Network for attack detection. C. Hsieh *et al.* proposed a DDoS detection system in [58] that employs neural networks on Apache Spark big data computing clusters to handle high data rates traffic. *DeepLog* [32] employs deep learning algorithms for detecting anomalies in system logs collected from enterprise hosts. H. Siadati *et al.* used machine learning-based algorithms to identify anomalous logins within an enterprise network [129]. In [139], D. Tang *et al.* developed data-driven models to detect relatively low-rate DoS attacks that exhibit abnormal patterns in the frequency, variation, and distribution of TCP flows.

It is worth noting that most existing works focus on developing high-accuracy models and algorithms to detect specific types of attacks. These efforts demonstrate the effectiveness of machine learning-based methods in addressing network security challenges. Furthermore, these advancements lay the foundation for developing data-driven solutions in asset management and distributed attack detection, offering tailored approaches to large enterprises.

C. Highlights

In this section, we have summarized the latest advancements in, as well as the adoption of programmable networking and machine learning techniques in the field of network security.

The programmable networking paradigm offers flexible and dynamic traffic forwarding and measurement capabilities,

outperforming traditional systems. That said, there are still practical challenges to be addressed before a wide adoption is realized. Key challenges include scalability issues arising from limited resources in both control and data planes (*e.g.*, switch memory size and flow entries), the need for skilled administrators comfortable with coding, and compatibility with existing network infrastructures. Existing methods empower network and cyber teams to adjust the visibility levels and the granularity of network telemetry for real-time security inferences. Known techniques allow for collecting precise, fine-grained statistics based on the network size, composition of connected assets, and the evolution of attacks.

Machine learning techniques (data-driven models) promise to automatically classify hosts' traffic or detect attacks by trained network data models instead of relying solely on manually defined thresholds and/or signatures. However, applying machine learning algorithms to network security requires the community to overcome certain challenges. These barriers include striking a balance between descriptive attributes and scalability, handling false positives that lead to operational implications, and ensuring the explainability of inferences. Careful consideration of these challenges is essential for successfully adopting machine learning-based solutions in network security.

VI. DISCUSSION ON RESEARCH GAPS

After extensively reviewing current techniques of asset behavior monitoring and distributed attack detection for enterprise networks, we found several open issues that require further investigation in future work.

A. Dynamic and Scalable Host Monitoring

The complexity of networked assets and the dynamic nature of their communication patterns pose challenges for legacy methods that rely on static configurations and inferences. While dynamic networked graphs can effectively capture host behavioral profiles, maintaining such graphs for a large and high data- rates network becomes infeasible. Therefore, there is a need to develop methods that can achieve scalable monitoring of assets while ensuring dynamic and fine-grained visibility into necessary traffic portions. This represents a valuable future direction in network security research.

B. Role-Aware Network Attack Detection

Every networked asset can be a victim of distributed attacks. We note that different assets come with relatively distinct communication patterns and vulnerabilities. Therefore, an effective defense system demands some form of customization in the monitoring techniques based on types of assets contiguity to the network. Presently, attack detection appliances often apply generic detection mechanisms to the entire network or, at best, rely on some manual configurations by the network administrators, who manage policies for specific hosts or network segments. However, this approach may not fully incorporate the distinct characteristics and vulnerabilities of individual enterprise hosts. Hence, a valuable contribution would involve the development of automatic configurations for attack detection mechanisms utilizing the traffic profiles of enterprise hosts. By analyzing these profiles, automated configuration methods can dynamically tailor detection mechanisms to align with each host's unique requirements and attributes.

C. Explainable ML-Based Attack Detection

Despite the promising prediction quality offered by machine learning (ML) methods for tasks like host classification and attack detection in controlled environments, their performance in operational networks can be unknowingly impacted by various factors, such as limited training data, imperfect statistical features, and algorithmic biases, or concept drifts. In order to use the predictions from (black-box) ML models for high-stakes decision-making, network operators may require some assurance, assistance, or at least an explanation that helps them interpret and analyze specific inferences made by trained models. This is particularly important to avoid mishandling false-positives, leading to unnecessary disruptions and resource wastage.

D. Self-Driving Enterprise Security Systems

The complexity of configuring current network security and management systems poses challenges to the IT and cyber departments of large enterprises. These systems often rely on manual configurations and tweaks to specify intents like which assets/segments receive priority for protection, protecting against which types of threats (*e.g.*, scans or DDoS), setting detection thresholds, and choosing appropriate mitigation actions. Managing numerous and complex policies can be cumbersome and prone to human errors. Additionally, sub-optimal configurations (*e.g.*, inconsistent and conflicting rules) can compromise the security and stability of networks. To prevent or at least manage these risks and improve operational efficiency, there is a need to explore developing "self-driving" security systems. These systems are expected to operate (to a great extent) automatically, gradually becoming autonomous and independent of manual configurations.

VII. RELATED SURVEYS ON NETWORK SECURITY

We now discuss some of the existing survey papers that focused on different aspects of network security.

A. Attack Detection Methods

A group of literature reviews focuses on categorizing methods for detecting network attacks. Their key objectives, focused aspects, and the latest year of articles reviewed by those surveys are summarized in Table II. The features of our study are captured in the last row of this table.

J. Mirkovic *et al.* [102] provided a taxonomy of DDoS attacks and corresponding defense mechanisms. The authors of [49] categorized the system architecture of underlying modules inside network intrusion detection systems (NIDS). S. T. Zargar *et al.* [164] highlighted DDoS defense mechanisms with a focus on where on the network they are applied and when defense actions take place. The authors in [66] comprehensively discussed vulnerabilities in the networking ecosystem targeted by emerging cyber-attacks and their countermeasures. The types and mechanisms of data mining and machine learning methods and their applications in cyber-security research have been discussed in [18]. Work in [144] summarizes flow-based intrusion detection techniques, datasets, and prototypes. A. Voronkov *et al.* [148] thoroughly reviewed the usability aspect of firewall configurations. A. Tundis *et al.* [143] reviewed existing vulnerability scanner tools applied for benign or malicious purposes. The authors in [107] discussed the functionalities of popular next-generation firewalls (NGFW) and their efficacy in coping with emerging network threats. C. Chen *et al.* [23] focused on the architecture of situational awareness systems for network security, which include data collection, situational understanding, prediction, and visualization. S. Sengupta *et al.* [125] comprehensively discussed the effective methods to defend against attacks originating from moving targets.

B. Specific Attacks on Certain Network Types

A cluster of survey papers studied attacks specific to certain network types, which are more vulnerable given their distinct characteristics. Table III summarizes these survey papers, showing their focus and the latest year of articles reviewed. The features of our survey are shown in the last row of this table.

X. Chen *et al.* [24] summarized security problems in wireless sensor networks and discussed the efficacy of existing defense techniques. The authors in [124] and [4] highlighted security issues of software-defined networks and provided a list of key requirements for an effective defense architecture. Works in [21] and [13] focused on network intrusion detection for IoT networks. N. Agrawal *et al.* [3] particularly focused on defense mechanisms against DDoS attacks for cloud computing networks. J. Cao *et al.* [20] summarized the security challenges, requirements, and gaps in 3GPP 5G networks. A. Bhardwaj *et al.* [14] surveyed solutions developed by academia and industry to combat DDoS attacks on cloud networks.

C. The Focus of our Survey

Prior surveys primarily categorized certain attack types (*e.g.*, DDoS) and corresponding defense methods depending on aspects such as target locations, attacking techniques, and

the exploited network vulnerabilities. In contrast, our survey focuses on studying a broad range of distributed network attacks (not limited to DDoS), countermeasure techniques, and opportunities promised by emerging paradigms, specifically for enterprise networks. Our survey reviews relevant research papers published until 2023 and provides valuable insights into unique challenges and opportunities for enterprise network security. This survey fills a gap in the existing literature by providing a comprehensive reference specifically tailored to the needs of researchers and practitioners working in enterprise network security.

VIII. CONCLUSION

This survey focused on distributed network attacks on enterprise-connected assets and highlighted various countermeasures, including asset monitoring and attack detection systems. We discussed two types of distributed attacks (reconnaissance and DDoS) on enterprise assets. We reviewed existing methods (developed by academia and industry) for monitoring the behaviors of enterprise hosts and detecting distributed attacks. We highlighted the capabilities of two emerging/rising technologies (*i.e.*, programmable networks and ML) that bring new opportunities in addressing enterprise network security concerns. Lastly, we highlight several open issues as valuable future directions that are worthwhile to be explored. This paper provides a solid reference and inspires future research addressing enterprise network security issues.

REFERENCES

- [1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A Multifaceted Approach to Understanding the Botnet Phenomenon," in *Proc. ACM IMC*, Jul 2006, pp. 41–52.
- [2] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zipper ZMap: Internet-Wide Scanning at 10 Gbps," in *Proc. USENIX WOOT*, San Diego, CA, Dec 2014.
- [3] N. Agrawal and S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.
- [4] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.
- [5] M. Aimon *et al.*, "Performance Evaluations of IPTables Firewall Solutions under DDoS attacks," vol. 11, 12 2015.
- [6] Akamai Technologies, "How Securing Recursive DNS Proactively Protects Your Network," <https://blogs.akamai.com/2017/10/how-securing-recursive-dns-proactively-protects-your-network.html>, 2017, accessed: 2021-05-14.
- [7] M. Albanese, S. Jajodia, and S. Noel, "Time-Efficient and Cost-Effective Network Hardening using Attack Graphs," in *Proc. IEEE/IFIP DSN*, Jun 2012, pp. 1–12.
- [8] E. Albin and N. C. Rowe, "A Realistic Experimental Comparison of the Suricata and Snort intrusion-detection systems," in *Proc. WAINA*, Mar 2012, pp. 122–127.
- [9] A. G. Alcoz, M. Strohmeier, V. Lenders, and L. Vanbever, "Aggregate-Based Congestion Control for Pulse-Wave DDoS Defense," in *Proc. ACM SIGCOMM*, Amsterdam, Netherlands, Aug 2022.
- [10] H. M. J. Almohri, L. T. Watson, D. Yao, and X. Ou, "Security Optimization of Dynamic Networks with Probabilistic Graph Modeling and Linear Programming," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 4, pp. 474–487, 2016.
- [11] T. Bakhshi and B. Ghita, "User Traffic Profiling," in *Proc. IEEE ITA*, Sep 2015, pp. 91–97.
- [12] H. Ballani and P. Francis, "CONMan: A Step towards Network Manageability," in *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug 2007, p. 205–216.
- [13] E. Benkhelifa, T. Welsh, and W. Hamouda, "A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.
- [14] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed Denial of Service Attacks in Cloud: State-of-the-Art of Scientific and Commercial Solutions," *Computer Science Review*, Feb 2021.
- [15] P. Bhattacharya and S. K. Ghosh, "Analytical Framework for Measuring Network Security using Exploit Dependency Graph," *IET Information Security*, vol. 6, no. 4, pp. 264–270, 2012.
- [16] K. Borders, J. Springer, and M. Burnside, "Chimera: A Declarative Language for Streaming Network Traffic Analysis," in *Proc. USENIX Security*, Bellevue, WA, Feb 2012.
- [17] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow," in *Proc. IEEE LCN*, Oct 2010, pp. 408–415.
- [18] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [19] M. Callau-Zori, R. Jiménez-Peris, V. Gulisano, M. Papatrantaflou, Z. Fu, and M. Patiño Martínez, "STONE: A Stream-Based DDoS Defense Framework," in *Proc. ACM SAC*, Coimbra, Portugal, Mar 2013.
- [20] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A Survey on Security Aspects for 3GPP 5G Networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 170–195, 2020.
- [21] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.
- [22] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Measuring Botnets in the Wild: Some New Trends," in *Proc. ASIACCS*, Singapore, Republic of Singapore, Apr 2015.
- [23] C. Chen *et al.*, "A Survey of Network Security Situational Awareness Technology," in *Proc. ICAIS*, New York, NY, USA, Jul 2019.
- [24] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor Network Security: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.
- [25] Cisco Security, "Cisco Firewall Best Practices," <https://bit.ly/3jQEWtj>, accessed: 2020-08-31.
- [26] Cloudflare, "Famous DDoS attacks — The largest DDoS attacks of all time," <https://www.cloudflare.com/en-gb/learning/ddos/famous-ddos-attacks/>, 2022, accessed: 2022-09-14.
- [27] A. S. da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "Atlantic: A framework for anomaly traffic detection, classification, and mitigation in SDN," in *Proc. IEEE/IFIP NOMS*, Apr 2016, pp. 27–35.
- [28] D. Day and B. Burns, "A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines," in *Proc. ICDS*, Feb 2011, pp. 187–192.
- [29] L. De Carli, R. Sommer, and S. Jha, "Beyond Pattern Matching: A Concurrency Model for Stateful Deep Packet Inspection," in *Proc. ACM CCS*, Scottsdale, Arizona, USA, Nov 2014, p. 1378–1390.
- [30] J. Deng, H. Li, H. Hu, K.-C. Wang, G.-J. Ahn, Z. Zhao, and W. Han, "On the Safety and Efficiency of Virtual Firewall Elasticity Control," in *Proc. NDSS*, Jan 2017.
- [31] C. Dietzel, M. Wichtlhuber, G. Smaragdakis, and A. Feldmann, "Stellar: Network Attack Mitigation Using Advanced Blackholing," in *Proc. ACM CoNEXT*, Dec 2018.
- [32] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs Through Deep Learning," in *Proc. ACM CCS*, Dallas, Texas, USA, Nov 2017.
- [33] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A Search Engine Backed by Internet-Wide Scanning," in *Proc. ACM CCS*, Denver, Colorado, USA, Oct 2015.
- [34] Z. Durumeric, M. Bailey, and J. A. Halderman, "An Internet-Wide View of Internet-Wide Scanning," in *Proc. USENIX Security*, San Diego, CA, USA, Aug 2014.
- [35] Z. Durumeric *et al.*, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *Proc. USENIX Security*, Washington, D.C., USA, Aug 2013.
- [36] Dynatrace, "Network monitoring," <https://bit.ly/3RleesZ>, 2022, accessed: 2022-9-28.
- [37] M. Eslahi, M. V. Naseri, H. Hashim, N. M. Tahir, and E. H. M. Saad, "BYOD: Current State and Security Challenges," in *2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, 2014, pp. 189–192.

- [38] ESNet, "Firewall Performance Issues," <https://fasterdata.es.net/network-tuning/firewall-performance-issues/>, 2018, accessed: 2017-09-01.
- [39] D. Eswaran, C. Faloutsos, S. Guha, and N. Mishra, "SpotLight: Detecting Anomalies in Streaming Graphs," in *Proc. ACM KDD*, London, United Kingdom, Aug 2018.
- [40] D. Everson, A. Bastola, R. Mittal, S. Munde, and L. Cheng, "A Comparative Study of Log4Shell Test Tools," in *IEEE Secure Development Conference*, Oct 2022.
- [41] S. K. Fayaz *et al.*, "Bohatei: Flexible and Elastic DDoS Defense," in *Proc. USENIX Security*, Washington, D.C., USA, Aug 2015.
- [42] S. K. Fayazbakhsh, L. Chiang, V. Sekar, M. Yu, and J. C. Mogul, "Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions Using Flowtags," in *Proc. USENIX NSDI*, Seattle, WA, Apr 2014, p. 533–546.
- [43] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, p. 87–98, Apr 2014.
- [44] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," in *Proc. ESORICS*, Sep 2005, pp. 319–335.
- [45] K. Fukuda, J. Heidemann, and A. Qadeer, "Detecting Malicious Activity With DNS Backscatter Over Time," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3203–3218, Oct 2017.
- [46] K. Fukuda and J. Heidemann, "Who Knocks at the IPv6 Door? Detecting IPv6 Scanning," in *Proc. ACM IMC*, Boston, MA, USA, Oct 2018.
- [47] P. Gao, X. Xiao, D. Li, Z. Li, K. Jee, Z. Wu, C. H. Kim, S. R. Kulkarni, and P. Mittal, "SAQL: A Stream-based Query System for Real-Time Abnormal System Behavior Detection," in *Proc. USENIX Security*, Baltimore, MD, USA, Jul 2018.
- [48] J. Garcia-Alfaro, F. Cuppens, N. Cuppens-Boulahia, S. Martinez, and J. Cabot, "Management of Stateful Firewall Misconfiguration," *Comput. Secur.*, vol. 39, p. 64–85, Nov 2013.
- [49] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.
- [50] Gartner Peer Insights, "IT Infrastructure Monitoring Tools Reviews and Ratings," <https://www.gartner.com/reviews/market/it-infrastructure-monitoring-tools>, 2022, accessed: 2022-9-28.
- [51] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.
- [52] E. Glatz, "Visualizing Host Traffic through Graphs," in *Proc. International Symposium on Visualization for Cyber Security*, Ottawa, Ontario, Canada, Sep 2010, p. 58–63.
- [53] G. Gu, R. Perdisci, J. Zhang, W. Lee *et al.*, "BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection," in *Proc. USENIX Security*, vol. 5, no. 2, Jul 2008, pp. 139–154.
- [54] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," in *Proc. NDSS*.
- [55] A. Gupta, R. Harrison, M. Canini, N. Feamster, J. Rexford, and W. Willinger, "Sonata: Query-driven Streaming Network Telemetry," in *Proc. ACM SIGCOMM*, Budapest, Hungary, Aug 2018.
- [56] T. Heinrich, R. R. Obelheiro, and C. A. Maziero, "New kids on the drdos block: Characterizing multiprotocol and carpet bombing attacks," in *Proc. PAM*, Virtual Event, Mar 2021.
- [57] H. Heo and S. Shin, "Who is Knocking on the Telnet Port: A Large-Scale Empirical Study of Network Scanning," in *Proc. ASIACCS*, Incheon, Republic of Korea, Jun 2018.
- [58] C. Hsieh and T. Chan, "Detection DDoS Attacks Based on Neural-Network using Apache Spark," in *Proc. ICASI*, May 2016.
- [59] Q. Hu, M. R. Asghar, and N. Brownlee, "Measuring IPv6 DNS Reconnaissance Attacks and Preventing Them Using DNS Guard," in *Proc. IEEE/FIP DSN*, Luxembourg City, Luxembourg, Jun 2018.
- [60] F. Iglesias and T. Zseby, "Analysis of Network Traffic Features for Anomaly Detection," *Machine Learning*, vol. 101, no. 1, pp. 59–84, 2015.
- [61] D. Ippoliti, C. Jiang, Z. Ding, and X. Zhou, "Online Adaptive Anomaly Detection for Augmented Network Flows," *ACM Trans. Auton. Adapt. Syst.*, vol. 11, no. 3, Sep. 2016.
- [62] H. Irshad, G. Ciocarlie, A. Gehani, V. Yegneswaran, K. H. Lee, J. Patel, S. Jha, Y. Kwon, D. Xu, and X. Zhang, "TRACE: Enterprise-Wide Provenance Tracking for Real-Time APT Detection," *IEEE Transactions on Information Forensics and Security*, Sep 2021.
- [63] H. Ismail, H. S. Hamza, and S. M. Mohamed, "Semantic Enhancement for Network Configuration Management," in *Proc. IEEE GCIoT*, Alexandria, Egypt, Dec 2018, pp. 1–5.
- [64] J. H. Jafarian, E. Al-Shaar, and Q. Duan, "An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2562–2577, 2015.
- [65] A. Jakalan *et al.*, "Social Relationship Discovery of IP Addresses in the Managed IP Networks by Observing Traffic at Network Boundary," *Comput. Netw.*, vol. 100, no. C, p. 12–27, May 2016.
- [66] J. Jang-Jaccard and S. Nepal, "A Survey of Emerging Threats in Cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973 – 993, 2014.
- [67] M. Javed and V. Paxson, "Detecting Stealthy, Distributed SSH Brute-forcing," in *Proc. ACM CCS*, Berlin, Germany, Nov 2013.
- [68] S. Jin *et al.*, "A Covariance Analysis Model for DDoS Attack Detection," in *Proc. IEEE ICC*, Paris, France, Jun 2004.
- [69] J. Jusko and M. Rehak, "Revealing Cooperating Hosts by Connection Graph Analysis," in *Proc. SecureComm*. Berlin, Heidelberg: Springer Berlin Heidelberg, Mar 2013, pp. 241–255.
- [70] M. Kablan, A. Alsudais, E. Keller, and F. Le, "Stateless Network Functions: Breaking the Tight Coupling of State and Processing," in *Proc. USENIX NSDI*, Boston, MA, USA, Mar 2017, p. 97–112.
- [71] P. Kalmbach, D. Hock, F. Lipp, W. Kellerer, and A. Blenk, "NOracle: Who is Communicating with Whom in My Network?" in *Proc. ACM SIGCOMM Posters and Demos*, Beijing, China, Aug 2019.
- [72] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multi-level Traffic Classification in the Dark," in *Proc. ACM SIGCOMM*, Oct 2005.
- [73] M. Karami and D. McCoy, "Understanding the Emerging Threat of DDoS-As-a-Service," in *Proc. USENIX LEET*. Washington, D.C.: USENIX Association, Aug 2013.
- [74] J. Khalid, A. Gember-Jacobson, R. Michael, A. Abhashkumar, and A. Akella, "Paving the Way for NFV: Simplifying Middlebox Modifications Using StateAlyzt," in *Proc. USENIX NSDI*, Santa Clara, CA, Mar 2016, p. 239–253.
- [75] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A Network Intrusion Detection System Based on A Hidden Na'ive Bayes Multiclass Classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 492–13 500, 2012.
- [76] S. Kong, R. Smith, and C. Estan, "Efficient Signature Matching with Multiple Alphabet Compression Tables," in *Proc. SecureComm*, Sep 2008.
- [77] C. Kreibich and J. Crowcroft, "Honeycomb: Creating Intrusion Detection Signatures Using Honey pots," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, p. 51–56, Jan. 2004.
- [78] M. Kühner, T. Hüpperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *Proc. USENIX Security*, San Diego, CA, USA, Aug 2014.
- [79] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-wide Traffic Anomalies," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 219–230.
- [80] E. Leblond, "Suricata to the Log4j Rescue," <https://www.stamus-networks.com/blog/suricata-to-the-log4j-rescue>, 2021, accessed: 2023-06-16.
- [81] K. Lee, J. Kim, K. Kwon, Y. Han, and S. Kim, "DDoS Attack Detection Method Using Cluster Analysis," *Expert Systems with Applications*, vol. 34, pp. 1659–1665, 04 2008.
- [82] P. J. Lee, H. Guo, and B. Veeravalli, "Enhancing CII firewall performance through hash based rule lookup," in *Proc. IEEE TENCON*, Nov 2017, pp. 2285–2290.
- [83] W. Lee and S. J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 227–261, Nov. 2000.
- [84] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker Graphs: An Approach to Modeling Networks," *J. Mach. Learn. Res.*, vol. 11, p. 985–1042, Mar 2010.
- [85] H. Li, H. Hu, G. Gu, G.-J. Ahn, and F. Zhang, "vNIDS: Towards Elastic Security with Safe and Efficient Virtualization of Network Intrusion Detection Systems," in *Proc. ACM CCS*, Toronto, Canada, Oct 2018.
- [86] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-Oriented DDoS Blocking Scheme for Botnet-Based Attacks," in *Proc. IEEE ICUFN*, Jul 2014, pp. 63–68.

- [87] C. Liu, A. Raghuramu, C.-N. Chuah, and B. Krishnamurthy, “Piggy-backing Network Functions on SDN Reactive Routing: A Feasibility Study,” in *Proc. ACM SOSR*, Santa Clara, CA, USA, Apr 2017.
- [88] G. Liu, K. K. Ramakrishnan, M. Schlansker, J. Tourrilhes, and T. Wood, “Design Challenges for High Performance, Scalable NFV Interconnects,” in *Proc. the Workshop on Kernel-Bypass Networks*, Los Angeles, CA, USA, Aug 2017.
- [89] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, “Systematically Evaluating Security and Privacy for Consumer IoT Devices,” in *Proc. IoT S&P*, Dallas, Texas, USA, Nov 2017.
- [90] M. R. Lyu and L. K. Y. Lau, “Firewall Security: Policies, Testing and Performance Evaluation,” in *Proc. COMPSAC*, Oct 2000, pp. 116–121.
- [91] M. Lyu, H. H. Gharakheili, C. Russell, and V. Sivaraman, “Enterprise dns asset mapping and cyber-health tracking via passive traffic analysis,” *IEEE Transactions on Network and Service Management*, 2023.
- [92] M. Lyu, H. Habibi Gharakheili, and V. Sivaraman, “Classifying and Tracking Enterprise Assets via Dual-Grained Network Behavioral Analysis,” *Computer networks*, Dec 2022.
- [93] —, “PEDDA: Practical and Effective Detection of Distributed Attacks on Enterprise Networks via Progressive Multi-Stage Inference,” *Computer networks*, 2023.
- [94] M. Lyu, D. Sherratt, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, “Quantifying the Reflective DDoS Attack Capability of Household IoT Devices,” in *Proc. ACM WiSec*, Boston, MA, USA, Jul 2017.
- [95] M. Antonakakis *et al.*, “Understanding the Mirai Botnet,” in *Proc. USENIX Security*, Vancouver, BC, USA, Aug 2017.
- [96] C. Ma, X. Du, and L. Cao, “Analysis of Multi-Types of Flow Features Based on Hybrid Neural Network for Improving Network Anomaly Detection,” *IEEE Access*, vol. 7, pp. 148 363–148 380, 2019.
- [97] ManageEngine OpManager, “ManageEngine OpManager, the trusted network monitoring software.” <https://www.manageengine.com/network-monitoring/>, 2022, accessed: 2022-09-28.
- [98] A. McGregor, “Graph Stream Algorithms: A Survey,” *SIGMOD Rec.*, vol. 43, no. 1, p. 9–20, May 2014.
- [99] Microsoft 365, “Create an Inbound Port Rule,” <https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/create-an-inbound-port-rule>, 2022, accessed: 2022-09-28.
- [100] J. Mirkovic, G. Prier, and P. Reiher, “Source-End DDoS Defense,” in *Proc. IEEE NCA*, Apr 2003, pp. 171–178.
- [101] J. Mirkovic *et al.*, “D-WARD: A Source-End Defense Against Flooding Denial-of-Service Attacks,” *IEEE Trans. Dependable Secur. Comput.*, vol. 2, no. 3, pp. 216–232, Jul. 2005.
- [102] J. Mirkovic and P. Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.
- [103] L. Muñoz González, D. Sgandorra, A. Paudice, and E. C. Lupu, “Efficient Attack Graph Analysis through Approximate Inference,” *ACM Trans. Priv. Secur.*, vol. 20, no. 3, Jul 2017.
- [104] U. Mustafa, M. M. Masud, Z. Trabelsi, T. Wood, and Z. A. Harthi, “Firewall Performance Optimization using Data Mining Techniques,” in *Proc. IWCNC*, Jul 2013.
- [105] Y. Nadji, M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee, “Beheading Hydras: Performing Effective Botnet Takedowns,” in *Proc. ACM CCS*, Berlin, Germany, Nov 2013.
- [106] S. Narayana *et al.*, “Language-Directed Hardware Design for Network Performance Monitoring,” in *Proc. ACM SIGCOMM*, Los Angeles, CA, USA, Aug 2017.
- [107] K. Neupane, R. Haddad, and L. Chen, “Next Generation Firewall for Network Security: A Survey,” in *Proc. IEEE SoutheastCon*, Tampa Bay Area, FL, USA, Apr 2018, pp. 1–6.
- [108] J. Newsome, B. Karp, and D. Song, “Polygraph: Automatically Generating Signatures for Polymorphic Worms,” in *Proc. IEEE S&P*, May 2005, pp. 226–241.
- [109] E. L. Ngoupé, S. Stoesel, C. Parisot, S. Hallé, P. Valtchev, O. Cherkaoui, and P. Boucher, “A Data Model for Management of Network Device Configuration Heterogeneity,” in *Proc. IEEE/IFIP IM*, Ottawa, Canada, May 2015, pp. 1230–1233.
- [110] L. Nobach, O. Hohlfeld, and D. Hausheer, “New Kid on the Block: Network Functions Visualization: From Big Boxes to Carrier Clouds,” *SIGCOMM Comput. Commun. Rev.*, vol. 46, no. 3, pp. 7:1–7:8, Jul. 2018.
- [111] R. Ogie, “Bring Your Own Device: An Overview of Risk Assessment,” *IEEE Consumer Electronics Magazine*, vol. 5, no. 1, pp. 114–119, 2016.
- [112] G. Oikonomou, J. Mirkovic, P. Reiher, and M. Robinson, “A Framework for a Collaborative DDoS Defense,” in *Proc. ACSAC*, Dec 2006, pp. 33–42.
- [113] A. Orebaugh and B. Pinkard, *Nmap in the Enterprise: Your Guide to Network Scanning*. Elsevier, 2011.
- [114] Palo Alto Networks, “PA-3000 Series Datasheet,” <https://bit.ly/2MPcSk2>, 2018, accessed: 2018-28-1.
- [115] A. Panda, S. Han, K. Jang, M. Walls, S. Ratnasamy, and S. Shenker, “NetBricks: Taking the V out of NFV,” in *Proc. USENIX OSDI*, Savannah, GA, USA, Nov 2016.
- [116] S. Patton, W. Yurcik, and D. Doss, “An Achilles? Heel in Signature-based IDS: Squealing False Positives in SNORT,” in *Proc. RAID*, Oct 2001.
- [117] V. Paxson, “Bro: A System for Detecting Network Intruders in Real-Time,” *Computer networks*, vol. 31, no. 23–24, pp. 2435–2463, 1999.
- [118] J. J. Pfeiffer III *et al.*, “Attributed Graph Models: Modeling Network Structure with Correlated Attributes,” in *Proc. ACM WWW*, Seoul, Korea, Apr 2014.
- [119] N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic Security Risk Management Using Bayesian Attack Graphs,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.
- [120] H. Rahmani, N. Sahli, and F. Kamoun, “DDoS Flooding Attack Detection Scheme Based on F-Divergence,” *Computer Communications*, vol. 35, no. 11, pp. 1380–1391, 2012.
- [121] P. Richter and A. Berger, “Scanning the Scanners: Sensing the Internet from a Massively Distributed Network Telescope,” in *Proc. ACM IMC*, Amsterdam, Netherlands, Oct 2019.
- [122] M. Roesch *et al.*, “Snort: Lightweight intrusion detection for networks.” in *Lisa*, vol. 99, no. 1, 1999, pp. 229–238.
- [123] K. Salah, K. Elbadawi, and R. Boutaba, “Performance Modeling and Analysis of Network Firewalls,” *IEEE Transactions on Network and Service Management*, vol. 9, no. 1, pp. 12–21, Mar 2012.
- [124] S. Scott-Hayward, S. Natarajan, and S. Sezer, “A Survey of Security in Software Defined Networks,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623–654, 2016.
- [125] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, “A Survey of Moving Target Defenses for Network Security,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1909–1941, 2020.
- [126] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, “Are We Ready for SDN? Implementation Challenges for Software-Defined Networks,” *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.
- [127] A. K. Shrivastava and A. K. Dewangan, “An Ensemble Model for Classification of Attacks with Feature Selection Based on KDD99 and NSL-KDD Data Set,” *International Journal of Computer Applications*, vol. 99, no. 15, 2014.
- [128] J. Shun and H. A. Malki, “Network intrusion detection system using neural networks,” in *Proc. ICNC*, vol. 5, Oct 2008, pp. 242–246.
- [129] H. Siadati and N. Memon, “Detecting Structurally Anomalous Logins Within Enterprise Networks,” in *Proc. ACM CCS*, Dallas, Texas, USA, Nov 2017.
- [130] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, “Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, Aug 2019.
- [131] Snort, “SID 1-58813,” https://www.snort.org/rule_docs/1-58813, 2021, accessed: 2023-06-16.
- [132] H. Solomon, “Rio Games Faced Olympic-Sized DDoS Attacks,” <https://www.itworldcanada.com/article/rio-olympics-faced-olympic-sized-ddos-attacks/386207>, 2020, accessed: 2020-08-18.
- [133] R. Sommer *et al.*, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” in *Proc. IEEE S&P*, Oakland, California, USA, May 2010.
- [134] R. Sommer and V. Paxson, “Enhancing Byte-Level Network Intrusion Detection Signatures with Context,” in *Proc. ACM CCS*, 2003, p. 262–271.
- [135] R. Sommer, V. Paxson, and N. Weaver, “An Architecture for Exploiting Multi-Core Processors to Parallelize Network Intrusion Prevention,” *Concurr. Comput. : Pract. Exper.*, vol. 21, no. 10, p. 1255–1279, Jul 2009.
- [136] A. Sridharan and T. Ye, “Tracking Port Scanners on the IP Backbone,” in *Proc. ACM LSAD*, Kyoto, Japan, Aug 2007.
- [137] A. Starschenko, N. Tcholtchev, A. Prakash, I. Schieferdecker, and R. Chaparadza, “Auto-Configuration of OSPFv3 Routing in Fixed IPv6 Networks,” in *Proc. ICUMT*, Oct 2015, pp. 196–205.

- [138] B. Sullivan, "Rio 2016 Olympics Suffered Sustained 540Gbps DDoS Attacks," <https://bit.ly/38kN0Pb>, 2018, accessed: 2018-05-01.
- [139] D. Tang, J. Chen, X. Wang, S. Zhang, and Y. Yan, "A New Detection Method for LDoS Attacks based on Data Mining," *Future Generation Computer Systems*, vol. 128, pp. 73–87, Mar 2022.
- [140] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: Finding Bots in Network Traffic Without Deep Packet Inspection," in *Proc. ACM CoNEXT*, Dec 2012.
- [141] H. Tegenaw and M. Kifle, "Application Aware Firewall Architecture to Enhance Performance of Enterprise Network," in *Proc. IEEE AFRICON*, Sep 2015, pp. 1–10.
- [142] M. Thomas and A. Mohaisen, "Kindred Domains: Detecting and Clustering Botnet Domains Using DNS Traffic," in *Proc. IW3C2*, Seoul, Korea, Apr 2014.
- [143] A. Tundis, W. Mazurczyk, and M. Mühlhäuser, "A Review of Network Vulnerabilities Scanning Tools: Types, Capabilities and Functioning," in *Proc. ACM ARES*, Hamburg, Germany, Aug 2018.
- [144] M. Umer, M. S. Ramzan, and Y. Bi, "Flow-Based Intrusion Detection: Techniques and Challenges," *Computers & Security*, vol. 70, 06 2017.
- [145] M. Vallentin, R. Sommer, J. Lee, C. Leres, V. Paxson, and B. Tierney, "The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware," in *Proc. RAID*, Gold Coast, Australia, Sep 2007, p. 107–126.
- [146] A. K. Vasu *et al.*, "Improving Firewall Performance by Eliminating Redundancies In Access Control Lists," *International Journal of Computer Networks*, vol. 6, pp. 92–107, Sep 2014.
- [147] J. J. Villalobos, I. Rodero, and M. Parashar, "An Unsupervised Approach for Online Detection and Mitigation of High-Rate DDoS Attacks Based on an In-Memory Distributed Graph Using Streaming Data and Analytics," in *Proc. IEEE/ACM BDCAT*, Austin, Texas, USA, Dec 2017.
- [148] A. Voronkov, L. H. Iwaya, L. A. Martucci, and S. Lindskog, "Systematic Literature Review on Usability of Firewall Configuration," *ACM Comput. Surv.*, vol. 50, no. 6, Dec 2017.
- [149] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, "DDoS Defense by Offense," *ACM Trans. Comput. Syst.*, vol. 28, no. 1, Aug 2010.
- [150] A. Wang, W. Chang, S. Chen, and A. Mohaisen, "A Data-Driven Study of DDoS Attacks and Their Dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 03, pp. 648–661, May 2020.
- [151] C. Wang, D. Zhang, H. Lu, J. Zhao, Z. Zhang, and Z. Zheng, "An Experimental Study on Firewall Performance: Dive into the Bottleneck for Firewall Effectiveness," in *Proc. International Conference on Information Assurance and Security*, Nov 2014, pp. 71–76.
- [152] L. Wang *et al.*, "An Attack Graph-Based Probabilistic Security Metric," in *Data and Applications Security XXII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [153] P. Wang, J. Luo, W. Li, and Y. Qu, "NCP: A Network Control Programmable Platform of Trustworthy Controllable Network," in *Proc. IEEE DCSW*, Jun 2009, pp. 221–226.
- [154] W. Wang, B. Yang, and Y. V. Chen, "Detecting Subtle Port Scans through Characteristics Based on Interactive Visualization," in *Proc. RIIT*, Atlanta, Georgia, USA, Oct 2014.
- [155] A. Wedgbury and K. Jones, "Automated Asset Discovery in Industrial Control Systems: Exploring the Problem," in *Proc. ICS-CSR*, Ingolstadt, Germany, Sep 2015, p. 73–83.
- [156] A. Welzel, C. Rossow, and H. Bos, "On Measuring the Impact of DDoS Botnets," in *Proc. European Workshop on System Security*, Amsterdam, The Netherlands, Apr 2014.
- [157] A. K. Y. Wong, P. Ray, N. Parameswaran, and J. Strassner, "Ontology mapping for the interoperability problem in network management," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 10, pp. 2058–2068, 2005.
- [158] A. Wool, "A Quantitative Study of Firewall Configuration Errors," *Computer*, vol. 37, no. 6, pp. 62–67, Jun 2004.
- [159] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming Botnets: Signatures and Characteristics," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, p. 171–182, Aug. 2008.
- [160] K. Xu, F. Wang, and L. Gu, "Behavior Analysis of Internet Traffic via Bipartite Graphs and One-Mode Projections," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, p. 931–942, Jun 2014.
- [161] F. Yarochkin, Y. Huang, Y. Hu, and S. Kuo, "Mining Large Network Reconnaissance Data," in *Proc. IEEE PRDC*, Vancouver, BC, Canada, Dec 2013.
- [162] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, "Enabling Security Functions with SDN," *Comput. Netw.*, vol. 85, no. C, pp. 19–35, Jul 2015.
- [163] X. Yu, W. Feng, D. Yao, and M. Becchi, "O3FA: A Scalable fFinite Automata-Based Pattern-Matching Engine for Out-of-Order Deep Packet Inspection," in *Proc. ACM/IEEE ANCS*, Mar 2016, pp. 1–11.
- [164] S. T. Zargar *et al.*, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, Mar 2013.
- [165] M. Zhang *et al.*, "Poseidon: Mitigating Volumetric DDoS Attacks with Programmable Switches," in *Proc. NDSS*, San Diego, CA, USA, Feb 2020.
- [166] W. Zhang, G. Liu, W. Zhang, N. Shah, P. Lopreiato, G. Todeschi, K. Ramakrishnan, and T. Wood, "OpenNetVM: A Platform for High Performance Network Service Chains," in *Proc. ACM HotMiddlebox*, Florianopolis, Brazil, Aug 2016, p. 26–31.
- [167] Y. Zhang, B. Anwer, V. Gopalakrishnan, B. Han, J. Reich, A. Shaikh, and Z.-L. Zhang, "ParaBox: Exploiting Parallelism for Virtual Network Functions in Service Chaining," in *Proc. ACM SOSR*, Santa Clara, CA, USA, Apr 2017, p. 143–149.
- [168] F. Zhu, M. W. Mutka, and L. M. Ni, "Service Discovery in Pervasive Computing Environments," *IEEE Pervasive Computing*, vol. 4, no. 4, p. 81–90, Oct 2005.