Bot ahead! - Proactive Botnet Defense in ISP Networks using Digital Twins

Christian Dietz, Marcel Antzek, Gabi Dreo Rodosek
Bundeswehr University Munich, Neubiberg, Germany
Email: {Christian.Dietz, Marcel.Antzek, Gabi.Dreo}@unibw.de

***Abstract*—The increasing reliance on online services across critical sectors (e.g., banking, healthcare, and government) makes robust network infrastructures paramount. However, these services are frequently targeted by malicious actors leveraging distributed overlay networks (botnets) to execute attacks (e.g., data theft, service disruption, and ransomware campaigns). The impact of such attacks correlates with the scale and capabilities of the compromised infrastructure. Therefore, early and proactive detection of botnet infections is crucial for limiting their growth and mitigating potential damage. This paper presents and evaluates a Digital Twin-based approach for proactive botnet detection and multi-level mitigation within large-scale networks, particularly ISP infrastructures. Through simulations using a dedicated network topology framework, we demonstrate how these strategies can effectively limit botnet growth and significantly reduce their impact on critical network operations.**

I. Introduction

The Internet infrastructure has become critical in recent years, with a growing number of services (e.g., banking, healthcare, government, and commerce services) being provisioned online [1]. However, this pervasive reliance has also rendered the Internet and its services targets for criminal actors [2]. These criminal actors aim to compromise data integrity, disrupt service availability, extort a ransom, or execute other malicious activities. A common method for executing such attacks involves hiring botnets [3]. These distributed overlay networks, composed of compromised machines, are frequently leveraged for a wide array of malicious activities. Consequently, proactive defense against botnets is of high importance for network operators and Internet Service Providers (ISPs) to mitigate the impact of attacks and to safeguard their infrastructure [4]. Thus, preventing the growth and further compromise of machines by botnets is crucial to limit their overall attack power [5].

Existing network simulation approaches often face limitations when applied to modeling contemporary network complexity at scale, making their use challenging [6]. Furthermore, they are often only capable of simulating known behavior, which means the botnet must already be spread, detected, and analyzed to simulate its possible next actions, making such approaches reactive by design [7], [8].

To overcome the limitations of existing approaches, our approach enables simulation at scale in a secure, sandboxed digital twin (DT) environment, allowing the execution of real botnet samples and studying their behavior without prior analysis [9]. In the context of botnet campaigns, this enables the automatic testing of multiple mitigation strategies in parallel to identify an optimized mitigation that reduces the impact on productive environments [10]. The main goal of this work is to demonstrate the application of DTs in detecting and mitigating malware attacks that spread across large-scale ISP networks.

Therefore, we propose extending KubeNSF, an existing large-scale network simulation framework and the successor of DMEF [9], into a DT environment. This DT enables the secure execution of self-spreading malware, allowing for real-time analysis of their propagation behavior, as well as the simulation of diverse mitigation strategies. Ultimately, this DT environment provides network operators with a powerful tool to optimize their mitigation strategies, allowing them to fine-tune responses based on various critical constraints, including anticipated downtime costs or the desired effectiveness of mitigation actions.

Our approach evaluates proactive and multi-layered defenses that combine network telemetry (e.g., Netflow) with DNS-based early detection [11], as well as collaboration scenarios [12]. By embracing the principles of defense-in-depth, our work emphasizes the importance of integrating multiple, complementary defense mechanisms across the network stack to improve overall resilience against botnet-driven threats.

Our main contributions are: i) A scalable DT environment to analyze large-scale malware campaigns. ii) An evaluation of a hybrid defense approach addressing multiple phases of the botnet life-cycle. iii) An environment for automated and reproducible generation of labeled malware traffic.

The remainder of this paper is organized as follows: Section II introduces terminology used throughout the paper. Section III presents the scenarios, requirements, and assumptions. Section IV provides related work. Section V presents the experimental setup and methodology used for functional validation. Section VI presents the results and evaluation. Finally, Section VII closes the paper and presents future work.

II. Terminology

This Section defines the terminology used in this paper to ensure understandability. We describe the malware addressed by our approach and explain the hypervisor technologies used to create the digital twin.

a) Malware and Botnets: Malware software is designed to disrupt, damage, or gain unauthorized access to a computer or networks. Malware detection refers to the process of detecting the presence of malware on a host system or of identifying

978-3-903176-75-1

malicious executables [13]. One type of malware that features a complex infrastructure is a botnet. Botnets are networks of devices infected with malware, enabling a malicious actor to remotely control them. A bot is a compromised device remotely controlled by a botmaster. Botnets act according to a botnet life cycle [14] when compromising new devices.

b) Proactive vs Reactive: Proactive refers to any action taken prior to an attack or compromise, without any external trigger event. Reactive describes any action taken after a compromise is detected, and their approaches are often triggered by an external event. Due to the spreading characteristics of botnets and their varying behavior across the phases of the botnet life cycle [11], [15] some reactive actions on one bot are proactive measure on yet uncompromised systems.

c) Detection and Mitigation: Detection identifies the botnet behavior (e.g., scanning for open ports and credentials) caused by compromised systems. We adhere to detection as proactively performed without externally triggered events, and mitigation as a reactive process triggered by the detection output. Mitigation refers to a set of actions that prevent a system from being compromised and joining a botnet, and isolating infected systems. Mitigation can be performed on different levels [16] and range from creating firewall rules that isolate vulnerable or already infected systems at the edge level, to DNS and BGP-based quarantine and sinkholing [14], [17].

d) Digital Twin: A DT in ISP networks is a formally designed, continuously synchronized digital replica of the network or its components. It integrates real-time telemetry, configuration data, topological information, and behavioral models to provide a dynamic, holistic view of the network's current and projected state, enabling advanced use cases such as capacity planning and failure simulation.

e) Cloud-native and Containerization: In this paper, we use the following terms: Cloud-native, Containerization, Containers, Scaling, and Self-Healing [18]. We adhere to the definition presented in The Cloud Native Computing Foundation (CNCF) Cloud Native Glossary [19].

III. Scenario, Requirements and Assumptions

In this Section, we describe the network environment in which we evaluate our defense approaches. Next, we define requirements that the DT-based defense approaches should fulfill. Finally, we describe our assumptions about the scenario.

A. Scenarios

This work focuses on corporate and ISP networks threatened by botnets. As botnets are usually characterized by i) a life-cycle of multiple stages and ii) the use of different Internet protocols and infrastructures, as well as iii) spread globally across different Autonomous Systems (ASs), detection and mitigation vary depending on the life-cycle phase addressed or the general perspective on the threat.

B. Requirements

To address the distributed nature of the botnet threat in an ISP network environment, our work focuses on multiple parts of the Internet infrastructure. Thus following 9 requirements should be fullfilled: a) Scalability, b) Resource efficiency, c) Multi-level capability, d) Real-time adaptability, e) Interoperability, f) Data security and isolation., g) Ease of use, h) Automation and i) Self-healing capabilities.

a) Scalability: Given that typical corporate and ISP networks can vary in size, ranging from small networks with few routers to networks with hundreds of devices, a Network Digital Twin (NDT) approach must be scalable by design. The system must efficiently handle hundreds of network nodes and the associated network traffic, data processing, and state management, without compromising performance or stability. Scalability is crucial to ensure that the DT can accurately mirror real-world environments, from smaller enterprise networks to large-scale ISP backbones, providing a versatile platform for comprehensive botnet threat analysis and mitigation.

b) Resource Efficiency: To achieve the required scalability and ensure economic viability, a large-scale DT environment must be effective and resource-efficient. The network environment is created and operated multiple times, which can increase operational costs and human resources for infrastructure maintenance.

c) Multi-level Capability: A NDT for botnet detection and mitigation must support the simulation of multiple levels of the Internet infrastructure (e.g., DNS resolution, inter- and intra-AS routing, edge-level features (e.g., NAT, firewalls)).

d) Real-time Adaptability: Real-time adaptability must incorporate changes from the reference network's state into its digital replicas to ensure i) realistic simulations for taking precise mitigation actions, and ii) avoiding damage or service degradation to production systems where the simulated state is out of sync with the network situation.

e) Interoperability: A network DT system must support various network protocols and multiple virtualization technologies. Some real-world deployments might incorporate legacy devices that are not compatible with containerization, requiring full Virtual Machine (VM) support.

f) Data Security and Isolation: As network DTs typically duplicate large portions of real operational data and configurations, they are a valuable source of information or a potential attack vector for malicious actors. As DTs are used to analyze real malware, the DT must support data security and isolation.

g) Ease of Use: Given the complexity of large corporate and ISP network environments, and to minimize the likelihood of human errors, the design, deployment, and operation of the resulting network DTs must be easy to manage.

h) Automation: Automation enables DT systems to operate at scale, respond in real-time, and continuously evolve with minimal human intervention. In security-critical DT environments, it enables the automated enforcement of security guidelines, thereby minimizing the chance of human errors and misconfigurations. Automation not only allows DTs to mirror an existing system but also enables automated actions based on simulations in the mirrored environments.

i) Self-healing: Due to the propagation of malware within NDTs, self-healing is a critical operational requirement.

The system must automatically revert to a clean, known-good state any time after an experiment has been completed. This includes resetting compromised nodes, restoring configurations, and purging introduced malware or artifacts. This process can also be triggered manually by a network operator.

C. Assumptions

We assume that the corporate network operators and ISPs have the resources and telemetry to operate one or more DT infrastructures within their network topology. Further, we assume that operators of large-scale networks usually use Intrusion Detection Systems, IP/DNS blacklisting, and netflow-based monitoring (e.g., ntop) as well as DNS resolvers. We assume that our approach can also be applied to smaller corporate networks, as it follows cloud-native principles and can be deployed on rented cloud infrastructure. We assume that botnets are still evolving and maintaining operations in the networks against which we evaluate our mitigation approaches. In some cases, botnets may operate in other domains (e.g., different ASs) than those they initially target. Thus, we have taken collaborative mitigation approaches into account.

IV. Related Work

The detection and mitigation of botnets in ISP networks has received significant attention over the past decades, with research spanning from static rule-based detection systems [13] to adaptive anomaly detection using machine learning [20], [21]. However, most traditional methods fail to address sophisticated and distributed modern botnets operating across different layers of the Internet hierarchy [11].

DT technologies have emerged as promising tools, and several studies have explored their use in creating virtual representations of networks for real-time monitoring and proactive defense. Wang et al. [22] proposed a blockchain-assisted DT framework for early botnet detection in industrial IoT networks, combining packet header inspection with deep learning to detect abnormal behavior. Alam et al. [23] designed a DT-assisted DDoS detection mechanism for autonomous core networks, leveraging online learning to adapt detection models in real time. While these studies demonstrate the potential of DTs for cyber defense, their scope is often limited to enterprise or IoT contexts, with simplified network models and a lack of full Internet protocol stack emulation. They rarely consider real-world ISP-level dynamics (e.g., BGP-based inter-domain routing, DNS infrastructure, or realistic NAT traversal).

Discrete Event Simulations (DES) or network simulators (e.g., NS-3 [24], OMNeT++ [25]) offer scalability but abstract away critical behaviors (e.g., dynamic routing updates, firewalls, and DNS resolution), making them unsuitable for modeling real-world botnet behavior. These simulations rarely incorporate actual malware binaries due to safety concerns, which limits the realism of threat modeling [26].

Dietz et al. [14] and Kolias et al. [27] utilize real malware for behavioral analysis, but are limited to host-level analysis or confined to isolated network testbeds, lacking the routing complexity and scale typically seen in ISP environments. Dietz et al. [9] and Adalsteinsson et al. [28] executed real malware in emulated networks to overcome the inherent limitation of DES, but are still limited in scalability, as they rely on full VM-based simulation, which creates significant overhead when used to emulate thousands of systems [29].

As containers offer advantages in resource utilization and operational flexibility compared to VMs, containerization has seen increased application in the field of large-scale network analysis. Utilizing the container runtime Docker and SDN, DockSDN provides a network simulation framework with considerable flexibility and scalability [30].

The SEED emulator [31] represents an emulation of large-scale Internet environments, including ASs and their associated routing protocols. It offers a programmable interface that utilizes customizable Python classes for topology declaration, which are subsequently translated into a set of Docker containers. Further, SEED incorporates graphical user interfaces that provide real-time visualization of network dynamics.

Holterbach et al. [32] developed the mini-Internet project to simulate network topologies comprised of multiple ASs. The project utilizes Docker containers and supports overlay networks via SDN. However, complex topologies are defined by a collection of custom configuration files, making deployments and modifications challenging. The containerlab project (https://containerlab.dev/) leverages abstract configuration files. These files simplify the topology declaration, with the resulting architecture being deployed using a combination of Docker containers, VMs, and virtual Ethernet pairs.

Cloud-based methodologies address the limitations of local deployments by enabling the automated provisioning of multi-node simulation environments, thus enhancing reproducibility and fidelity. Niehaus et al. [33] demonstrate a Kubernetes-based platform for building and operating scalable, distributed, and repeatable simulation scenarios, with a specific application to power grid analysis. Borsatti et al. [34] propose a method that optimizes operational Kubernetes deployments using a dedicated DT deployment.

However, realistic ISP topologies with edge-level NAT and firewalling, intra-AS routing via OSPF, inter-AS routing via BGP, and a custom DNS infrastructure to deploy realistic malware samples within a controlled environment to observe propagation and C2 behavior are still missing.

V. Experimental Setup

This Section presents a technical description of the three case studies conducted within our DT environment to evaluate various botnet detection and mitigation strategies.

A. Underlying Infrastructure and Experimental Setup

The experiments were conducted on consumer-grade hardware using an Intel(R) Core(TM) i7-8750H CPU (6 Cores, 12 Threads) and 16GB DDR4 RAM. The underlying KubeNSF framework internally utilizes a Kubernetes cluster as a simulation environment, running Kubernetes in Docker (kind (https://kind.sigs.k8s.io/) with one control plane node and two worker nodes. The router containers are emulated using

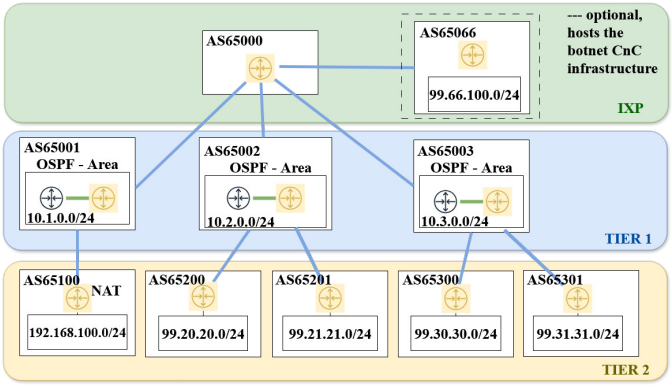

Fig. 1. Multi-level Infrastructure

FRRouting images, an open-source routing protocol suite. Additionally, all router components implemented traffic recording functionality. This enables the capability for selective network traffic capturing, offering the generation of labeled datasets on all protocol levels within the proposed simulation environment. As simulated network components, a lightweight nginx container image was chosen as the base image to minimize the resource footprint while offering basic connectivity.

We conduct 3 case studies to evaluate different detection and defense approaches: i) DNS-based early detection and blackholing of Botnet DGA domains, ii) collaborative detection and IP/Port Filtering based on IP blacklisting and remotely triggered blackholing (RTBH) and iii) hybrid multi-level defense strategies using OSPF and BGP-based routing capabilities.

Each case study addresses a specific segment of the Internet hierarchy, encompassing the edge, inter-AS, and intra-AS levels, thereby providing a comprehensive assessment of defense mechanisms across diverse network scales. In our malware analysis use case, all hosts were configured with an OpenSSH server instance with weak credentials. All case studies utilized an example network topology, as depicted in Figure 1. eBGP routes are represented by blue connections, OSPF routes by green, and DNS resolvers by yellow systems. Unless otherwise specified, OSPF routes are redistributed into BGP. All routers run iptables and dynamic packet filtering across all layers of the network. We empirically determined a global route redistribution convergence time of approx. 1 minute. The topology's design ensures horizontal and vertical scalability (e.g., additional end-users per subnet and AS or more ASes) that can be deployed through manifest file modifications.

For each case study, we employed the same bot, characterized by a centralized infrastructure and self-spreading capabilities via scanning and password brute-forcing [14], [27]. During our case studies, each bot automatically scanned random IPs within four distinct subnets. Each subnet contains 50 intentionally vulnerable, containerized hosts with weak SSH credentials. After a successful infection, newly compromised bots attempted to connect to the C2 server to join the botnet. The C2 server logged the initial compromise of each infected host. Our case studies simulate an ISP offering a protection service. We vary the proportion of customers using this protection service to observe its impact on botnet propagation and reflect scenarios where not all customers subscribe to the service plan or use the provider's infrastructure (e.g., DNS).

B. Case Study 1: DNS-based early detection and blackholing of Botnet DGA domains

In this case study, we compared the mitigation of DNS-based Domain Generation Algorithms (DGAs) at the customer level with provider-level approaches at the ISP's DNS resolvers. Botnets commonly use DGAs to generate large volumes of domain names to establish C2 communication, thereby evading static blacklists.

Our experiments focus on autonomous bot scanning and infection attempts. We simulated varying adoption rates of DNS-based detection among customers (50%) and different proportions of customers utilizing the ISP's default DNS resolver (25%, 50%, 75%). The simulation also incorporated bots scanning both within and across AS boundaries. The current detection methodology relies on blacklisting.

This case study is based on the experimental setup and topology, with modifications to DNS as a detection and mitigation vector. We enhanced the bots to employ DGA-based anti-blacklisting techniques. Upon successful infection of a new device, the bot randomly attempted to connect to a C2-URL generated by a DGA algorithm to join the botnet. In the event of a connection failure, the bot retried after a 5-second delay using a different DGA-generated URL.

C. Case Study 2: Collaborative Detection and IP/Port Filtering based on C2 Infrastructure Information.

This case study examines the effectiveness of collaborative detection and filtering mechanisms within the simulated DT environment for mitigating early botnet activity. We focus on IP-level filtering strategies, which are often easier to deploy at scale and allow inter-organizational collaboration. Our approach covers:

1) Static blacklisting, where known malicious IP addresses are blocked on a firewall level.
2) Remotely triggered blackholing, which is used to divert unwanted traffic away from the target network.
3) Fine-grained IP and port blocking, enabling selective filtering of traffic based on both IP and service port.
4) Protocol-level blocking, where entire protocols (e.g., IRC, Telnet) known to be abused by botnets were denied.
5) IP Blacklists exchange-based collaboration, where local blacklist updates are shared among trusted partners.

The approaches were evaluated in a controlled virtual environment that emulated realistic traffic, using the proposed DT replica of an enterprise network depicted in Figure 1. Detection was collaboratively managed across multiple simulated domains with dynamic rule propagation.

D. Case Study 3: Hybrid multi-level defense strategies using OSPF and BGP-based Routing capabilities

This case study examines the effectiveness of integrated detection and mitigation strategies deployed across multiple

layers of the Internet infrastructure, leveraging routing mechanisms such as OSPF or BGP. By adopting a hybrid multi-level defense approach, the system can benefit from the strengths of both intra-AS (OSPF) and inter-AS (BGP) routing protocols to detect and mitigate botnet activities more efficiently. OSPF enables fine-grained, rapid detection and localized response within ASs, while BGP facilitates broader, coordinated mitigation efforts across multiple network domains. This combined strategy enhances the overall resilience and scalability of defense mechanisms in large-scale network environments, allowing for the timely identification and containment of threats before they spread widely. This case study is based on the reference topology shown in Figure 1.

VI. Evaluation

In this Section, we describe the qualitative and quantitative evaluation of the DT approach for multi-level botnet defense to limit the spread and attack power of botnets.

A. Qualitative Evaluation

In this Section, we perform a qualitative evaluation of our network DT approach. We describe the evaluation criteria and present the evaluation results for our approach.

1) Evaluation criteria: The evaluation criteria are derived from the requirements presented in Section III-B.

a) Scalability describes the DT's ability to be applied to various topologies and is evaluated with a varying number of ASs and Hosts per AS.
b) Resource efficiency ensures low operational overhead using our DT and reduces the cost of redundant hardware.
c) Multi-level capability is the ability to address network-based attacks at different Internet hierarchy levels (end-users, ISPs, Internet Backbone).
d) Real-time adaptability describes a DT's ability to change during runtime, mirroring the dynamic network state as closely as possible at any time.
e) Interoperability ensures that a DT's network can consist of a variety of different systems and services, into which data is incorporated.
f) Data security/Isolation describes the ability to prevent accidental spread of malware during its execution.
g) Ease of Use avoids distraction and supports network operators during dynamic and stressful conditions (e.g., ongoing attacks in a productive environment).
h) Automation is the DT's ability to provide and deploy computational resources with minimal human effort, contributing to faster response times and easier use.
i) Self-Healing is the DT's capability to detect a broken state (e.g., from executed malware) and automatically recover or reset to a known-good state.

2) Qualitative Evaluation Results: The qualitative results of our DT approach are as follows.

a) Scalability: Our DT approach is scalable by design. Tests were run on consumer-grade hardware using 295 containerized workloads, with no degradation observed. Migrated to a cloud, our approach scaled a network topology to 3,200 container instances, operating effectively with 42 GB RAM.

b) Resource efficiency: Performance was measured under baseline and high-load conditions on consumer-grade hardware. Our multi-AS reference simulation scenario (Figure 1) demonstrated notable efficiency, with a mean CPU utilization of 6.84% and an average memory usage of 6.35 GB. During a computationally intensive botnet spread (250 infected hosts initiating scans), peak CPU utilization was below 65% and memory usage only 7.1 GB. This demonstrates the DT's capability for complex, large-scale simulations on commodity hardware without excessive resource consumption.

c) Multi-level capability: The framework demonstrated versatility by implementing defenses at distinct levels of the Internet hierarchy. This included DNS-based blackholing, IP/Port filtering (ISP level), and infrastructure manipulation via OSPF/BGP-based routing policies to isolate and redirect malicious traffic. This demonstrates the DT's ability to host and analyze the synergistic effects of diverse, hybrid, multidimensional defense architectures.

d) Real-time adaptability: Our DT approach exposes granular interfaces (interactive consoles, programmatic APIs) to all simulated components, enabling dynamic reconfiguration and interactive control during execution. Operators can respond to simulation events (e.g., botnet detection) by altering security policies (e.g., activating BGP-based blackholing). This supports Human-in-the-Loop Analysis ("what-if" scenarios) for manually adjusting parameters and observing the consequences on botnet propagation or mitigation efficacy, as well as automated synchronization with external systems, while ensuring high fidelity by allowing the DT's configuration to mirror changes in production environments automatically.

e) Interoperability: The DT approach supports heterogeneous network protocols and multiple virtualization technologies. Protocol interoperability is shown in multi-level defense case studies, integrating DNS, OSPF, and BGP. Recognizing that not all systems can be containerized, the architecture explicitly provides transparent integration of full VMs. This capability is crucial for integrating legacy systems, proprietary network appliances, or specific OSs. The hybrid virtualization used in our approach transparently integrates containerized and full VMs, enabling them to communicate as if they were on a unified network fabric. This balances performance and efficiency with modeling diverse IT environments.

f) Data Security/Isolation: To prevent malware from spreading, our DT approach was built upon the KubeNSFs "secure-by-default" posture. By default, the entire simulation environment operates within an isolated virtual network. All egress traffic originating from the simulated components is automatically null-routed through multiple cascade packet filters to minimize chances for accidental misconfiguration. While isolation is the default, controlled and auditable data flows are possible through programmatic interfaces, allowing legitimate interactions without compromising the host or external networks. This ensures realistic simulation scenarios and data-capturing, where potential botnets might receive code

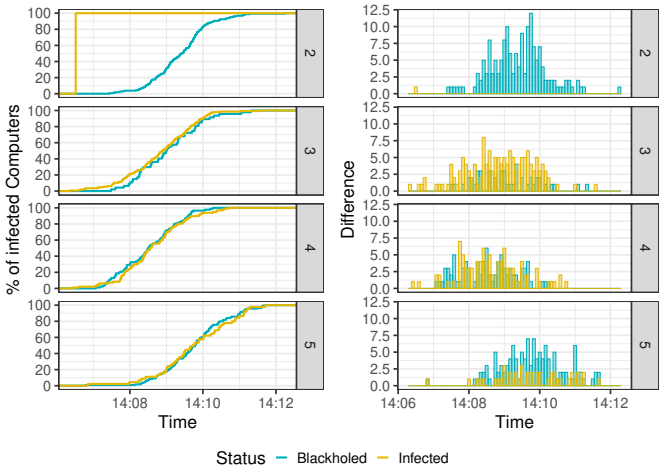

Fig. 2. Results of the DNS-based sinkholing

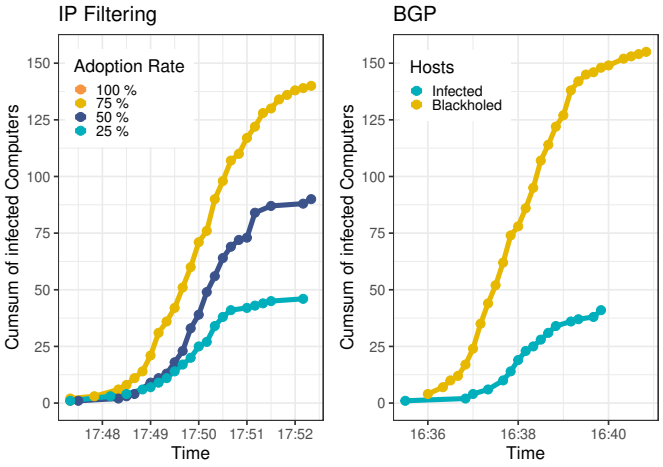

Fig. 3. Results of the IP Filtering and BGP Blackholing

updates from external sources, which would not be received in fully isolated environments.

g) Ease of Use: By adhering to cloud-native principles, the usability of our DT approach was significantly enhanced. It utilizes declarative Kubernetes specifications to define all simulation components. This approach enables the creation of version-controlled and reproducible simulations. Further, routine operational tasks (e.g., deploying or tearing down complex network topologies) are abstracted into single-command invocations. This operational simplicity enables operators to focus on analyzing experimental outcomes rather than addressing the challenges of infrastructure management.

h) Automation: The KubeNSF framework supports a holistic automation strategy that encompasses both the underlying infrastructure and experimental workloads. The base simulation environment can be programmatically provisioned using standard IaC tools (e.g., Terraform, Ansible). Specific network topologies are deployed automatically via declarative manifests and scripted workflows. This end-to-end automation facilitates the rapid and consistent instantiation of complex experimental setups, requiring minimal manual intervention.

i) Self-Healing: To support a high-throughput workflow, our DT approach provides highly efficient and automated reset capabilities. The ability to return the DT to a known-good state ensures experimental validity. A deployment of the simulation environment, with the internal components pre-cached, averages only 4.23 minutes. More granularly, deploying a complex topology as described in Figure 1 takes approximately 90 seconds, with teardown requiring a similar duration. Consequently, an operator can execute a full reset in roughly three minutes. This capability, combined with the real-time reconfiguration of individual components, allows researchers to efficiently conduct numerous ”what-if” scenarios, significantly accelerating the experimental lifecycle.

B. Quantitative Evaluation Results

In this Section, we perform a quantitative evaluation of our three case studies.

a) Results of Case Study 1: Figure 2 presents the different adoption rates among the users using the IPs protection service. Case-1 represents the scenario where no user is using the protection service, while Case-2 represents the scenario where all users use the protection service. In Case-2, the initial infection happens fast, but all bots can be effectively blackholed. Case 3 represents 25% adoption rate, Case 4 represents 50% adoption rate and Case 5 75% adoption rate of the customers. It can be seen that a higher adoption rates result in a decreased slope of the infection graphs (slower infection in the network) and reduces the overall number of active bots relative to the adoption rate. Figure 2 summarizes the results of our case study and shows that the spread of the botnet is effectively reduced, the more customers use the ISP's protection service.

b) Results of Case Study 2: Figure 3 shows that the spread of the botnet can be reduced by approximately the same proportion as the customer adoption rate of the ISP's IP- and port-based filtering techniques. With a 100% adoption rate, only initial infections occur, but all subsequent spreading is immediately prevented. The initial infection points are overpainted by the other cases in Figure 3.

c) Results of Case Study 3: The simulation is initiated with a single compromised host within the 99.66.100.0/24 subnet. This host propagates the infection throughout the topology and designates another host within the same subnet as the C2 server (utilizing a DNS-based rendezvous mechanism). Leveraging BGP, a more specific route for the C2 server's IP address was announced globally, redirecting all C2-bound traffic of other network areas to a designated analysis server. This BGP sinkholing technique effectively isolated all infected nodes from their C2 infrastructure, thereby neutralizing the botnet's operational capabilities and preventing its further expansion. The BGP sinkholing mechanism proved to be a precise and effective mitigation tool. Log data from the analysis server verified the interception of C2 traffic from all compromised hosts located in remote subnets. As anticipated by the design,

the initial infected host within the 99.66.100.0/24 subnet was not redirected, since its local traffic to the C2 server did not depend on inter-subnet BGP routing, thereby demonstrating the targeted application of the mitigation technique.

In summary, the results demonstrate that the initially defined requirements for the hybrid defense concept were comprehensively met. The proposed architecture and its underlying methodology provided the necessary framework to address the challenges of multi-level botnet containment successfully.

VII. Conclusion and Future Work

Botnets represent a significant and adaptive threat to Internet-connected systems and services. One approach to detecting and mitigating botnets focuses on DT infrastructures. In this paper, we introduced scalable DT infrastructures that were effectively leveraged for proactive detection and mitigation of spreading malware attacks in large-scale ISP networks.

Our results showed that DTs can be effectively used to simulate multiple defense-in-depth scenarios, with different combinations of detection and mitigation actions, to conduct proactive what-if analysis. Our results highlight the role of ISPs in effective botnet defense and deterrence.

Building upon these results, we plan to extend our simulations to include a more diverse set of network topologies to investigate deep learning approaches (e.g., Graph Neural Networks (GNNs)). GNNs can leverage both the underlying network topology and the C2 overlay structures of botnets, potentially leading to more sophisticated and effective defense mechanisms. We will implement a more diverse set of attack actions within the simulated botnets. This will enable us to simulate multiple independent attack campaigns in parallel, which can then serve as reference data to study ML-based differentiation and attribution of different botnet operators.

Acknowledgment

This work was partially funded by the Federal Ministry of Research, Technology and Space of Germany (#16KISK002).

References

[1] S. Bellamkonda, "Cybersecurity in Critical Infrastructure: Protecting the Foundations of Modern Society," *International Journal of Communication Networks and Information Security*, 2020.

[2] N. Provos *et al.*, "Cybercrime 2.0: when the cloud turns dark," *Commun. ACM*, Apr. 2009.

[3] M. F. Safitra *et al.*, "Cyber Resilience: Research Opportunities," in *Proceedings of the 6th International Conference on Electronics, Communications and Control Engineering*, Aug. 2023.

[4] A. S. Mashaleh *et al.*, "Evaluation of machine learning and deep learning methods for early detection of internet of things botnets," *International Journal of Electrical and Computer Engineering*, Aug. 2024.

[5] M. Albanese *et al.*, "Adaptive Cyber Defenses for Botnet Detection and Mitigation," in *Adversarial and Uncertain Reasoning for Adaptive Cyber Defense*, 2019.

[6] C. L. Staudt *et al.*, "Generating realistic scaled complex networks," *Applied Network Science*, Oct. 2017.

[7] L. Böck *et al.*, "An Overview of the Botnet Simulation Framework," *The Journal on Cybercrime & Digital Investigations*, Dec. 2020.

[8] C. Ebojoh and A. Yeboah-Ofori, "Agent Based Simulation of Botnet Volumetric and Amplification Attack Scenarios Applied to Smart Grid Systems," in *Proceedings of the 4th International Conference on Intelligent Engineering and Management (ICIEM)*, May 2023.

[9] C. Dietz *et al.*, "DMEF: Dynamic Malware Evaluation Framework," in *IEEE/IFIP Network Operations and Management Symposium*, Apr. 2022.

[10] T. Li *et al.*, "Generative AI Empowered Network Digital Twins: Architecture, Technologies, and Applications," *ACM Computing Surveys*, 2025.

[11] C. Dietz *et al.*, "How to Achieve Early Botnet Detection at the Provider Level?" in *Proceedings of the 10th IFIP WG 6.6 International Conference on Management and Security in the Age of Hyperconnectivity*, vol. LNCS-9701. Springer International, Jun. 2016.

[12] A. Prasad and S. Chandra, "BotDefender: A Collaborative Defense Framework Against Botnet Attacks using Network Traffic Analysis and Machine Learning," *Arabian Journal for Science and Engineering*, 2024.

[13] A. Sperotto *et al.*, "An Overview of IP Flow-Based Intrusion Detection," *IEEE Communications Surveys & Tutorials*, 2010.

[14] C. Dietz *et al.*, "IoT-Botnet Detection and Isolation by Access Routers," in *Proceedings of the 9th International Conference on the Network of the Future*. IEEE, 2018.

[15] M. Asadi *et al.*, "Botnets Unveiled: A Comprehensive Survey on Evolving Threats and Defense Strategies," *Transactions on Emerging Telecommunications Technologies*, 2024.

[16] J. Steinberger *et al.*, "Collaborative DDoS defense using flow-based security event information," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, Apr. 2016.

[17] R. Anghel *et al.*, "Peering into the Darkness: The Use of UTRS in Combating DDoS Attacks," in *Proceedings of the 28th European Symposium on Research in Computer Security*, 2024.

[18] C. Fernandes *et al.*, "Cloud Native Manifesto," 2025. [Online]. Available: https://www.ngmn.org/wp-content/uploads/NGMN_Cloud_Native_Manifesto.pdf

[19] The Cloud Native Computing Foundation, "Cloud native glossary." [Online]. Available: https://glossary.cncf.io/

[20] M. Ahmed *et al.*, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, Jan. 2016.

[21] S. Wang *et al.*, "Machine Learning in Network Anomaly Detection: A Survey," *IEEE*, 2021.

[22] M. M. Salim *et al.*, "A Blockchain-Enabled Secure Digital Twin Framework for Early Botnet Detection in IIoT Environment," *ResearchGate*, Apr. 2025.

[23] Y. Yigit *et al.*, "Digital Twin-Enabled Intelligent DDoS Detection Mechanism for Autonomous Core Networks," *IEEE Communications Standards Magazine*, Sep. 2022.

[24] L. Campanile *et al.*, "Computer Network Simulation with ns-3: A Systematic Literature Review," *Electronics*, vol. 9, no. 2, Feb. 2020.

[25] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[26] E. Babulak and M. Wang, "Discrete event simulation," *Aitor Goti (Hg.): Discrete Event Simulations. Rijeka, Kroatien: Sciyo*, 2010.

[27] C. Kolias *et al.*, "DDoS in the IoT: Mirai and other Botnets," *IEEE Computer*, vol. 50, 2017.

[28] H. Adalsteinsson *et al.*, "Using Emulation and Simulation to understand the Large-scale Behavior of the Internet," Sandia National Laboratories, Tech. Rep., Oct. 2008. [Online]. Available: 10.2172/1130403

[29] F. Ramalho and A. Neto, "Virtualization at the network edge: A performance comparison," in *Proceedings of the 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, Jun. 2016.

[30] E. Petersen and M. Antonio To, "DockSDN: A hybrid container-based software-defined networking emulation tool," *International Journal of Network Management*, vol. 32, no. 2, 2022.

[31] W. Du *et al.*, "SEED Emulator: An Internet Emulator for Research and Education," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, New York, NY, USA, 2022.

[32] T. Holterbach *et al.*, "An open Platform to teach how the Internet practically works," *SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 2, May 2020.

[33] F. Niehaus *et al.*, "Modern ICT Network Simulator for Co-Simulations in Smart Grid Applications," *International Conference on Cyber Warfare and Security*, vol. 17, no. 1, Mar. 2022.

[34] D. Borsatti *et al.*, "KubeTwin: A Digital Twin Framework for Kubernetes Deployments at Scale," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, 2024.