Issues in Information Systems
Volume 27, Issue 1, pp. 605-618, 2026

DOI: https://doi.org/10.48009/1_iis_2026_141

Open-Source Tools as a Pedagogical Equalizer in Underfunded IT Programs

Gary Janchenko, *Robert Morris University, janchenko@rmu.edu*
Karen Paullet, *Robert Morris University, paullet@rmu.edu*
Evan Kinney, *Robert Morris University, kinney@rmu.edu*
Tyler Stewart, *Robert Morris University, stewartty@rmu.edu*

Abstract

University budgets are tightening nationwide, limiting their ability to purchase the proprietary software traditionally used for information technology (IT) instruction. Financial challenges jeopardize students' opportunities for meaningful hands-on learning experiences and industry-relevant skills development. The paper reviews the impact of funding cuts on IT programs and assesses open-source technologies as viable and cost-effective alternatives. Researchers identify and evaluate open-source tools that can augment or replace commercial software, while maintaining instructional quality and supporting core learning outcomes. The study provides recommendations for institutions seeking sustainable approaches to technology integration in the face of ongoing budget constraints.

Keywords: information technology, cybersecurity, open-source tools, pedagogy, data science, networking

Introduction

Colleges and universities across the United States are under financial constraints that have a direct impact on their ability to support academic programs focusing on technology. With increasing demand for sophisticated digital infrastructure, institutions are dealing with flat or shrinking budgets, creating what EDUCAUSE calls "the greatest systematic challenge facing higher education IT" (2024) in the years to come. Such budgetary constraints limit the ability of institutions to invest in the proprietary software and platforms which have traditionally been used to deliver hands-on instruction in information technology (IT) programs. Universities are challenged to sustain the technological environments students need to develop industry aligned competencies, without adequate funding.

Unfortunately, the budgetary constraints coincide with the growing dependence of higher education on digital tools and demands from industry employers to produce learners with skillsets relevant to new demands. Open-source technologies are deeply integrated throughout higher education, according to research from the Apereo Foundation, with 89% of .edu domains using at least one open-source technology and more than 26,000 installations across institutions (Apereo, 2025). Though open-source tools being widely used, few campuses have a formal strategy for leveraging open-source solutions to meet instructional needs, resulting in IT programs becoming reliant on costly proprietary systems that may no

longer be financially sustainable. As budgets tighten, the gap between what it takes to teach effectively and what is available for use in the classroom becomes more and more difficult to bridge.

For institutions in need of reducing costs, open-source software offers an attractive alternative for providing quality IT instruction. Community-source and open-source models have demonstrated great promise in delivering sustainable, mission-critical software that meets the needs of higher education, particularly when institutions collaborate to share development and support responsibilities (EDUCAUSE, 2024). Similarly, recent analysis has pointed to how open-source tools, from learning management systems to data analytics platforms, can dramatically reduce IT costs while maintaining or even improving instructional quality (Ugurluoglu, 2024). The tools eliminate licensing fees and offer the flexibility, transparency and customizability often absent in proprietary systems.

Drawing on these trends, this paper discusses the potential for open-source technologies to serve as viable replacements for proprietary software in University IT programs when faced with lack of funding. The study seeks to offer a framework for institutions interested in sustaining hands-on, experiential learning in resource-constrained environments by identifying specific open-source tools that support core IT competencies such as cyber forensics, networking, cybersecurity, programming, and data management. In so doing, the research contributes to an emerging conversation about how higher education can regain technological independence and instructional resilience through purposeful adoption of open-source solutions. This study seeks to answer the below research question.

RQ1: What specific open-source tools can higher education institutions implement to maintain required technical competencies in IT courses despite declining technology budgets?

Literature review

As of September 2025, state appropriations for higher education totaled $129.0 billion, representing a 4% increase from the previous year (National Education Association, 2025). While this is an increase in most states, 17 states experienced a decrease in funding due to inflation. This decrease could also be attributed to the 17 states perceiving higher education as having alternative funding through tuition (National Education Association, 2025). Because of this, higher education institutions can be more susceptible to funding cuts. In addition to this challenge, states often cap tuition increases, which adds further financial strain to higher education (National Education Association, 2025).

Miller (2009) shares how pressure to reduce budgets associated with funding cuts has bled into the Information Technology (IT) departments. Because of this, it is possible for industry-level technologies, to be absent from IT courses. To help offset some of the budget pressure, Miller (2019) recommends a few solutions, such as cutting low-hanging fruit, renegotiating contracts, and re-evaluating usage patterns. In this context, low-hanging fruit could be things such as newsletters, printers across campus, or other physical items (Miller, 2019). Additionally, the contracts that higher education institutions have with Internet Service Providers (ISPs), software vendors, or even food service providers could be renegotiated to gain better rates. Part of this could also come from evaluating usage patterns. For example, a software application previously required in an earlier course may no longer be in use and could therefore be eliminated to reduce expenses. Similarly, if enrollment declines, the institution may be able to renegotiate the associated software license for a smaller-scale plan that better aligns with class needs.

In response to the cost associated with tools needed for the classroom in IT courses, open-source options could act as a solution. Safari and Zolkipli (2025) explain how, in cybersecurity training, tools such as Kali Linux, Wireshark, and Metasploit have provided cost-effective ways for hands-on learning. Since these tools are primarily open-source, they are free or have a low cost/donation requirement to access the software (Safari & Zolkipli, 2025). Because of this, these tools can be valuable alternatives to some of the more costly tools. For example, FTK could cost around $10,000/yr for a license, depending on what level of training and support is desired with the software (Exterro, n.d.). In contrast, Autopsy is an open-source tool that provides similar functionality with no cost for licensing (Autopsy, n.d.).

When considering industry standards, open-source tools tend to be supported by professionals with expertise in their fields of choice (Safari & Zolkipli, 2025). In fact, tools such as Nmap and OpenVAS provide active communities that frequently update the tools with plugins, architecture patches, and emerging trends (Safari & Zolkipli, 2025). This community-driven support system enables engagement between students and industry professionals. Thus, exposure to best practices and how to fully utilize the skills associated with these tools can be developed (Safari & Zolkipli, 2025). The ability for students to not only access the support network provided but also to engage with production-like infrastructure that some of these open-source tools are hosted on helps them prepare for careers in technology.

In IT courses, lectures alone can be insufficient to prepare students for their post-graduation careers. The solution to this is hands-on learning, which is critical for IT as it is a field that requires learning by doing (Huffman, 2026). Malik and Zhu (2022) explain how there is a growing need for computer science and information technology education, with theory and practice being taught. This refers to the idea of project-based learning (PBL), where teaching approaches combine theory with hands-on activities. Malik and Zhu (2022) use a networking course as an example to illustrate how the theory of developing a network helps establish a baseline; however, without working with servers and switches, the students are not adequately prepared for that type of role. The same concept could be applied to cybersecurity. A student could learn the theory of network monitoring, but without using a tool such as Splunk or Wireshark, they would not know how to literally monitor the traffic. Humans naturally learn through trial and error, which is not possible without hands-on learning; thus, emphasizing the need to incorporate this method of learning in the IT classroom (Huffman, 2026).

Institutional Shifts Toward Open-Source Solutions

Universities facing budget limitations have increasingly turned to open-source technologies as a practical solution for sustaining high-quality instruction in information technology programs. Saint Louis University (SLU) has created a university-based Open-Source Program Office (OSPO), known as Open-Source with SLU (OSS). The initiative, which offers a sustainable model for the integration of open-source tools into coursework, was developed in response to the high costs and limitations of traditional software instruction (Holdener, E., et.al, 2024). OSS allows students to work directly with open-source software maintained at the university, which allows the institution to reduce reliance on expensive proprietary tools while still providing authentic, real-world development experience. The approach was developed specifically to reduce long-term costs and create a financially sustainable instructional model.

Carnegie Mellon University (CMU) has also incorporated open-source tools into its IT and computer science curriculum, providing more opportunities for hands-on learning without the high licensing costs of commercial software. CMU's Open-Source Programs Office redesigned a software engineering course to

serve as a remote internship and give students the opportunity to work on real open-source projects with industry mentors (Carnegie Mellon University, 2026). The model provides students access to professional tools and distributed development environments, many of which are open-source, at minimal or no cost to the institution. The program benefits from partnerships with organizations like Red Hat and the Eclipse Foundation and shows how open-source ecosystems can provide good IT education even in tight budget environments.

CMU's redesigned Computing @ Carnegie Mellon (C@CM) course, which shifted from an expensive, lab-based model to an online format based on OLI's open-source infrastructure (Carnegie Mellon University, 2026). This example is different, even though it is still at CMU, because it shows how open-source tools cut internal university costs. The redesign means less computer-lab time and less TA staffing are needed, while keeping a 98% pass rate. This shows how open-source platforms can help reduce the operational costs in the IT-skills courses even in well-funded institutions.

The Open Education Resource universities (OERu), a consortium of 36 publicly funded institutions, shows how open-source technologies can make a big difference across universities in terms of scalability and cost-effectiveness in higher education. OERu runs a fully open, distributed digital learning ecosystem that is built entirely on free and open-source software (FOSS) and supports hundreds of thousands of learners on an annual IT infrastructure budget of less than $10,000. The model demonstrates the ability of institutions to provide IT-related training, digital laboratories and online learning environments at a fraction of the cost while preserving flexibility, independence and pedagogic quality (Lane, et.al., 2021). The OERu example demonstrates that open-source tools are not just viable options to proprietary systems but can also increase institutional resilience and expand access to technology-enabled learning experiences.

Methodology

The methodology of this study is a structured review of open-source IT tools and their industry standard counterparts to determine how well they can be used for hands on learning experiences in information security, networking and data science courses. Researchers initially identified commercially available platforms used across a broad spectrum of areas such as intrusion detection, IT security, network simulation, and data science. Then they selected open-source tools with similar capabilities. This process included the review of technical capabilities and the degree to which each tool supports realistic, scenario-based learning. The idea was to see if open-source platforms can provide the same instructional value as proprietary systems, without the licensing cost to academic programs.

Based on this comparison, the researchers conducted a comprehensive review of the implementation of these tools in information technology courses in higher education. Special emphasis was placed on how these tools can assist students in building skills, practice repeatedly and meet industry expectations for hands on proficiency. The methodology combines these results to provide a strong evidence base for the integration of open-source tools into IT security, networking and data science curricula, thereby maintaining the rigor of instruction and reducing financial barriers.

Open-Source Alternatives for Data Science Courses

Higher education institutions face mounting pressure to deliver rigorous technical instruction while operating under tightening fiscal constraints. The proliferation of robust, freely available open-source

software has emerged as a viable strategy for sustaining technical competencies in information technology, data science, and analytics curricula without recurring license fees. The literature reviewed below identifies specific open-source tools that institutions can implement, the pedagogical frameworks within which they have been deployed, and the documented outcomes reported across diverse academic settings.

Python has emerged as the dominant open-source language for data analytics instruction, owing to its readability, its extensive scientific library ecosystem, and its absence of licensing costs. Kabir, Ahmed, Islam, and Ahmed (2024) conducted a systematic literature review of Python's role in data analytics and concluded that NumPy, Pandas, Matplotlib, Seaborn, and Scikit-learn collectively provide functionality spanning the entire analytical workflow, from data ingestion and cleaning through visualization and predictive modeling. Their review documents that these libraries are deployed across retail, finance, healthcare, and manufacturing industries, suggesting that students trained in this stack acquire skills with direct labor-market relevance. The authors note that Python's principal limitation, scalability for very large datasets, is mitigated through integration with free distributed-computing frameworks such as Apache Spark via PySpark and Dask (Kabir et al., 2024).

Institutional case studies confirm the viability of Python as a curricular foundation. Paffenroth and Kong (2015) describe how Worcester Polytechnic Institute structured its graduate Data Science program around Python, IPython notebooks, scikit-learn, NumPy, SciPy, and pandas, deploying these tools across introductory through capstone-level coursework. The authors emphasize that Python's open-source license and extensive library catalog allowed instructors to carefully calibrate the difficulty and scope of assignments while exposing students to the same tools used in cutting-edge research, including convex optimization, deep learning, and scientific visualization (Paffenroth & Kong, 2015). Similarly, Duda et al. (2021) document a collaboratively developed Python data science curriculum at the University of Michigan that uses Jupyter notebooks to teach NumPy, Pandas, Matplotlib, and Seaborn to learners with no prior programming experience. Their curriculum is released as an open educational resource on GitHub, further extending its cost-saving potential to other institutions seeking to adopt rather than develop a data science introduction.

R offers a complementary open-source environment particularly well suited to statistics-heavy curricula. Çetinkaya-Rundel and Ellison (2021) describe an introductory data science course at Duke University built entirely around R, the tidyverse package suite, R Markdown for reproducible reporting, and Git and GitHub for version control and collaboration. The course was designed for students with no statistical or programming prerequisites and demonstrated that an open-source toolchain could carry students from zero experience to completion of a fully reproducible data science project on a dataset of their choice within a single semester (Çetinkaya-Rundel & Ellison, 2021). Importantly, the authors used RStudio Cloud, a managed, web-based environment, to eliminate the installation friction that historically consumed instructional time and required local technical support, an explicit acknowledgment that infrastructure decisions affect both pedagogical outcomes and institutional resource allocation.

For programs unable to commit substantial time to programming instruction, several open-source platforms provide graphical interfaces for analytics. Pynam, Spanadna, and Srikanth (2018) compared five such tools, RapidMiner, Weka, KNIME, R, and Orange, and concluded that each offers distinct pedagogical advantages. Weka and KNIME were judged accessible to learners without coding knowledge, RapidMiner suitable for more technically oriented students seeking exposure to advanced algorithms, and Orange, notable for its Python-based architecture and active user community. The authors emphasize that all five tools support standard analytical workflows including data preprocessing, classification, clustering, regression, and visualization, and that the choice among them can be aligned with course objectives and student backgrounds rather than dictated by budget constraints (Pynam et al., 2018). This finding is particularly relevant for institutions seeking to introduce analytics within business, healthcare, or social science programs where programming may be a secondary objective.

A recurring theme across the literature is that the value of open-source software extends beyond the absence of license fees to include the elimination of local installation, configuration, and support costs. Çetinkaya-Rundel and Ellison (2021) explicitly framed their decision to use RStudio Cloud as a means of avoiding the loss of entire class periods to troubleshooting individual students' laptops. Tufino, Oss, and Alemani (2023) reached a similar conclusion in their integration of Python data analysis into a first-year physics laboratory course at the University of Potsdam, where Jupyter Notebooks were deployed both in the Anaconda environment and considered for cloud platforms such as Google Colab. The authors found that students with diverse prior programming experience successfully developed data-analysis competencies within four weeks of instruction, and that 86 percent of students reported plans to deepen their Python skills further (Tufino et al., 2023). The use of GitHub as a free version-control and learning-management platform in the Duke course (Çetinkaya-Rundel & Ellison, 2021) further illustrates how freely available industry-standard infrastructure can replace proprietary learning management systems for specific instructional purposes.

The cumulative evidence from these case studies suggests that open-source toolchains do not require institutions to compromise on the rigor or relevance of technical instruction. Duda et al. (2021) report measurable gains in student-reported confidence in Python programming, problem solving, and collaboration, with 62 percent of program alumni still actively coding after participation. Tufino et al. (2023) document that students transferred newly acquired Python competencies into subsequent laboratory experiences, suggesting durable skill acquisition rather than short-term tool familiarity. Paffenroth and Kong (2015) observed that access to a free language with an extensive library ecosystem allowed students to engage with research-level problems early in their academic careers, an outcome difficult to achieve when proprietary licensing costs constrain which tools can be deployed across an entire department or program.

Open-Source IT Security and Networking Course Alternatives

Information security and networking programs are increasingly adopting open-source security tools and open-source networking platforms to provide rigorous hands-on instruction while avoiding the high licensing costs of commercial products. Open-source can be leveraged to offer genuine technical training, improve access to sophisticated security and networking situations, and ensure quality of curriculum even when facing budgetary limitations through the use of open-source in lab exercises and simulation-based learning. The tools below provide cost-free open-source alternatives to expensive industry-standard software for use in IT, security, and networking courses.

Splunk Enterprise Security is one of the most widely used commercial Security Information and Event Management (SIEM) tools across multiple industries. It is able to ingest, index, and correlate machine, user, and security data from any platform and provides a single pane of glass for Security Operations Center (SOC) Analyst (Splunk, Inc., n.d.-a). The platform’s core efficiency is based on a schema-on-read architecture and the Common Information Model (CIM) that standardizes different sources of data, such as firewall logs and endpoint telemetry, into a standard format for multiple-platform analysis. This normalization enables Risk-Based Alerting (RBA) that combines a number of low fidelity events into a risk score to prioritize high threats and reduce analyst alert fatigue. Splunk Enterprise Security also provides native integrations with the MITRE ATT&CK framework, enabling SOC teams to map their detection coverage to specific adversary tactics and identify visibility gaps in real time. The system acts as a central information hub for threat intelligence and automatically compares ingested machine-generated data against known malicious indicators, helping to accelerate incident response. The platform is licensed either by daily

data ingest volume (gigabytes per day) or by Splunk Virtual Compute units, both of which put a production deployment well over five and six-figure annual costs (Splunk, Inc., n.d.-b), out of reach for most college programs.

Wazuh SIEM is a free and open-source platform that offers the same basic SIEM functionalities that a student would need to learn when using Splunk. It uses agent- and syslog-based log collection, centralized indexing and search, rule- and decoder-based correlation, dashboards and visualizations, and out-of-the-box mapping of detections to the MITRE ATT&CK framework (Wazuh, Inc., n.d.-a; Wazuh, Inc., n.d.-b). Wazuh Inc. developed it with a growing open-source community. It is licensed for unlimited academic and commercial use. Wazuh has a separated architecture with lightweight agents, a central manager and an indexing engine that makes it possible to collect logs at scale and monitor accuracy in real time across a number of operating systems. OpenSearch stack integration offers students the ability to have data visualization and complex long-term trend analysis without the responsibility of commercial licensing costs. Additionally, native support for Vulnerability Detection and File Integrity Monitoring (FIM) expands the platform's usefulness beyond traditional SIEM boundaries, offering an integrated analysis of the host-level security posture. These features provide learners with hands-on experience with enterprise-grade security workflows, from initial ingestion to finalized incident.

The functional gap between Splunk and Wazuh decreases considerably when the goal is instructional purposes rather than petabyte-scale telemetry. Splunk also has a more feature-rich search language (SPL), the third-party Splunkbase app ecosystem, premium content packs, and vendor support contracts with formal service level agreements (SLAs). Wazuh has a rules engine of its own, a dashboard based on Kibana, and a search interface that is more limited but almost identical, and is community supported. A recent peer-reviewed evaluation in Computers & Security demonstrated that Wazuh covers a meaningful share of MITRE ATT&CK techniques out of the box and can be extended with custom rules to reach parity in many detection categories, and recent academic work has demonstrated that Wazuh detections can be augmented with machine-learning models for higher-fidelity threat detection.

In the classroom, Wazuh can be used to build a SOC simulation lab that provides students a sandbox environment replicating typical enterprise security operations (Javid, 2024; Sutton et al., 2026). With Wazuh agents deployed on multiple virtual assets, instructors can replicate real-world attack vectors such as brute-force attempts or lateral movement and give their students an understanding of how raw telemetry is converted into actionable alerts. The hands-on nature of this exercise requires students to interact directly with the underlying algorithms and rulesets and therefore allowing them to gain a deeper understanding of the logic required to tune a SIEM for reliable detections. In addition, the inclusion of Vulnerability Detection modules allows learners to engage in proactive risk management and security patch prioritization, as well as reactive incident response workflows. Students can all navigate leveraging frameworks like NIST and MITRE to perform all aspects of the SIEM lifecycle, including log ingestion, parsing, detection, triage, and response, at an enterprise-level all at zero licensing cost.

In 2015, Gartner coined the term "Security Orchestration, Automation and Response" (SOAR). It describes the type of platform that ingests alerts (usually from a SIEM) and is able to execute automated playbooks for triage, enrichment, containment, and case management (Gartner, n.d.). These platforms are the central hub marketplace of more than 700 downloadable content packs, and dozens of prebuilt analyst playbooks (Palo Alto Networks, n.d.). Splunk SOAR, like Cortex XSOAR, takes a "Python-first" approach to provide the level of customization needed for complex enterprise workflows, with an abstraction layer of "Apps" and "Assets" to enable large-scale API connectivity (Splunk, Inc., n.d.-c). Tines takes a different

commercial approach: a no-code visual workflow builder organized around a small set of simple "actions" wired together on a storyboard, heavily used by mid-market and enterprise SOCs to automate repetitive analyst work (Tines, n.d.). These systems allow for facilitating containment actions such as host isolation via EDR or removal of an identity token while maintaining a forensic audit trail via unified case management.

Tracecat is an open-source, AI-native security orchestration, automation, and response (SOAR) platform offering a transparent, code-first alternative to enterprise platforms like Tines and Splunk SOAR. It brings a combination of event-driven workflows, AI-driven agents, case management, and lookup tables into a single unified platform with a focus on architectural visibility and engineering control. It uses Temporal for workflow orchestration to ensure that complex, long-running security processes maintain their state and handles retries during network issues or service outages. Additionally, it uses nsjail for secure, isolated sandboxing for running custom Python scripts and automation components. This enables engineers to pull Python automations directly from Git repositories and to describe playbook logic with YAML-as-code. It then enables security orchestration to be integrated seamlessly into standard DevSecOps CI/CD pipelines and version control workflows. This code-first platform, enhanced with a native AI assistant and specialized LLM agents, can automate alert triage, summarization, and enrichment to efficiently transform telemetry into actionable security cases. Tracecat's flexible deployment model on Docker, Kubernetes, or AWS Fargate allows a highly scalable but resource-efficient environment. The platform provides a functional and vendor-neutral framework for building and maintaining enterprise-grade security automation (TracecatHQ, n.d.; Tracecat, n.d.).

The differences between the commercial platforms and Tracecat are real, but again, not as extensive as they may seem from the outside of the SOC use case. Tines and XSOAR provide vendor-curated integration marketplaces, 24/7 enterprise support, AI-assisted playbook generation, and polished case-management interfaces. Tracecat is open-source, self-hosted, and has no vendor SLAs and fewer certified content packs. This trade is conducive to a classroom setting, not prohibitive: Tracecat can be torn down and redeployed per student or per lab section. Forcing students to author their own integrations against vendor REST APIs makes a great educational opportunity in itself.

In the classroom Tracecat can be used to shift students from theoretical-based detection to operational incident response management. While SIEM is about identifying threats, a SOAR-driven lab requires students to automate the remediation steps by mapping NIST or ISO response phases directly into functional logic (Cichonski et al., 2025). Students can be given a structured lab to consume a Wazuh alert via webhook, enrich indicators against community-tier APIs like VirusTotal or AbuseIPDB, and make containment decisions via conditional logic. To simulate the complex nature of production without the cost of licenses, Tracecat can be used to build playbooks against mock applications like a fictional HR portal or firewall, allowing students to account for real-world constraints such as API rate limits, branching logic, and error handling. This approach allows for these skills to be easily transferred to enterprise platforms such as Tines or XSOAR. Tracecat's basic functions of triggers, state management, and case documentation are almost identical to the paid enterprise platforms. Tracecat can also be run in a containerized environment to provide this enterprise-level experience at no cost.

Proofpoint is the leading commercial enterprise email security platform and has been named a Leader in the Gartner Magic Quadrant for Email Security in 2024 and 2025, ranked highest in Ability to Execute (Proofpoint, Inc., 2025). It is primarily used as a secure email gateway where an organization points their MX records to Proofpoint, which scans all inbound messages in-line for phishing, business email

compromise, and malware prior to delivery. Proofpoint's Targeted Attack Protection provides URL and attachment sandboxing, account takeover detection, and supply-chain threat intelligence (Proofpoint, Inc., n.d.; Proofpoint, Inc., 2024). The platform is licensed per-mailbox and is one of the more expensive items in an enterprise security budget. Their Nexus AI framework employs large language models for sentiment and intent analysis with a relationship-based model to detect payload-less threats and impersonation of executives to further enhance their protection. Additionally, the system decreases the attack surface by using computer vision to detect credential-harvesting sites and specific engines to block Telephone-Oriented Attack Delivery (TOAD) and compromised vendor domains. Proofpoint's TAP system provides pre-delivery protection through URL rewriting and sandboxing, but it also has an API-based integration with Threat Response Auto-Pull (TRAP) to automatically quarantine malicious messages post-delivery for all enterprise mailboxes.

Sublime Security's approach is different by design, and more transparent. The Sublime Security Platform is free and self-hostable as a Community Edition for up to 100 mailboxes; it integrates via API with Microsoft 365 and Google Workspace and analyzes messages post-delivery, rather than acting as a gateway (Sublime Security, n.d.-a). Detections are written in code using Sublime's Message Query Language (MQL), and the entire collection of detections is published as open-source on GitHub for anyone to read, fork and contribute (Sublime Security, n.d.-b; Sublime Security, n.d.-c). Supporting indicator data, including disposable email domains, macro-capable file extensions, and archive types, are also published in an open repository (Sublime Security, n.d.-d). This architecture allows the platform to execute deep content inspection with natural language understanding (NLU) to identify intent and urgency in phishing attempts, and computer vision to identify fraudulent logos and credential harvesting pages. Sublime's API provides the flexibility of SaaS, single-tenant or fully self-hosted deployment, giving security teams full control over their data and detection logic. The platform further automates the defensive lifecycle through its Autonomous Detection Engineer (ADE) that constructs and maintains coverage for new attacks based on live signals. In addition, its automated triage capabilities allow what is termed "herd immunity" in which the system automatically quarantines new messages across the organization matching the patterns of previously reported or identified threats.

Sublime offers features and transparency that better meet the needs of today's enterprise security analyst than many traditional platforms. While Proofpoint is largely a black box, Sublime exposes its entire detection model. Analysts can see exactly what MQL rule fired on a specific attack, adjust logic for their environment, write tests, and push changes into production. This transparency enables a "detection-as-code" workflow, allowing analysts to master the underlying logic of Natural Language Understanding and behavioral signals. Many large organizations are adopting Sublime as its model aligns with the industry shift toward programmable, version-controlled, and explainable security operations. Proofpoint's enterprise benefits are more operational, such as in-line blocking at the MX-gateway, vendor-managed threat intelligence, and managed response services, representing traditional infrastructure for scale. However, Proofpoint can cost organizations thousands, if not millions, of dollars per year, while Sublime generally cuts costs by more than half and is free and open-source for up to 100 mailboxes. In contrast, Sublime lets analysts work directly with email defense mechanics via open-source collaboration and its GitHub detection repository. This approach provides technical sophistication and flexibility not possible with a purely administrative interface, bridging theoretical defense to industry-standard implementation and preparing analysts to handle the complexity of enterprise threats with a defensible understanding of detection.

In the classroom, Sublime Security offers a set of pre-built labs that can be used for challenging hands-on detection engineering exercises without having to develop content from scratch (Sublime Security, n.d.-e).

Using these modules, students programmatically query the Message Data Model for advanced threats using Message Query Language (MQL), to look at attachment metadata or fuzzy string matching to identify brand impersonation (Bertiger et al., 2025). Students can look for inconsistencies in message authentication headers (SPF, DKIM and DMARC) and evaluate directionality to detect lateral phishing attempts or malicious URLs within message bodies (Rose et al., 2019). In addition, the platform provides the ability to conduct end-to-end simulations where students can analyze a phishing attack and then pivot to a SIEM to correlate email data to web proxy and IDS logs. The process provides students with the ability to not only find malicious content, but also write and improve proactive detection-as-code rules. These learning tools allow students to participate in workflows like those utilized by security teams in Fortune 500 environments and build a professional portfolio of MQL rules that demonstrates the technical competence and logical thinking required for modern security engineering roles.

Results

The analysis found a robust ecosystem of open-source IT tools that can deliver the technical competencies required in information security, networking, and data science courses even as institutional technology budgets continue to shrink. The results from all categories studied show that freely available platforms provide functionality on par with their industry-standard counterparts answering the below research question.

RQ1: What specific open-source tools can higher education institutions implement to maintain required technical competencies in IT courses despite declining technology budgets?

The study shows that the addition of open-source security tools to higher-education IT curricula is very advantageous for student learning. These free technologies provide hands-on functionality that is very similar to the capabilities of industry-standard systems, allowing students to practice threat detection, network configuration, traffic analysis and incident response in realistic environments. By using open-source tools, institutions can keep offering rigorous technical competencies in IT security, networking and data science courses without the financial burden of recurring software licensing fees. The findings confirm that open-source ecosystems not only maintain instructional quality but also enhance access to real-world skill development for students across technology disciplines.

The free tools surveyed in this study directly answer the research question by demonstrating that institutions can sustain every core IT competency without proprietary licensing. Across data science, security, and networking, each costly commercial platform has a documented open-source counterpart that delivers comparable instructional value: Python with its NumPy, Pandas, Matplotlib, Seaborn, and Scikit-learn ecosystem (alongside R and the tidyverse) covers the full analytical workflow from data ingestion through predictive modeling; Wazuh replicates Splunk Enterprise Security's SIEM functions, including log collection, correlation, dashboards, and native MITRE ATT&CK mapping, at zero licensing cost; Tracecat reproduces the trigger, state-management, and case-documentation logic of enterprise SOAR platforms like Tines and Cortex XSOAR in a containerized, re-deployable environment; and Sublime Security's free Community Edition exposes its entire detection-as-code model as an open alternative to Proofpoint's black-box email security. The pattern is consistent: where a proprietary tool may carry five- and six-figure annual costs (FTK at roughly $10,000 per year, Splunk well into six figures), the open-source equivalent eliminates the licensing fee while preserving the hands-on, scenario-based learning that IT instruction requires. Institutional case studies reinforce this, from Duke and Worcester Polytechnic building entire data science

curricula on free toolchains to Wazuh-based SOC simulation labs, showing that these tools are not merely theoretical substitutes but proven, deployable answers to maintaining technical rigor under budget constraints.

The results also reveal that professional communities that maintain and improve open-source IT tools make them more effective. The tools examined have active groups of practitioners who contribute updates, plugins and architectural improvements. This ongoing process of development makes sure that the tools stay current with industry practices and new threats. More importantly, these communities allow students to observe real-world problem solving, to follow discussions among experienced professionals, and to access extensive documentation and peer-generated tutorials. Research indicates that this shared expertise ecosystem fosters a more authentic classroom learning environment, acting as a bridge for students to the professional worlds they will enter after graduation.

Conclusion and Future Research

The review found evidence that, alongside technical capability, these tools are already being successfully integrated into higher-education programs. Universities that have already implemented open-source alternatives to the information technology curriculum showed that open-source platforms can provide the repeated practice, remote lab access and scalable deployment, needed to maintain the instructional quality within financial constraints. Institutions reported better student engagement, greater flexibility in designing their labs and significant cost savings by removing recurring licensing fees. In summary, these results demonstrate that a variety of open-source tools can be applied to maintain core competencies for IT security and networking courses to where students can still get real hands-on training even with budget constraints.

The analysis further validates that open-source platforms significantly improve hands-on, practice-based learning in IT security, networking and data science courses. Students can download these tools onto their own computers, play with them outside the scheduled lab time, and repeat tasks until they get them right. This flexibility supports more in-depth learning than lecture-based instruction alone, especially in areas where the skills of troubleshooting, configuration, and iterative testing are essential. The findings indicate that students who work directly with hands-on tools and other open-source technologies build stronger technical intuition and greater confidence in applying theoretical concepts. This lends support to the conclusion that open-source ecosystems not only lower financial barriers but also enhance the pedagogical quality of IT programs through continuous, hands-on interaction with industry-relevant tools.

Future research should focus on expanding the evaluation of open-source IT tools across a broader range of information technology environments to determine how these platforms perform under varying instructional conditions. Our findings show that open-source tools can be successfully used to support the development of hands-on skills. However, more research is needed to understand the long-term student outcomes such as certification readiness, job placement, and transferability of skills to enterprise-level systems. Further work to enhance the evidence base would include exploring the impact of other instructional models, such as hybrid labs, virtualized environments or cloud-based simulations, on tool adoption and student proficiency.

Future work should also investigate the incorporation of community supported platforms within collaborative learning frameworks that link students with industry practitioners. This includes considering mentorship opportunities, contribution-based learning and role of open-source participation in development of professional identity. Further, studies should examine institutional barriers to adoption, such as faculty

training requirements, infrastructure limitations, and curriculum fit. Future studies that explore these areas will help institutions to more effectively use open-source technologies in providing high-quality IT security and networking education under continuing budget constraints.

References

Apero Communications (2025, December 5). Open Source in Higher Ed: Early findings from Apereo's OSSHE Study – and how you can get involved. https://www.apereo.org/news/2025/open-source-higher-ed-early-findings-apereos-osshe-study-and-how-you-can-get-involved

Autopsy. (n.d.) Download autopsy for free. https://www.autopsy.com/

Bertiger, A., Filar, B., Lurtha, A., Meschiari, S., Mitchell, A., Scholten, S., & Sharath, V. (2025). Evaluating LLM generated detection rules in cybersecurity.arXiv. https://arxiv.org/abs/2509.16749

Carnegie Mellon University (2026). Open source programs office (OSPO). https://www.library.cmu.edu/services/ospo

Carnegie Mellon University (2026). How a re-designed college level course led to improved outcomes and better allocation of resources. Carnegie Mellon University. DataLab. https://www.cmu.edu/datalab/research/success-story-c-at-cm.html

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2025). Incident response recommendations and considerations for cybersecurity risk management: A CSF 2.0 community profile (NIST Special Publication 800-61 Rev. 3.) National Institute of Standards of Technology. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r3.pdf

Exterro. (n.d.) FTK physical license. https://store.exterro.com/products/ftk?srsltid=AfmBOopdcaK0sw8l9-97Pdv0qjPWq0ZnIvXLa6exbS0gOAeo-oKb7tls

Fuchs, I.H. (2024) Challenges and opportunities of open source in higher education. EDUCAUSE. https://www.educause.edu/research-and-publications/books/tower-and-cloud/challenges-and-opportunities-open-source-higher-education

Gartner, (n.d.) Security orchestration, automation and response (SOAR) [Glossary Entry]. Retrieved May 14, 2026 from https://www.gartner.com/en/information-technology/glossary/security-orchestration-automation-response-soar

Holdener, E., & Shown, D. (2024, July 12). Building software engineering capacity through a university open source program office. FSE Companion 2024, Puerto de Galinas, Brazil

Huffman, T. (2026, January 14). The importance of hands-on-learning in higher education. McCowngordan.. https://mccowngordon.com/the-importance-of-hands-on-learning-in-higher-education/

Javid, H. (2024). Practical applications of Wazuh in on-premises environments [Bachelor's thesis, JAMK University of Applied Sciences.Theseus. https://www.thesus.fi/bitstream/handle/10024/862355/Javid_Hafiz.pdf

Lane, D.C., & Goode, C. (2021). Open for all: The OERu's next generational digital learning ecosystem. *International Review of Research in Open and Distributed Learning*. OER Foundation, New Zealand, Otago Polytechnic, New Zealand

Miller, F. (2009, July 30) Rationalizing it rationing: 10 ways to cut the IT budget (and what not to cut). EDUCAUSE. https://er.educause.edu/articles/2009/7/rationalizing-it-rationing-10-ways-to-cut-the-it-budget-and-what-not-to-cut

Safari, A.S.M., & Solkipli, J.F. (2025). Evaluating the role of open-source tools in enhancing cybersecurity training for ethical hackers. *International Journal of Advances in Engineering and Management (IJAEM).* 7(6), 423-430. DOI: 10.35629/52520706423430

Palo Alto Networks. (n.d.) Optimize operations with Cortex XSOAR. Retrieved May 14, 2026 from https://www.paloaltonetworks.com/cortex/cortex-xsoar

Proofpoint, Inc. (n.d.) Email security service: Threat protection solutions. Retrieved May 14, 2026 from https://www.proofpoint.com/us/products/threat-defense

Proofpoint, Inc. (2025). Proofpoint named a leader in 2025 Gartner magic quadrant for email security [Press release]. https://www.proofpoint.com/us/newsroom/press-release/proofpoint-named-leader-2025-gartner-magic-quadrant-email-security

Rose, S., Nightingale, J.S., Garfinkel, S., & Chandramouli, R. (2019). Trustworthy email (NIST Special Publication 800-177 Rev. 1). National Institute of Standards and Technology. https;//nvlpubs.nist.gov/nistpubs/SpecialPublications/Nist.sp.800-177R1.pdf

Splunk, Inc. (n.d.) Splunk Enterprise Security. Retrieved May 14, 2026 from https://www.splunk.com/en-us/products/enterprise-security.html

Splunk, Inc. (n.d.) Pricing – Workload pricing and ingest pricing. Retrieved May 14, 2026 from https://www.splunk.com/en+us/product/pricing.html

Sublime Security. (n.d.) sublime platform – A free and open platform for detecting and preventing email attacks [GitHub repository]. Retrieved May 14, 2026 from https://github.com/sublime-security/static-files

Sublime Security (n.d). Sublime labs. Retrieved May 14, 2026 from https://labs.subline.security/

Sutton, S., Bungei, V., Wang, X., Frank, J., & Djan, E. (2026). Teaching endpoint protection through Wazuh: A project-based approach to cybersecurity education. *Journal of the Colloquim for Information Systems Security Education.* 13(1). https://doi.org/10.53735/cisse.v13i1.244

Tines. (n.d.) Intelligent workflow platform. Retrieved May 14, 2026 from https://www.tines.com/

Tracecat. (n.d.) Tracecat – Open-source Ai-native SOAR. Retrieved May 14, 2026, from https://www.tracecat.com/

TracecatHQ. (n.d.) Tracecat: Open-source security automation platform for teams and AI agents [GitHub repository]. Retrieved May 14, 2026 from https://github.com/TracecatHQ/tracecat

Wazuh, Inc. (n.d.) Wazuh – The open source security platform. Retrieved May 14, 2026 from https://wazuh.com/

Wazuh, Inc. (n.d.) Getting started with Wazuh – Capabilities. Retrieved May 14, 2026 from https://documentation.wazuh.com/current/getting-started/index.html

Ugurluoglu, S. (2024). Reducing IT costs in Universities. Cut IT costs with open source. Revolutionize your tech. https://www.linkedin.com/pulse/reducing-costs-universities-serkan-ugurluoglu-cgqwf/

Vogel, V., & Golden, C. (2026, March 2). Smart spending: Inexpensive strategies for IT professional development in higher education. EDUCAUSE. https://er.educause.edu/articles/2026/3/smart-spending-inexpensive-strategies-for-it-professional-development-in-higher-education