Exploring Undergraduate Research Opportunities through Senior Projects

Dr. Bingyu Liu, Wentworth Institute of Technology

Dr. Bingyu Liu (she/her) is an assistant professor at the department of School of Computing & Data Science at Wentworth Institute of Technology. She was a postdoctoral research fellow at the Institute for Experiential AI (EAI) in Northeastern University. She received her PhD in Computer Science from Illinois Institute of Technology in 2022. Her work focuses on research and education in security and privacy, emphasizing secure computation, applied cryptography, and differential privacy, while also exploring interdisciplinary connections such as trustworthy AI, health care and theoretical foundations.

Dr. weijie pang, Wentworth Institute of Technology

Weijie Pang is a tenure-track assistant professor in School of Computing and Data Science at Wentworth Institute of Technology. After received Ph.D. from Worcester Polytechnic Institute and B.S. and M.S. from Beijing University of Technology, Dr. Pang finished the Post-doc program in Department of Mathematics and Statistics at McMaster University. Her research area is applied data science, financial mathematics, pandemic modeling and stochastic analysis.

Exploring Undergraduate Research Opportunities through Senior Projects

Abstract

Engaging undergraduates in research fosters curiosity, creativity, and independent thinking, yet several challenges limit its accessibility and effectiveness. A key issue is the gap between regular coursework and research work, where students may complete team-based projects without mastering the skills needed for independent research. Limited resources further restrict meaningful research experiences for both students and faculty, reducing the potential for sustained mentorship. To better understand how students develop research capabilities, this paper presents a case study that follows one student's progression from a Senior Project to a completed research paper that was published and presented at a cybersecurity conference. The project involves designing a comprehensive open-source IRC botnet for penetration testing that simulates realistic attack scenarios. Students begin with limited knowledge, then initialize proposals, form teams, receive instructor feedback, and follow structured weekly plans. Regular meetings, experiment design, evaluation metrics, and mentorship enable students to achieve their research goals. Results show the project meets its objectives, and the completed paper is currently presented and published in a research track conference. More generally, the proposed approach reflects a scalable model for teaching undergraduate research methods. Structured proposal development, iterative feedback, team-based collaboration, and conference-level dissemination form a pedagogical sequence that can be adapted across disciplines. Such practices align with established models in undergraduate research training that emphasize mentorship, authentic problem solving, and exposure to the full research lifecycle. By studying how students develop skills within these structured environments, this work offers insights into how institutions can bridge the gap between coursework and independent research while accommodating resource limitations.

Key words: Undergraduate research, Senior project, Cybersecurity, Professional development, IRC Botnet Detection, Network Security

Introduction

In terms of STEM education[1], undergraduate research plays a significant role in shaping students' academic and future career pathways. Many programs encourage undergraduate participation in research activities to foster critical thinking and problem-solving skills, promote engagement with emerging technologies, and facilitate the application of theoretical knowledge to real-world challenges. Such experiences enhance students' academic profiles by strengthening background, and cultivating transferable skills, including critical thinking, academic writing, and oral presentation. Beyond disciplinary knowledge acquisition, studies indicate that undergraduate involvement in research is associated with increased likelihood of pursuing advanced degrees and enhanced career preparedness, underscoring the importance of undergraduate research opportunities in supporting long-term academic and professional development.

Despite the emphasis on experiential learning in STEM curricula[1], a persistent gap remains between coursework-based knowledge and the expectations of research projects. This discrepancy often limits students' ability to independently manage, and sustain research activities, particularly when prior exposure to research practices is minimal. Expanding opportunities for students to engage in team-based projects and to develop competencies essential for independent research, such as project design, collaboration, and self-driven which is therefore critical. Moreover, limited access to the resources further restricts meaningful research engagement for both students and faculty. Addressing this gap requires object-oriented curricular design and faculty-supports to better align coursework with research-based learning outcomes. Although many students are interested in participating in research projects, they often lack the foundational knowledge to start and may not feel comfortable with research training. This lack of preparation creates barriers that limit full engagement in research experiences and ultimately undermines students' motivations for advanced education and careers in STEM fields.

In this paper, we address these challenges by examining a specific case study on the development of a comprehensive botnet designed to test the security measures of a computer network. To delve deeper into the challenges, we present a case study showcasing students' progression from senior projects to published research. Based on this case study, we illustrate how to define clear research objectives, integrate research into coursework, guide students through the research process step by step, and provide adequate resources and mentorship. In our case study, students begin their projects by submitting a proposal on IRC Botnet with limited prior knowledge. Instructors then provide feedback based on the students' backgrounds and collaborate with them to develop a weekly progress plan. The project, designed and implemented as an open-source tool, is an IRC botnet intended for penetration testing, offering a versatile and realistic testing environment. This stage ensures that students can find teammates to collaborate with and establish an appropriate timeline for designing and implementing project features and functionalities that leverage an IRC C&C structure to simulate realistic attack scenarios. Regular meetings during this phase are essential to ensure that all team members share a common understanding of the project goals. More importantly, instructors support students in designing experiments that integrate multiple evaluation metrics while fostering a systematic, step-by-step approach to problem solving. The evaluation results suggest that the final project outputs demonstrate meaningful progress thus far and provide direction for subsequent future work. More broadly, the approach illustrated in this case represents a generalizable model for teaching undergraduate research in the STEM education. These practices align with established approaches to undergraduate research training, which emphasize mentorship, problem solving, and engagement with the research lifecycle. By documenting how students develop research competencies within structured environments, this work offers insights into how institutions can bridge the gap between coursework and independent research while addressing resource constraints.

Challenges to Undergraduate Research Participation

Prior to engaging in research, many students encounter unforeseen challenges that may discourage them from pursuing research opportunities. Students who enter university with a primarily job-oriented focus may pursue goals that differ from those centered on gaining research experience and developing a deeper understanding of their discipline. While immediate practical training may not always align with research participation, undergraduate research offers substantial long-term academic benefits. These benefits include a deeper understanding of course material through the application of theory to real-world problems, the development of critical thinking and analytical skills, and improvements in academic writing, data analysis, and presentation skills. Moreover, for students interested in advanced study, undergraduate research offers strong preparation for graduate education and research-intensive academic career pathways.

However, there are many challenges that may impede progress. Insufficient foundational knowledge and research skills can present significant barriers that limit students' ability to progress and fully benefit from research experiences. In the following section, we examine the challenges encountered by students and discuss active learning strategies that were implemented to help mitigate these barriers.

- *Lack of Motivation and Interest in Research:* A common challenge at the undergraduate level is limited motivation or interest in research activities. Many students prioritize course or job-oriented projects they perceive as immediately interesting and meaningful, rather than engaging in foundational research tasks that may initially appear routine or abstract. In the absence of clear and well-defined objectives and outcomes, students may experience reduced motivation and uncertainty about their roles and level of involvement, which can hinder sustained engagement in research. Achieving self-motivated learning requires that learners actively monitor and adjust their approaches to learning.
- *Insufficient Foundational Knowledge and the gaps in understanding:* Many undergraduate students lack adequate foundational knowledge and essential research skills, including academic critical thinking, scholarly literature review, experimental design and implementation, and research presentation.
- *Time Management Challenges:* When faced with multiple concurrent responsibilities, students have no idea how to balance coursework, examinations, part-time employment, and personal commitments alongside the demands of research activities.
- *Feedback Challenges:* Feedback latency and unclear expectations from mentors can hinder students' confidence and engagement in research activities, underscoring the need for structured mentorship practices and clearer communication to support undergraduate researchers effectively.
- *Limited Access to Research Resources:* Restricted access to laboratories, datasets, specialized software, and academic research papers presents a significant barrier for undergraduate research, particularly in STEM disciplines. In some fields, the limited availability of advanced equipment, such as graphical processing units (GPUs) required for computer programs in Cybersecurity, Machine Learning, and Deep Learning, which

constrains students' ability to conduct meaningful and computationally intensive research. Instructions may be necessary for the instructor to provide the required help and documents, or to give a demonstration if necessary.

Despite these considerable barriers, the student was able to reduce their fears through the implementation of targeted strategies. More importantly, learning is part of the training process, students need to be prepared for unforeseen challenges and encouraged to step outside their comfort zone. By addressing these limitations and challenges through the **EUR** (Exploring Undergraduate Research) structured teaching method, the results suggest that students are more inclined to pursue further research rather than becoming discouraged and frustrated by the research process.

Given the limited time and detailed instructions, students are expected to work independently and exhibit a high degree of self-motivation. The project consists of three phases: Initialization, Design, and Implementation. By encouraging undergraduate research, faculty not only strengthen the students' academic background but also their academic research and professional network, and foster innovation grounded in the scientific method.

Methodology: EUR Framework

In our work, we adopt a structured method **EUR**, to help undergraduate students reconstruct their prior knowledge and apply it to real-world problems. More importantly, fostering *self-directed learning* motivates students and enables their research to have greater impact. This framework integrates into the academic curriculum, ensuring that research opportunities become a core component of students' educational experience. Students start with the project. It provides students with the opportunity to initialize, design and implement solutions to large-scale projects in a team-based environment. Projects are encouraged to be interdisciplinary. For example, combining AI with Cybersecurity. Students are required to produce A comprehensive end-of-project report and deliver oral presentations on their projects.

EUR Framework:

- Achieving *self-directed learning* requires that learners assess the demands of a task, evaluate their own knowledge and skills, plan their approach, monitor their progress, and adjust their strategies as needed. Weekly reports are required to document and track students' progress. As a study shows that Students who naturally monitor their own progress and explain what they are learning along the way generally show greater gains in self-monitoring skills.
- Monitor performance and provide timely feedback based on students' weekly reports, with particular attention to their research progress.
- Structured and systemic paper-reading plan, which aims to develop skills for reading research papers to ensure sufficient input for high-quality research output.

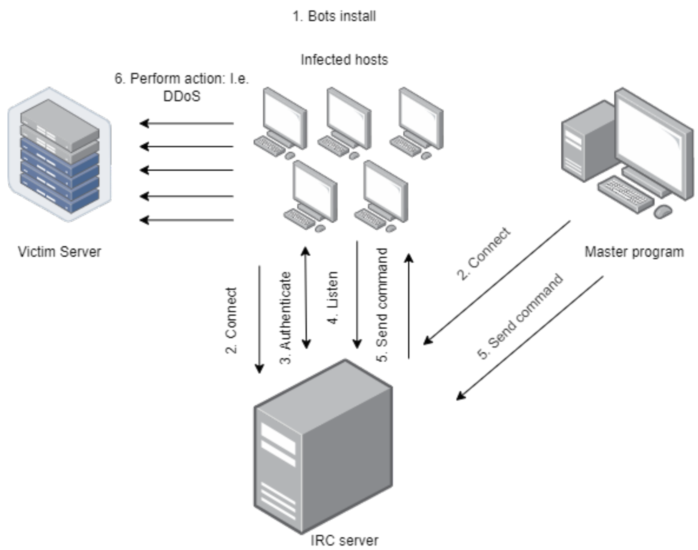


Figure 1: Student Sample Work for Workflow of CRYNET, the Bots and Master connect and authenticate and then begin to communicate. At the micro level, careful design of each component was required to achieve a seamless macro-level workflow. The workflow can be demonstrated in the protocol that articulates how the System runs on a high level.

Initialization Phase

Project proposal Students begin with a project proposal. The purpose of the proposal is to provide a general overview of the project, helping teams define the scope and giving the instructor an opportunity to assess the project's suitability. The proposal also serves as a foundation for the design document that will be developed in the next stage. Below is an example of a student's work illustrating the background and motivation.

For example:

Background and motivation: *Project predicated on the principles of offensive security. Rather than an approach of security by obscurity, which is employed by other fields, the Cybersecurity industry encourages developing mechanisms to test and scrutinize the current state of security. Thus,* CRYNET *will be a project with the purpose of exemplifying the dangers of a comprehensive IRC botnet with advanced malicious bots.*

Project Aims and Objectives: CRYNET *seeks to compile the malware threats of DDOS attacks[2,3], ransomware, resource hogging, and more in order to create a formidable offensive security package. By allowing a user to customize the capabilities of a bot[4,5,6], one can spread bots for a wide range of security testing purposes. Whether that be for enumeration/reconnaissance, system manipulation, or denial of service attempts,* CRYNET *will be an effective means to control and manage said bots. This arbitrary level of bot complexity will allow for one to attempt a clandestine botnet verse a botnet of aggressive ransomware. Although there are some public botnet frameworks available, there does not appear to be an open-source botnet that allows for all of these faucets of malware to be deployable in one program.*

Feedback from the Instructor: *This is a strong start. As a next step, please review the references provided on IRC botnets and begin developing your design. You should also examine relevant platforms, potential experiments or benchmarks, refine the proposed design, and clearly define the expected deliverables.*

Design Phase

Project Design In this design phase, each team defines its project deliverables. The Project Plan document provides a detailed description of the activities required to produce these deliverables. It also serves as a foundation for the design document that will be developed in the subsequent phase. Essentially, the project team divided the major deliverables into smaller, more manageable components, identifies the activities required to complete them, and estimates the necessary resources and time schedules. From Figure 1, we can see sample work from students related to the workflow of the CRYNET.

Below is a general list of elements the instructor will be evaluating in the proposal:

- Backgroun and Motivation
- Project Outline
- Description of the Deliverables
- Evaluation Criteria (see Table 1 for example)
- Project Plan and Timing (see Figure 2 for more details)

The design document fulfills multiple pedagogical functions. First, it requires students to initiate the formal documentation of their project. Second, it provides a space to articulate the expected system behavior through a detailed architectural description. Depending on the project, this may include software, hardware, network, and/or testing architectures. Finally, the design document establishes the overall project scope, including milestones, success and failure criteria, and potential stretch goals that can be pursued if progress permits or deferred for future work.

Task Name	Duration	Start	Finish	Task description
Project planning and design	14 days	5/2/2024	5/19/2024	Brainstorming and design portion of senior project: concept creation and discovering of protocols/technologies to use
Sprint 1 and phase-I presentation	21 days	5/20/2024	6/9/2024	Creating functional IRC Server/Client infrastructure: ensuring that IRC server runs and bots are to connect and receive commands
Sprint 2 and phase-II presentation	21 days	6/10/2024	6/30/2024	Coding malicious aspects of bots: packet crafting/pinging, file encryption, network traversal, etc.
Sprint 3 and poster/final presentation work	21 days	7/1/2024	7/21/2024	Adding capabilities of master program to spawn bots with certain configuration and the ability to send command through IRC server
				In addition, the bulk of the poster/presentaiton work will be completed during this sprint
Final presentation and wrap up	21 days	7/22/2024	8/11/2024	Project is finalized, tested, and presented/demo'd. Any additional "wrapping up" is completed and the project is made publicly available on Github
Estimated finishing date: 7/25/2024				

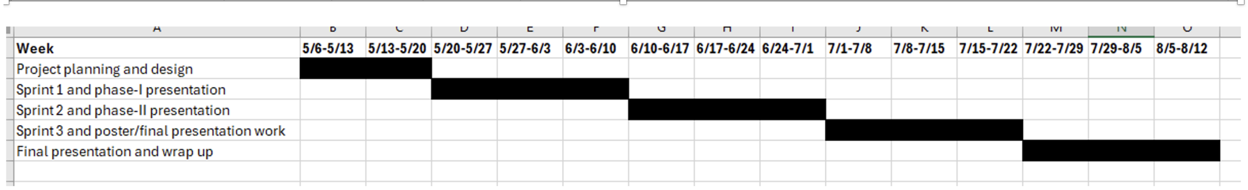


Figure 2: Project Plan and Timing CRYNET

Feedback from the Instructor: *Good work. Below are some key libraries that may be required for the implementation. GNU Compiler Collection[7]: Used to compile the IRC bot Crypto++[8]: Provides cryptographic functions such as encoding, encryption, and hashing; Libircclient[9]:*

Facilitates IRC communication by providing a simple interface for connecting to and interacting with IRC servers; Libtins[10]: Used for packet crafting and sending.

Implementation Phase

In this step, the focus is on *feedback-oriented* learning. The project is designed for students and enables them to consolidate their prior knowledge while working on a term project. It includes Experiment Setup for IRC server, IRC Bots (3 $\times$), Master Program and Victim Machine. Below is a **student sample of a weekly progress report**.

Project: CryNet (Advanced Botnet)
Team Members: xxxx
Progress 5/11/2024 - 5/17/2024:
I spent this week deliberating on how CryNet *should be structured, the technologies that will make the project most efficient, and researching certain concepts involved in the project. In addition, initial project setup and documentation was started.*

Details:

- Researched and read documentation on Internet Relay Chat servers and clients (https://datatracker.ietf.org/doc/html/rfc1459)
- PERSON worked on the atmospheric system API (documentation here)
- Examined example IRC servers in different languages to see what language offers the best support for this form of network programming
- Experimented with IRC protocol by creating a sample IRC server and client in Python in order to familiarize with the concept
- Created initial documentation for the project and created visuals to show the data flow between components Created the Git repository for versioning and source code storage

Major Decisions or Milestones:

- Agreed on a 3-component linear structure: IRC Client $\rightarrow$ IRC Server $\rightarrow$ Master Program
- Decided the programming languages to use: C++ (IRC Client), Java (IRC Server), and Python (Master Program)

Goals for next week:

- Start IRC Server development
- Write or use basic IRC client to test IRC Server functionality
- Start writing documentation
- Refresh on concepts such as network programming

Student Sample Work

Project Outline: CryNet will consist of three major components: an IRC server, IRC client (bot), and a master program. The IRC server will be either be written in C++ or Python using vetted libraries to maintain establish an IRC server. The IRC client will be written in C++ and using the library "libircclient" to connect to the IRC server. Moreover, the IRC client will contain a variable amounts of malicious behaviors, dictated by the master program, that will all be custom written (using certain C++ libraries to perform standard tasks). The "master" program will be written in Python and will be responsible for spawning bots/embedding them into a host executable. This program will also connect to the IRC server and enter commands to an IRC channel in which all active bots are listening on. Commands will be sent from this program to the IRC server, and then all bots will parse the command, determine if the message was for them, and then proceed with the request action. Consequently, it is essential that all three of theses components are inter-connected and network accessible.

Table 1: Five attack scenarios on the IRC Bot actions

	Attack Scenarios
Scenario 1:	DDoS attack on a victim device Using UDP packets & TCP-XMAS packets
Scenario 2:	Remote "fork bomb" execution
Scenario 3:	Remote file system encryption using AES-256-CBC
Scenario 4:	Data exfiltration of critical local password hash files
Scenario 5:	Reporting of vital victim machine host information

Description of Deliverables: The project deliverables will consist of a GitHub repository with the source code which will generate an executable for the IRC server and a Python program to generate bots and communicate with the IRC server. The git repository[11] will supply an installation script to get the environment up and running. The major contributions of CRYNET can be summarized as the following:

- To our best knowledge, we designed and implemented *a novel* and *multifaceted Botnet.*
- We propose the CRYNET system and *Open-source Botnet*.
- *Multiple attack scenarios* have been thoroughly conducted and evaluated in the experiments.

Experimental Setup The IRC Bots were running on local virtual machines (VMs) imitating remote infected devices. It is important to note that the victim machine only was used for simulated DDoS attacks.

- IRC Server on Ubuntu 24.04 (64-bit)
  - Hardware: AWS EC2 instance, Intel Xeon CPU E5-2676 @ 2.40GHz, 1 GB RAM
- IRC Bots (3x) on Ubuntu 22.04.4 (64-bit)
  - Hardware: Intel i7-9700k @ 3.60GHz, VMs with 4 GB RAM each
- Master Program on Windows 10
  - Hardware: Intel i7-9700k @ 4.8GHz, 16 GB RAM
- Victim Machine on Windows 10
  - Hardware: Virtual Machine Intel i7-9700k @ 3.6GHz, 4 GB RAM

Evaluation Criteria: This project's most integral criteria for success are reliability and performance. Although an attractive and intuitive user interface is always desirable, a botnet is inept if consistent communications between bot, server, and user are unable to be made.

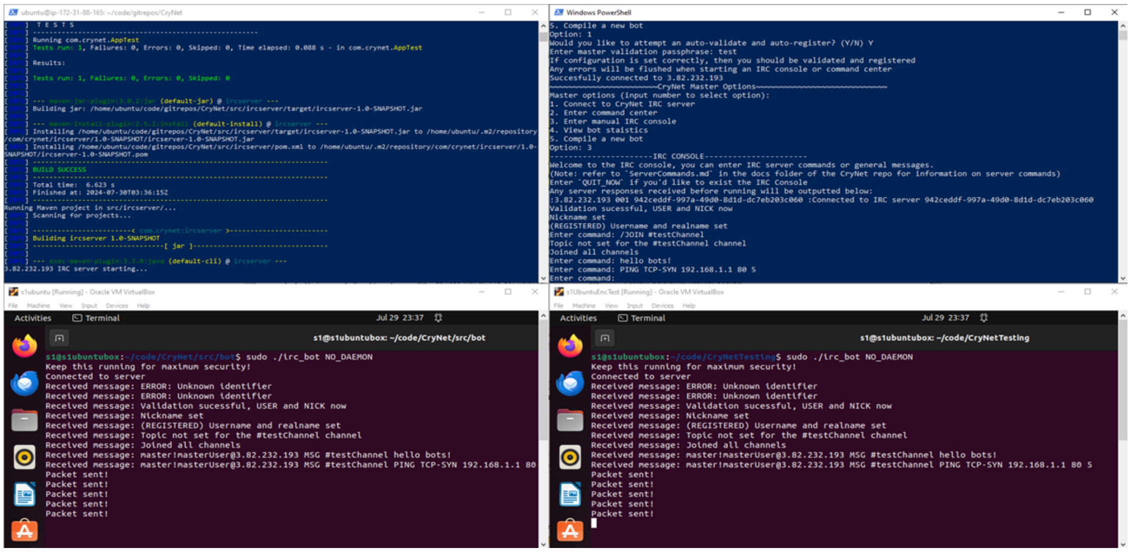
Therefore, the main priority will be to establish and maintain connectivity between the three major components. Commands sent from the user must consistently be received by bots on external nodes and said bots must parse and execute commands accordingly. In addition, bots must be able to consistently perform given command from the user successfully (e.g., ping a request URI). In order to test the effectiveness of the botnet/malware.

Resource Plan: This project will be coded in C++ and python in Visual Studio Code, maintained and saved in git, and use IRC and other networking protocols to maintain end-to-end network connection. All technologies used will be free and open-source. As a result, some sample experimental results are shown in the Figure 3. Open source would be found in the github[11].

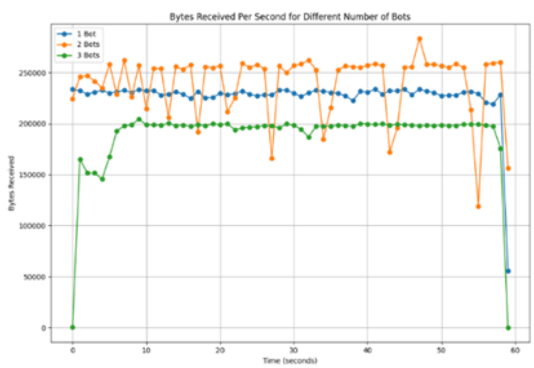
Feedback from the Instructor: *This research was conducted exclusively for scientific and educational purposes. All experiments were carried out in controlled environments, and no real-world systems or users were affected. The methods, models, and results presented are intended to support the advancement of cybersecurity research and are not designed for or applicable to malicious activities. Responsible disclosure and ethical standards were followed throughout the study.*

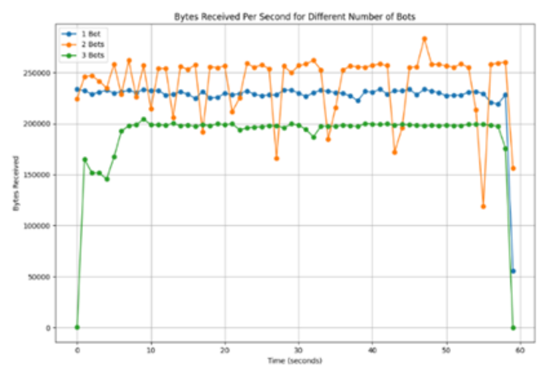
Achievements and Impact

Throughout the program, undergraduate research participation has been greatly strengthened. Although none of the students reported feeling highly comfortable with research processes, this indicates that many were still in the early stages of developing their research competencies. Their responses suggest that they viewed themselves as novice researchers, still navigating key challenges such as understanding methodologies, formulating research questions, and interpreting results. This identified a clear opportunity to provide structured mentorship and targeted resources to build students' confidence and technical competence, ultimately enabling a more enriching and productive research experience. Through these experiences and associated training, students developed a range of transferable skills. These included problem-solving abilities fostered through critical thinking, divergent and convergent reasoning, and iterative trial-and-error approaches. Students also gained the capacity to contextualize problems within broader disciplinary and real-world frameworks, enhancing their ability to connect high-level concepts with practical applications. Additional gains were observed in time management, as students learned to balance project milestones with academic responsibilities, and in collaborative skills, facilitated by interdisciplinary teamwork. Furthermore, communication and presentation skills improved as students prepared for showcases and conference presentations. In the final stage of the program, students' work was disseminated through posters, peer-reviewed publications, and presentations at international conferences. Within this supportive, research-driven academic environment, several barriers to participation were addressed. As a result, more students were encouraged and empowered to pursue meaningful undergraduate research opportunities. The program has demonstrated measurable improvements in student engagement, faculty collaboration, and departmental research productivity, establishing a strong foundation for



(a) Example Usage of CRYNET





(b) Received Packets Verse Seconds

(c) Bytes Received Verse Seconds

Figure 3: **Student Sample Work:** An example of all the components running in unison is displayed on Figure 3a. Four terminals are active, one as an SSH connection to the IRC server (top left), another running an instance of the master program (top right), and two terminals running non-daemonized bots (bottom). The IRC server does not provide user output (apart from starting and shutting down), but the CRYNET workflow steps for the master and bots are shown (Note: The Bots were intentionally run as non-background processes in order to highlight the process).The user runs the master program, connects to the IRC server, launches the IRC console, and begins sending messages to a channel. Likewise, the two bots connect to the server, join a preset channel, and listen for commands. The packets and bytes received during the DDoS attack scenario are shown in Figure 3b and 3c

.

long-term educational and professional success. In the future, it will be necessary to collect more comprehensive data to validate our method and explore interdisciplinary avenues for student collaboration

Conclusion

The senior project has been significantly enhanced by the **EUR** framework, which supports students in reconstructing their prior knowledge and applying it to real-world problems. By fostering self-directed learning, the framework not only motivates students but also enables their research to achieve greater impact. In addition, positive feedback has been received from students, indicating that they gained substantial skills and knowledge through this training. Future work will continue to apply these strategies to strengthen undergraduate research activities across multiple collaborative domains (e.g., AI in Cybersecurity), and multi-dimensional data will be collected to further verify the effectiveness of the **EUR** framework and its influence on students' research engagement.

References

[1] Dina Zohrabi Alaee and Benjamin M Zwickl. A case study approach to understanding a remote undergraduate research program. In *Physics Education Research Conference 2021*, 2021.

[2] Tong Anh Tuan, Hoang Viet Long, Le Hoang Son, Raghvendra Kumar, Ishaani Priyadarshini, and Nguyen Thi Kim Son. Performance evaluation of botnet ddos attack detection using machine learning. *Evol. Intell.*, 13 (2):283–294, 2020.

[3] Ping Wang, Sherri Sparks, and Cliff Changchun Zou. An advanced hybrid peer-to-peer botnet. *IEEE Trans. Dependable Secur. Comput.*, 7(2):113–127, 2010. doi: 10.1109/TDSC.2008.35.

[4] Malak Saleh Aljabri, Rachid Zagrouba, Afrah Shaahid, Fatima Alnasser, Asalah Saleh, and Dorieh M. Alomari. Machine learning-based social media bot detection: a comprehensive literature review. *Soc. Netw. Anal. Min.*, 13(1):20, 2023.

[5] Ying Xing, Hui Shu, Hao Zhao, Dannong Li, and Li Guo. Survey on botnet detection techniques: Classification, methods, and evaluation. *Mathematical Problems in Engineering*, 2021(1):6640499, 2021.

[6] Ziming Zhao, Zhaoxuan Li, Fan Zhang, Tingting Li, and Jianwei Yin. Poster: Combine topology and traffic to calibrate P2P botnet identification in large-scale network. In *Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos, Sydney, NSW, Australia, August 4-8, 2024*, pages 16–18. ACM, 2024.

[7] Free Software Foundation. Gnu compiler collection (gcc). `https://gcc.gnu.org/`, 2022.

[8] Weidai11. Cryptopp. GitHub, 2022. `https://github.com/weidai11/cryptopp`.

[9] Shaoner. Libircclient. GitHub, 2022. `https://github.com/shaoner/libircclient`.

[10] Mfontanini. Libtins. GitHub, 2022. `https://github.com/mfontanini/libtins`.

[11] Crynet. GitHub, 2024. https://github.com/s1-dev/CryNet.