Internet in a Box: Scaling Hands-On Education with the SEED Emulator

Wenliang Du*
Zhejiang University
Hangzhou, China
wenliangdu@zju.edu.cn

Chunling Yang
Zhejiang University
Hangzhou, China
chly@zju.edu.cn

Qishu Jin
Zhejiang University
Hangzhou, China
brucejin@zju.edu.cn

Abstract

Many important topics in cybersecurity, blockchain, and networking are difficult to teach through small, isolated lab environments. Routing attacks, DNS trust chains, malware propagation, blockchain consensus, and inter-domain routing policies depend on interactions among many hosts, routers, services, and administrative domains. Students therefore need hands-on activities that expose large-scale infrastructure behavior, but such activities are difficult to deploy safely and repeatably in ordinary classrooms.

This paper describes a collection of lab activities and demonstrations that use the SEED Internet Emulator as an "Internet in a box." The activities span three areas. In network security, students study BGP attacks, DNSSEC, Morris Worm, and controlled reenactments of famous Internet incidents such as BGP prefix hijacking, Mirai, WannaCry, SQL Slammer, NTP amplification, and Smurf attacks. In blockchain, students use private multi-node blockchains to study how blockchain works and how attacks on smart contracts work. In networking, the emulator supports scenarios involving network protocols, such as BGP and OSPF.

Since 2022, these activities have been deployed across seven offerings of two courses, reaching 360 students, and have also been used in instructor workshops and public cybersecurity demonstrations. Our experience suggests that emulator-based labs can make Internet-scale phenomena accessible on ordinary student machines while preserving safety, repeatability, and curricular flexibility.

CCS Concepts

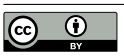
• **Security and privacy → Security protocols**.

Keywords

network emulation, computing education, cybersecurity, blockchain

ACM Reference Format:
Wenliang Du, Chunling Yang, and Qishu Jin. 2027. Internet in a Box: Scaling Hands-On Education with the SEED Emulator. In *Proceedings of the 58th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2027), February 17–20, 2027, Sacramento, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3857222.3860176

*Also with Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China.



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGCSE TS 2027, Sacramento, CA, USA
© 2027 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-3075-7/2027/02
https://doi.org/10.1145/3857222.3860176

1 Introduction

Hands-on activities are central to computing education, but some of the most important topics in cybersecurity, blockchain, and networking are difficult to teach with small, isolated lab environments. BGP prefix hijacking is not just a command typed on one router; it is an event that propagates through autonomous systems and changes how traffic flows across an Internet-like topology. DNSSEC is not simply a cryptographic operation; it depends on a chain of trust across resolvers, authoritative servers, parent zones, and the root. Blockchain concepts such as consensus, forks, transaction propagation, and eclipse attacks similarly depend on interactions among many nodes connected through a network. Students can learn the definitions of these mechanisms from lectures, but they often need a working infrastructure to see why the mechanisms matter and how failures unfold.

The challenge is scale. For many hands-on security labs, a small number of hand-crafted containers or virtual machines is sufficient. Such environments are effective for teaching local vulnerabilities, protocol details, or single-service attacks. However, they do not scale well to labs that require a miniature Internet with routers, autonomous systems, routing policies, DNS hierarchies, and blockchains. Manually constructing and configuring these environments is time-consuming for instructors and fragile for students. Physical networking laboratories and shared testbeds can provide high fidelity, but they often require dedicated hardware, reservations, and administrative support. Public blockchain testnets and single-node simulators provide useful development environments, but they give instructors limited control over topology, timing, failures, and attacks.

This paper examines the use of the SEED Internet Emulator as an "Internet in a box" for large-scale hands-on education [10]. SEED is an open-source, container-based emulator that allows a lab developer to construct a miniature Internet using the provided building blocks, consisting of essential elements of the Internet. The emulator then generates Docker-based emulation files that students can build and run on their personal computer, hence achieving "Internet in a box". This model separates lab construction from lab execution: instructors can create complex, reproducible Internet-like scenarios once, while students interact with the resulting environment through standard tools such as terminals, browsers, and scripts.

We have developed a variety of educational activities made possible by the SEED emulator. These activities cover three learning categories, including network security, blockchain, and networking. This paper makes the following contributions:

- We identify the educational need for scalable, Internet-like lab environments and explain why many infrastructure-level

topics are difficult to teach using only isolated virtual machines, public services, or physical testbeds.
- We describe how an "Internet in a box" approach can support hands-on activities across network security, blockchain, and networking while preserving reproducibility, containment, and student access on ordinary computing resources.
- We present concrete lab and demonstration designs enabled by SEED, including infrastructure-scale security labs, historical attack reenactments, blockchain labs, and networking lab that require multiple routers and autonomous systems.
- We report deployment experience since 2022 across course offerings, instructor workshops, and public demonstrations, summarizing practical lessons about setup, resource requirements, classroom use, and outreach.

The rest of the paper is organized as follows. We first discuss related work in hands-on cybersecurity education, network emulation, shared testbeds, blockchain education, and experiential learning. We then provide background on the SEED Internet Emulator and describe its use in cybersecurity, blockchain, and networking activities. Finally, we report deployment experience from courses, workshops, and public demonstrations, and conclude with lessons learned and future directions. We present this work as an experience report rather than a controlled learning-outcomes study; our evidence comes from repeated deployments, resource measurements, instructor observations, and public demonstration feedback.

2 Related Work

This work is related to several lines of hands-on computing education. We focus on the comparisons along these lines.

Hands-on cybersecurity labs. Hands-on cybersecurity education includes vulnerable virtual machines, guided labs, capture-the-flag challenges, and cyber-range scenarios. Prior surveys emphasize the value of concrete practice with attacks, defenses, and system behavior [26]. The original SEED Labs project is especially relevant: it provides a widely adopted collection of hands-on security labs [9]. Most original SEED labs, however, use small, hand-crafted environments with a few virtual machines or containers, which is effective for topics such as buffer overflows, web vulnerabilities, cryptography, access control, and many network attacks. Similarly, CTF-style exercises are valuable for learning in cybersecurity [6, 25, 27], but they also usually use a small setup. Our labs build on the existing tradition but target activities that require an Internet-like environment with many hosts, routers, and autonomous systems. These activities look artificial and less interesting if they are conducted in a small-scale setup.

Cyber ranges and networking testbeds. Cyber ranges and security testbeds such as KYPO and DETER provide realistic environments for attack-defense training, incident response, and security exercises [3, 28, 31]. These systems are valuable when a course needs centralized infrastructure, team exercises, or high-fidelity security operations. Our work is complementary: the labs in this paper are packaged as course materials that each student can run locally, without relying on a shared cyber range, hardware reservation, or central operator. This local deployment model lets students repeat experiments, reset their own Internet-like environment, and inspect routers, hosts, blockchain nodes, and services without affecting classmates.

Networking education has also used emulators and testbeds to expose students to working networks. Mininet supports software-defined networking experiments on a single machine [16], while Mini-Internet teaches autonomous systems, routing policy, and interdomain connectivity [12]. Tools such as Kathara, Containerlab, and GNS3 support hands-on network scenarios [5, 19, 23], and shared testbeds such as Emulab, GENI, CloudLab, and FABRIC provide access to larger research and teaching infrastructure [2, 4, 20, 29, 30]. We select SEED as our base, because of its extensible design and ever-growing ecosystem. Networking is only one layer of the emulation in SEED, new layers can be added and are being constantly added by the developer team. These layers include DNS infrastructure, public key infrastructure, darknet, botnet, and blockchain. For blockchain, in addition to Ethereum, two additional types of blockchain, Monero and Solana, were added recently. Due to the richness of the supported systems, the SEED emulator can be the foundation that serves a variety of course topics.

Blockchain education. Blockchain courses commonly use public testnets, local development tools, or manually configured private networks [18]. These approaches are useful for teaching wallets, transactions, and smart-contract programming, but they often give instructors limited control over topology, timing, and attacks. Our blockchain labs give each student a private multi-node blockchain that runs over an emulated network. Students can still use standard tools such as MetaMask, Remix, Web3, and Geth, but they can also observe transaction propagation, block creation, synchronization, partitions, performance effects and attacks.

Historical reenactments. Many security courses teach famous Internet incidents through lectures, readings, or case studies. We have developed activities that reenact historical attacks inside a controlled emulated Internet. This collection of activities is called *Yesterday Once More*. Historical attack reenactments are also related to experiential learning, scenario-based cybersecurity training, and serious games. Experiential learning theory argues that learners develop understanding through concrete experience and reflection [8, 15]. Cyber ranges and scenario-based training environments provide realistic settings for hands-on cybersecurity exercises [28, 31], while systems such as CyberCIEGE [14] use interactive simulation to help learners observe the consequences of security decisions. The *Yesterday Once More* activities differ in focus: rather than presenting generic training scenarios, they realistically recreate well-known Internet incidents inside a controlled emulated Internet, allowing learners to experience and interact with these historical incidents.

3 Background: The SEED Emulator

The SEED Internet Emulator is an open source project with the aim to construct an emulated Internet that can run on a single machine [10]. It is designed around reusable building blocks, which are implemented as Python classes for essential Internet elements, including hosts, routers, networks, autonomous systems, Internet exchange points, and links. Lab developers compose these classes to create hosts and networks, connect them into topologies, install and

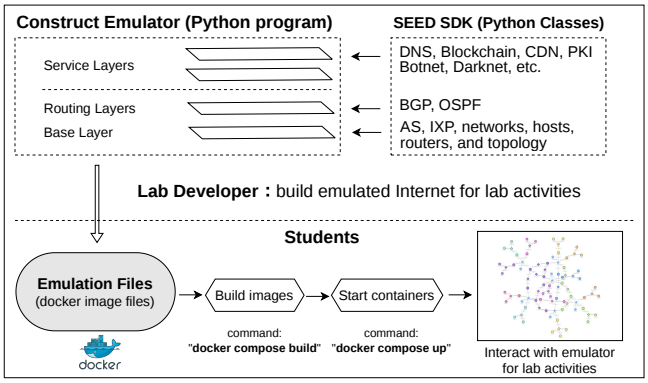
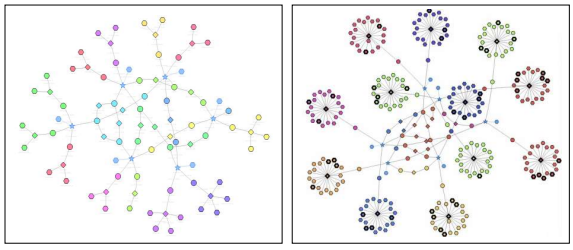


Figure 1: SEED Emulator workflow

configure software on hosts, and set up routing behavior on routers. SEED also provides service components for important Internet services. After the emulator is constructed, a compiler generates the final Docker-based container image files that students can build and run. Figure 1 shows this workflow.

To develop a lab that uses an Internet-like environment consisting of many nodes and network, the lab developer writes a Python program using classes from the SEED SDK. The program constructs the emulated Internet in layers. The base layer defines autonomous systems, networks, hosts, routers, Internet exchange points, and topology. Routing layers set up the routing infrastructure, including EBGP/IBGP peerings, network prefix announcements, and OSPF routing protocols. Service layers then add higher-level Internet services, such as DNS, blockchain, Content Delivery Network (CDN), Public Key Infrastructure, botnets, email, and darknets. The developer then develops lab activities based on the generated network.

Students do not need to manually configure every router, host, or service in the lab environment. They simply start the lab environment using standard Docker Compose commands. Once the containers are running, students interact with the emulated Internet through terminals, browsers, blockchain tools, and other standard software used by the lab. The emulator comes with a built-in visualization tool called InternetMap, which visualizes the network and packet flows. Two screenshots of the InternetMap visualization are given in Figure 2. Each node on the map represents a container, and users can click on each node to get a terminal to that container.



(a) BGP Attack Lab (b) Morris Worm Lab

Figure 2: Example topologies used in labs

This separation between scenario construction and student execution is central to the emulator's educational value. Instructors can create complex Internet scenarios once and reuse them across semesters, while students receive a reproducible environment that can run on ordinary machines. The same architecture supports the three application areas described in this paper: network security and attack reenactments, blockchain, and networking activities involving routing protocols and Internet-scale topology.

4 Educational Applications

In this paper, we explore how the SEED emulator can be used as the foundation for hands-on learning. We have developed lab activities in three fields: network security, blockchain, and networking. We describe these activities in this section, and then report our classroom and deployment experience in the next section.

4.1 Network Security

Many network security concepts are difficult to teach because they are not properties of a single host; they emerge from interactions among routing systems, DNS infrastructure, vulnerable services, malware, and human operators. Real attacks require realistic infrastructure, but using the real Internet for instruction is unsafe and ethically unacceptable. Conversely, overly simplified lab environments can strip away the distributed mechanisms that make these attacks meaningful, reducing both the realism of the exercise and students' sense that they are studying real Internet security problems.

4.1.1 Network Security Labs. Our security labs use SEED to expose students to infrastructure-level security mechanisms that are difficult to reproduce with conventional virtual machines. For network security, we have developed four labs: (1) BGP exploration and attack lab, (2) Morris worm attack lab, (3) DNS infrastructure lab, and (4) DNSSEC lab. Due to page limitation, we only give details to the first two labs. The last two labs, DNS infrastructure and DNSSEC, give students a complete DNS hierarchy in an emulated Internet, so students can learn how DNS works, attacks on DNS and the DNSSEC defense mechanism.

In the BGP exploration and attack lab, students work with an emulated Internet containing multiple autonomous systems and real routing software. They observe BGP update messages, configure routing policies, study path selection and anycast, and finally launch and mitigate a network prefix-hijacking attack. The lab therefore connects routing concepts with their security consequences in a setting where students can directly manipulate the control plane. The topology used in this lab is shown in Figure 2a. It contains 33 hosts, 41 routers, and 36 networks. With a little bit of learning of how to write emulator code, instructors can modify the provided emulator construction program to change the topology or conduct their specific customization.

In the Morris worm [24] attack lab, students focus on learning how worms self-duplicate and spread on the Internet. We create an emulated Internet consisting of 275 containers (see Figure 2b). Most of the containers run a vulnerable server that can be exploited. Part of the worm code is provided to students, but the self-duplication and spreading logic is intentionally omitted. Students' job is to finish these parts, and release their worm on one of the containers. Their goal is to get the worm automatically spread to the entire emulated Internet. On the InternetMap visualization tool, infected nodes will flash so students can see the results.

4.1.2 "Yesterday Once More" Reenactments. Many historically important Internet attacks are commonly taught through lectures, readings, or postmortem analyses. While these approaches explain what happened, they rarely allow students to observe the system-level dynamics that made the incidents possible. It will be more effective if we can let students experience in person what has happened, to gain the first-hand knowledge, and even to change the course of the event.

For this purpose, we created a collection of historical attack reenactments that we call *Yesterday Once More*. These scenarios recreate well-known Internet incidents inside a controlled emulated Internet, allowing students to observe how an attack unfolds across multiple systems and administrative domains. We have recreated seven famous attacks: Pakistan's Hijacking of YouTube (a BGP prefix hijacking) [22], the Mirai attack [1], the Morris Worm [24], the WannaCry ransomware [7], the SQL Slammer worm [17], the NTP amplification attack [21], and the Smurf attack [13]. More will be developed in the future.

We use the Mirai attack as an example to illustrate how these reenactments work. Mirai demonstrates how vulnerable Internet-connected devices can be scanned, compromised, enrolled into a botnet, and then coordinated to launch large-scale DDoS attacks. In our reenactment, we construct an emulated Internet with many hosts and services, including devices configured with weak or default passwords. We implement a simplified Mirai-like worm that scans the emulated Internet, logs in to vulnerable hosts, and installs a bot program on each compromised host. As the worm spreads, students can watch the botnet grow through the visualization tools and can inspect the infected hosts to see how compromise changes their behavior.

After enough bots have been enrolled, the attacker's command-and-control server sends a command to the botnet to launch a DDoS attack against a selected web server. Students can then observe the effect from both sides: the victim web server becomes difficult to access, while the attacker can see many bots sending traffic at the same time. This example shows why the emulator is essential to *Yesterday Once More*: it gives instructors control over the topology, vulnerable hosts, malware behavior, command-and-control channel, victim service, and observation points, while keeping the attack safely contained. Students see scanning, compromise, bot enrollment, command-and-control, and DDoS as one connected system, not isolated facts. The same scenario can be repeated, paused, modified, and discussed, which would be impossible on the real Internet and too artificial in a single-machine simulator.

4.2 Blockchain

Blockchain has become an important topic in distributed systems, cryptography, and cybersecurity curricula. Teaching it well is difficult because many core ideas are distributed and infrastructure dependent. Consensus, ledger immutability, transaction propagation, smart-contract execution, forks, and synchronization are hard to understand through lectures or single-node demonstrations alone. Students need to see independent nodes communicate, disagree, recover, and converge on shared state.

This need is especially important because blockchain systems span several layers of computing: the application layer, protocol layer, and infrastructure layer. Traditional teaching environments often expose only one layer at a time, leaving students with a fragmented view of deployed blockchain systems. Public testnets provide realistic interfaces but give instructors little control over the system. Local simulators such as Ganache and Hardhat are excellent for smart-contract debugging, but they hide the distributed peer-to-peer network. Manually configured private blockchains have higher fidelity, but setting up even a small private blockchain requires substantial work, let alone a larger one.

4.2.1 How SEED Emulator Helps. The SEED Internet Emulator addresses these limitations by automating the construction of private, multi-node blockchain networks. It implements an Ethereum service, which handles genesis-block distribution, account creation, node configuration, and peer-discovery setup. Lab developers can create a blockchain on top of an Internet topology with little effort. Figure 3 shows that a few lines of Python can create and customize a 20-node Ethereum blockchain (the code for generating the Internet topology is not included).

```
eth = EthereumService()
blockchain = eth.createBlockchain(consensus=ConsensusMechanism.POS)

nodes = []
total = 20 # Total number of Ethereum nodes (instructor can customize)
for i in range(total):
   e = blockchain.createNode(f'eth-{i}') # create an Ethereum node
   e.startMiner().enableGethHttp().unlockAccounts() # customize node
   nodes.append(e)

nodes[0].setBootNode(True) # set this node as a bootnode
nodes[5].setBootNode(True)
nodes[9].createAccount(balance=99*pow(10,5)) # create a pre-funded account
```

Figure 3: Create a private Ethereum blockchain using SEED

These nodes run real Ethereum client software (geth and lighthouse) on realistic Internet topologies. Users can interact with the blockchain using standard Ethereum tools, such as MetaMask, Remix, Web3, and Hardhat. SEED also provides a visualization tool that lets students inspect blocks and transactions.

4.2.2 Hands-on Labs. We have created a progression of blockchain labs that moves from foundations, to programming, to infrastructure reasoning. In the foundation lab, students connect MetaMask, Web3 programs, and geth to a private Ethereum network generated by SEED. They manage accounts, send transactions, query state, inspect blocks, and add full nodes. The goal is to help students become familiar with a blockchain platform, which is new to many of them. Students learn to use existing tools and implement their own programs to interact with this platform. Because the emulator creates a multi-node network and exposes its nodes to standard tools, students interact with a real distributed ledger rather than a single-process simulator.

The programming labs focus on smart contracts and use the same emulated blockchain as their base. We have developed two programming labs. The objective of the smart contract lab is to help students understand how smart contracts, a new type of program for many of them, work. Students write, deploy, and invoke contracts; distinguish local calls from state-changing transactions; transfer funds; and study contract-to-contract interaction. In the reentrancy attack lab, students are given a banking-style contract that contains a reentrancy vulnerability. Students first use Python Web3 scripts to deploy, fund, and interact with this victim contract. They next

deploy an attacker contract whose fallback function repeatedly re-enters the victim contract during the withdrawal process, with the goal of draining the victim contract's funds. Finally, they fix the victim contract and repeat the attack.

The three labs described above have been used across multiple course offerings. In addition to these labs, we are also experimenting with a new category of infrastructure labs. These labs expose the network beneath the blockchain and the blockchain's consensus protocol. Students can vary peer topology, introduce latency, change routing paths, and observe consequences for consensus, synchronization, majority-power attacks, and eclipse attacks [11]. The eclipse-attack lab, for example, isolates a victim node's peer connections to show how network-level control can distort the node's view of the blockchain. The Proof-of-Stake (PoS) lab examines Ethereum's complex PoS consensus protocol, helping students understand key aspects of the protocol, including validators, beacon nodes, endorsements, rewards, and penalties. These labs have not been fully evaluated; so far, they have only been used by a small group of students outside the classroom.

4.3 Networking

Although our classroom deployment so far has focused primarily on network security and blockchain, the SEED Internet Emulator also has strong potential for teaching computer networking. Many networking concepts are difficult for students to understand from diagrams alone because they depend on interactions among multiple routers, autonomous systems, routing policies, and traffic flows. The emulator can instantiate these components as a working Internet-like topology, allowing students to observe protocol behavior directly.

We have designed activities to cover a variety of networking topics, including autonomous systems (stub and transit), BGP, OSPF, EBGP and IBGP peering, MPLS forwarding, route reflector, route server, IP anycast, BGP filtering, and BGP communities. These topics normally require access to complex network infrastructure, but SEED allows them to be reproduced on standard machines in a controlled and repeatable form. We do not yet claim classroom evaluation for these activities, but we used them to train undergraduate and graduate students when they joined our research group.

5 Deployment and Experience

We have deployed the emulation-based activities in three settings: courses, instructor training, and public demonstrations.

5.1 Deployment in Courses

We have used the SEED Internet Emulator in instructional settings since 2022. As summarized in Table 1, the deployment includes seven offerings of two courses, reaching approximately 360 students. Five offerings were conducted in the United States and two in China.

Students used a pre-built virtual machine containing the needed software. Each VM was configured with 4 GB of RAM and 2 CPU cores and could run on Windows, macOS, or Linux hosts. Most students ran the emulator on personal laptops; a smaller number used cloud VMs when their devices could not run the provided image. Each lab was distributed as a separate emulator package. Students built the images once with Docker Compose and then started the containers for each lab session. Table 2 summarizes resource requirements for the representative labs. The image-building step may take more than 10 minutes because base Docker images must be pulled from DockerHub, but this cost is paid only once for each lab. Starting an already-built emulator is usually fast.

Table 1: Deployment of the lab activities (2022 - 2026)

Course	Student	Scale	Labs
Computer Security	Graduate and senior undergraduate students	3 offerings; 160 students total	Blockchain basic, smart contract, reentrance attack
Internet Security	Graduate and senior undergraduate students	4 offerings; 200 students total	DNSSEC, BGP attack, Morris Worm

Table 2: Resource requirement (for blockchain setup, the 57-node emulator runs a 15-node Ethereum blockchain)

Lab	Nodes	Memory	CPU (2 cores)	Building Time (s)	Start Time (s)
BGP/DNS	74	2.2 GB	13.8%	418	72
Morris	275	3 GB	29.9%	715	214
Blockchain	57	2.7 GB	33.0%	462	56

The deployments were feasible in all offerings. We report aggregate deployment observations and unsolicited public comments; no identifiable student data or graded performance data were collected. The main recurring issues were the followings. The first issue is cleanup. After finishing a lab, students sometimes forget to shut down the emulator properly, leaving behind containers, networks, or other state that can interfere with the next lab. To address this problem, we provide detailed cleanup and troubleshooting instructions that students can follow when such leftovers occur. The second recurring issue occurred in mainland China, where DockerHub access can be blocked; Docker proxies usually resolved the problem, though their availability varied. The third issues is related to computer memory. Some students' computers do not have enough memory, so they could not give their VMs 4 GB of RAM. For these students, we provide a smaller version of the emulator. This typically requires the instructor to modify a parameter in the emulator-generation code, so they can generate a customized emulator for these students.

Beyond our own courses, the emulator-based labs are distributed through the SEED Labs platform [9], which is used by over 1160 institutes in more than 80 countries. This broad SEED Labs adoption does not imply adoption of the new emulator-based activities, but it provides an existing distribution channel through which instructors can discover them. Due to this channel, 180 instructors have requested the manuals for the emulation-based labs. No data is available yet regarding how many instructors actually used the labs in their classrooms.

5.2 Yesterday Once More Demonstration

We brought the *Yesterday Once More* reenactment to the classroom. We set up a *shadow Internet*, as shown in Figure 4. Students connect

their smartphones to the provided WiFi access point, essentially attaching their phones to the networks inside the emulation. Their packets to the real Internet first get routed through the shadow Internet, before being released to the outside from the exit point.

While students were visiting the university website, we launched a BGP prefix-hijacking attack inside the emulator, hijacking the university's network prefix, redirecting the university-bound packets to a blackhole (packets to other destination were not affected). The attack has no effect on the real Internet; it only changes routing inside the shadow Internet, preventing the targeted packets from reaching the exit point. As a result, participants immediately found that they could no longer reach the university website. This form of participation made the demonstration more concrete than a screen-only presentation: the failure happened on the participants' own devices, while the cause remained visible inside the emulator and its visualization tools.

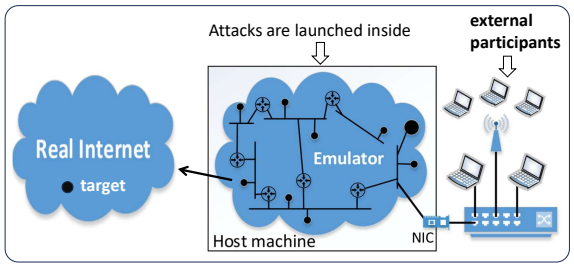


Figure 4: The emulator becomes a shadow Internet

After learning about this work, the university computing center invited us to set up a *Yesterday Once More* exhibition during Information Security Week. The exhibit demonstrated how BGP prefix hijacking and Mirai-style botnet attacks could affect university security, using the university's main website as the "target". During the 10-day event, we conducted 62 demonstrations for 2,008 attendees, including students, staff, faculty, and administrators from different schools. We trained several students to give the live demonstration. Some of the students were not computer science major, but they had no problem giving the demonstration after the training.

The exhibition was well received. Audience comments described the demonstration as "well designed," "a realistic experience," and "a treasure trove of information," and included reflections such as "the Internet is complex" and "cybersecurity is all around us." These comments were collected from a public feedback board and are reported only as illustrative reactions. The demonstration later received the 2025 Creative Works and Practical Cases of Cybersecurity Education award from China's education authority. These comments were informal, but they suggest that the reenactment helped non-specialist audiences connect Internet infrastructure failures with everyday security concerns.

5.3 Deployment in Training Workshops

We also organized two instructor workshops in China, in 2023 and 2025, with a total of 40 faculty participants. The workshops focused on hands-on cybersecurity education and included two emulator-based activities: the BGP attack lab and the Blockchain reentrance attack lab. Each activity had to fit within 1.5 to 2 hours.

The average completion rate across the two labs was about 80%. The completion rate was recorded by workshop facilitators as an aggregate operational measure, not as individual participant performance data. The main obstacle was again Docker image access in mainland China; some participants had to configure alternative proxies or use VPNs. Although Docker proxies can address the issue, their reliability varied over time: proxies that worked during preparation were sometimes unavailable during the workshop. Overall, the results were encouraging. Several instructors noted that the BGP lab was their first opportunity to interact directly with real BGP routers, observe routing behavior, and launch a prefix-hijacking attack in a controlled environment. For the Blockchain reentrance attack lab, many said that they had previously taught the attack only as a historical story; with SEED, they could launch it inside an emulated Internet, on a real blockchain.

6 Conclusion and Future Work

Many important topics in cybersecurity, blockchain, and networking are difficult to teach because they are properties of systems, not individual machines. Routing attacks, DNS trust chains, botnet propagation, blockchain consensus, and inter-domain routing policies all require students to reason about interactions among many hosts, routers, services, and administrative domains. This paper described how the SEED Internet Emulator can support this kind of instruction by packaging an Internet-like environment into reproducible, container-based labs that students can run on ordinary machines. The labs cover three subject areas: network security, blockchain, and networking.

The deployments since 2022 suggest that this approach is practical in courses, instructor workshops, and public demonstrations, while still requiring attention to setup support, resource limits, and regional access issues such as Docker image availability. The experience also shows that scalability is curricular as well as technical: once the environment can be generated and distributed reliably, instructors can design activities around phenomena that were previously too large, risky, or fragile for ordinary classrooms.

In our future work, we plan to expand the collection of emulator-based labs, especially for infrastructure-level blockchain security and advanced networking. Planned activities include eclipse attacks, additional consensus and fork experiments, BGP policy exercises, and labs involving route servers, MPLS, filtering, and BGP communities. We also plan to study how much infrastructure detail should be exposed at different course levels. Beginning students may benefit from simplified interfaces and guided observations, whereas advanced students can work directly with router configuration, blockchain clients, logs, and visualization data. Supporting this range of scaffolding will make the same emulation foundation useful across demonstrations, upper-level labs, graduate courses, and instructor training.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. We also thank Ant Group, the State Key Laboratory of Blockchain and Data Security at Zhejiang University, and the Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security for their support. The early stages of this work were supported by the U.S. National Science Foundation (NSF).

References

[1] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *Proceedings of the 26th USENIX Security Symposium (USENIX Security '17)*. USENIX Association, Vancouver, BC, Canada, 1093–1110.

[2] Ilya Baldin, Anita Nikolich, James Griffioen, Indermohan Inder S. Monga, Kuang-Ching Wang, Tom Lehman, and Paul Ruth. 2019. FABRIC: A National-Scale Programmable Experimental Network Infrastructure. *IEEE Internet Computing* 23, 6 (2019), 38–47. doi:10.1109/MIC.2019.2958545

[3] Terry Benzel. 2011. The science of cyber security experimentation: the DETER project. In *Proceedings of the 27th Annual Computer Security Applications Conference* (Orlando, Florida, USA) *(ACSAC '11)*. Association for Computing Machinery, New York, NY, USA, 137—-148. doi:10.1145/2076732.2076752

[4] Mark Berman, Jeffrey S. Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. 2014. GENI: A federated testbed for innovative network experiments. *Computer Networks* 61 (2014), 5–23. Special issue on Future Internet Testbeds – Part I. doi:10.1016/j.bjp.2013.12.037

[5] Gaetano Bonofiglio, Veronica Iovinella, Gabriele Lospoto, and Giuseppe Di Battista. 2018. Kathará: A container-based framework for implementing network function virtualization and software defined networks. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–9.

[6] Peter Chapman, Jonathan Burket, and David Brumley. 2014. picoCTF: A Game-Based Computer Security Competition for High School Students. In *Proceedings of the 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE '14)*. USENIX Association, San Diego, CA, USA.

[7] Qian Chen and Robert A. Bridges. 2017. Automated Behavioral Analysis of Malware: A Case Study of WannaCry Ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 454–460. doi:10.1109/ICMLA.2017.0-119

[8] John Dewey. 1938. *Experience and Education.* Macmillan, New York, NY, USA.

[9] Wenliang Du. 2015. SEED Labs: Using Hands-on Lab Exercises for Computer Security Education (Abstract Only). In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) *(SIGCSE '15)*. Association for Computing Machinery, New York, NY, USA, 704. doi:10.1145/2676723.2678290

[10] Wenliang Du, Honghao Zeng, and Kyungrok Won. 2022. SEED emulator: an internet emulator for research and education. In *Proceedings of the 21st ACM workshop on hot topics in networks*. 101–107.

[11] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse attacks on Bitcoin's peer-to-peer network. In *Proceedings of the 24th USENIX Conference on Security Symposium* (Washington, D.C.) *(SEC'15)*. USENIX Association, USA, 129—-144.

[12] Thomas Holterbach, Tobias Bü, Tino Rellstab, and Laurent Vanbever. 2020. An Open Platform to Teach How the Internet Practically Works. *SIGCOMM Comput. Commun. Rev.* 50, 2 (May 2020), 45—-52. doi:10.1145/3402413.3402420

[13] Craig A. Huegen. 1997. The Latest in Denial of Service Attacks: "Smurfing". https://www.pentics.net/denial-of-service/white-papers/smurf.cgi.

[14] Cynthia E. Irvine, Michael F. Thompson, and Ken Allen. 2005. CyberCIEGE: Gaming for Information Assurance. *IEEE Security & Privacy* 3, 3 (2005), 61–64.

[15] David A. Kolb. 1984. *Experiential Learning: Experience as the Source of Learning and Development.* Prentice Hall, Englewood Cliffs, NJ, USA.

[16] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (Monterey, California) *(Hotnets-IX)*. Association for Computing Machinery, New York, NY, USA, Article 19, 6 pages. doi:10.1145/1868447.1868466

[17] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. 2003. Inside the Slammer Worm. *IEEE Security and Privacy* 1, 4 (July 2003), 33–39. doi:10.1109/MSECP.2003.1219056

[18] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Princeton University Press, Princeton, NJ, USA.

[19] The Containerlab Project. [n. d.]. Containerlab. https://github.com/srl-labs/containerlab. Accessed: 2026-04-24.

[20] Robert Ricci, Eric Eide, and CloudLab Team. 2014. Introducing CloudLab: Scientific Infrastructure for Advancing Cloud Architectures and Applications. *;login: The USENIX Magazine* 39, 6 (Dec 2014). https://www.usenix.org/publications/login/dec14/ricci

[21] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)* (22 Feb). https://www.ndss-symposium.org/ndss2014/ndss-2014-programme/amplification-hell-revisiting-network-protocols-ddos-abuse/

[22] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. 2018. A Survey among Network Operators on BGP Prefix Hijacking. *SIGCOMM Comput. Commun. Rev.* 48, 1 (April 2018), 64—-69. doi:10.1145/3211852.3211862

[23] LLC SolarWinds Worldwide. 2022. GNS3: The Software that Empowers Network Professionals. Available at https://www.gns3.com/.

[24] Eugene H. Spafford. 1989. The internet worm program: an analysis. *SIGCOMM Comput. Commun. Rev.* 19, 1 (Jan. 1989), 17–57. doi:10.1145/66093.66095

[25] Valdemar Švábenský, Jan Vykopal, Milan Cermak, and Martin Laštovička. 2021. Cybersecurity Knowledge and Skills Taught in Capture the Flag Challenges. *Computers & Security* 102 (2021), 102154.

[26] Valdemar Švábenský, Jan Vykopal, and Pavel Čeleda. 2020. What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 2—-8. doi:10.1145/3328778.3366816

[27] Jan Vykopal, Valdemar Švábenský, and Ee-Chien Chang. 2020. Benefits and Pitfalls of Using Capture the Flag Games in University Courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 752–758.

[28] Jan Vykopal, Pavel Čeleda, Pavel Seda, Valdemar Švábenský, and Daniel Tovarňák. 2021. Scalable Learning Environments for Teaching Cybersecurity Hands-on. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–9. doi:10.1109/FIE49875.2021.9637180

[29] Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar. 2003. An integrated experimental environment for distributed systems and networks. *SIGOPS Oper. Syst. Rev.* 36, SI (Dec. 2003), 255—-270. doi:10.1145/844128.844152

[30] Alexander Wolosewicz, Prajwal Somendyapanahalli Venkateshmurthy, and Nik Sultana. 2025. Experience Report: Using the FABRIC Testbed to teach a Graduate Computer Networking course. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1246—-1252. doi:10.1145/3641554.3701923

[31] Muhammad Mudassar Yamin, Basel Katt, and Vasileios Gkioulos. 2020. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. *Comput. Secur.* 88, C (Jan. 2020), 26 pages. doi:10.1016/j.cose.2019.101636