The Use of Cyber-Defense Exercises in Undergraduate Computing Education

W. Michael Petullo, Kyle Moses, Ben Klimkowski, Ryan Hand, and Karl Olson
United States Military Academy

Abstract

This paper describes the placement of a large-scale cyber-defense exercise within the computer science and information technology curricula at an undergraduate institution, the United States Military Academy. Specifically, we describe the US National Security Agency Cyber-Defense Exercise as an example of a large-scale design, implement, and defend exercise. Furthermore, we provide evidence that the exercise inspires students to evaluate and create within the field of computer security. Our evidence includes examples of student research projects which benefited from unique opportunities for innovation. Finally, we provide the exercise documents that governed the 2016 Cyber-Defense Exercise and packet captures from our portion of the network.

1 Introduction

The Cyber-Defense Exercise (CDX) is an annual competition sponsored by the Information Assurance Directorate of the US National Security Agency (NSA). The CDX challenges the United States Service Academies[1] and the Royal Military College of Canada to design, implement, and defend an enterprise network against attack. The NSA provides the backbone exercise network and scoring infrastructure, acts as the competition referee, and fields a red cell with the task of compromising the confidentiality, integrity, and availability of the competitors' networks.

Each academy team builds a local network infrastructure in preparation for the CDX. A Virtual Private Network (VPN) connects each of these networks to the NSA's exercise network while isolating the entire exercise from the Internet and each organization's real-world network sensors. Once the exercise begins, the dedicated NSA red cell attacks each academy's network. (The CDX rules prohibit academy teams from executing network attacks.) The US Military Academy (USMA) has participated in the CDX since 2001.

Since the inaugural CDX, USMA has explored a number of strategies for integrating the event into its curriculum. In 2001, the CDX was the capstone event for a single course; this year we spread formal CDX preparations across a number of courses. Today, most participating USMA students receive time off from other classes and activities during the week of the actual CDX competition; this was not the case in 2001 [14, "Rules of Engagement"]. The design of our present curriculum helps students achieve a deeper understanding, and the CDX motivates them to contribute novel work in computer security. We describe examples of resulting research projects later.

[1] The United States service academies comprise of the United States Military, Naval, Air Force, Coast Guard, and Merchant Marine Academies. Each academy awards bachelor's degrees and provides service-specific training. Postgraduate institutions—including the Air Force Institute of Technology and the Naval Postgraduate School—have also participated in the CDX, competing in a separate graduate category.

2 Cyber-Defense Exercise design

Six documents describe the design of the CDX: the exercise directive, the network specification, the scoring specification, and the red-, white-, and gray-cell rules of engagement documents. The NSA publishes these documents in consultation with the faculty leadership of each participating team. We have made these documents available on the Internet [11].

The CDX divides its participants into four cells, labeled blue, white, gray, and red.

Blue The blue cell consists of the participating academy teams. While each academy team is actually competing against each other for points, they ostensibly form a single coalition network.

White NSA personnel make up the white cell, and they serve both as the higher headquarters for the exercise and as the exercise referees.

Gray Each academy's exercise network includes *gray* workstations which represent their users. These take the form of Virtual Machine (VM) images distributed by the white cell. To simulate large and changing networks, naïve users, zero-day vulnerabilities, and imperfect workstation hardening, these images often arrive bearing out-of-date installations, rootkits, and other compromises. Furthermore, the rules prohibit the academies from applying certain security updates to the software installed on the gray workstations. Within these constrains, each academy does their best to sanitize the gray workstations before integrating them into their network. NSA personnel make up the gray cell, and they operate each academy's gray workstations.

Red The exercise tasks the red cell with attacking the blue-cell networks. Thus the red cell attempts to leverage the gray workstation vulnerabilities along with other vulnerabilities they might find in order to compromise the security of the network.

The scored portion of the CDX spans four days, with students managing their systems from roughly 9:00 a.m. through 10:00 p.m. each day. The exercise network remains under attack during off hours, but the students cannot respond to off-duty attacks until the next morning.

The CDX also includes a number of *injects* and *forensic challenges*. Periodically, the white cell releases an exercise inject which forces the competing teams to react to events such as an unexpected user workstation or server image (either likely laden with malware) that must be added to the network

Permanent ID of this document: 1d99a38f432d84dc52936f334acc15d5.
Date: May 2, 2016.

on a short timeframe. The forensic challenges take the form of either Jeopardy-style Capture-the-Flag (CTF), malware-reverse-engineering, or host-/network-forensic puzzles.

The scoring of the CDX is primarily automated. One scoring agent continuously polls each service required by the exercise documents to assess availability. Another agent generates random-number tokens on each academy host which the red cell might compromise to violate confidentiality or integrity. It is each academy's responsibility to ensure the latter system exists on each of their hosts. The academy who loses the fewest points due to compromise or downtime and performs well enough on the injects and forensic challenges wins the competition.

3 Related work

Welch et al. described the first iteration of the inter-service-academy CDX [14]. Since then, a number of authors have described successive competitions. Of interest here are previous descriptions of CDX-curriculum integration at our own academy, the Air Force Institute of Technology, the Merchant Marine Academy, and the Royal Military College of Canada [4, 9].

A number of authors have proposed using competitions to produce labeled datasets for use in subsequent security experiments, such as testing intrusion detection systems. Sangster et al. proposed using the CDX in this way [13]. Since attacks during the CDX originate in some manner from red-cell hosts, placing sensors on the red- and blue-cell networks can help identify which packets represent malicious activity. Logs from individual hosts were found to further help corroborate classification.

Carlisle et al. of the Air Force Academy also described their curriculum, and in doing so raised some of the downsides of the CDX model [6, "Gaming the Curriculum"]. They claim CTF events provide clearer feedback, help maintain balance between the attackers and defenders, serve to better motivate students, and are less time-intensive. These were among the reasons the Air Force Academy did not participate in the 2016 CDX.

Chris Eagle of the Naval Postgraduate School and DEFCON fame echoes many of these concerns. In particular, he identifies the pervasive lack of feedback from the red cell during defensive competitions as an issue [7]. Furthermore, he asserts that some the exercise artificialities might hinder student learning [8]. We agree with many of the points raised by Carlisle et al. and Eagle, and we found the Air Force Academy's use of CTF-style events in introductory courses particularly innovative. However, our study demonstrates that with the right investment of resources and a good implementation strategy, events like the CDX provide a gainful educational experience that we could not recreate alone.

4 Curriculum integration and organization

The CDX is meant to be a significant educational experience for a select group of technology-focused students. Our teams primarily comprise of Computer Science and Information Technology majors. There are also a few students from other disciplines that have acquired a high level of relevant foundational knowledge. Many of our students will enter into one of the Army's communication-related fields, which often participate in the construction of networks. To them, the design and implement aspects of the CDX are directly relevant, and the defend phase informs their ability to build defensible networks. Other students will be assigned to the emerging cyber field which is responsible for network attack and defense; they directly benefit from the forensic, reverse-engineering, monitoring, and incident-handling aspects of the CDX. The experience of designing a sophisticated network from scratch provides insights which help prepare for participation in cyber operations in the highest capacity. Each participant gets to see the whole picture—from design to defense.

Indeed, the advent of cyber warfare has caused an increased demand for agile and innovative thinkers in the domain. We have moved away away from the static "castle" model of network security towards more focused and flexible paradigms, which require careful design and a deeper understanding of the network. These abilities follow partially from having conducted rigorous experiments with more advanced preventative and detection techniques.

There is no question that participating in a large-scale cyber-defense exercise is resource intensive. Most notable is the time spent by students and faculty preparing for the exercise, as well as acquiring the required theoretical knowledge and practical skills. Whether this commitment is worthwhile depends very much on the educational objectives and student outcomes of the program considering the exercise. This commitment is worthwhile for us, and the exercise provides a unique opportunity to address our objectives in ways that traditional classes cannot. Without the CDX, each academy would have difficulty recreating a purposeful, resourceful, and intelligent adversary; recreating persistent user behavior; and exposing their students to a network which approaches real-world sophistication.

In order to ensure our students and faculty have the time to meaningfully participate in the CDX, we integrate preparation into a number of courses. We also derive benefits from the USMA Cadet Competitive Cyber Team (C3T), an extracurricular club. A description of the components of this effort follows, and Figure 2 places the components in the context of an academic year. Our spring-semester CDX team totals around 25 students, most of which are in their final year of studying. Our honors program, C3T, and other classes provide our students with the requisite skills for the CDX.

Honors program Our honors program challenges qualified students to conceive of a year-long research project which they complete along with a faculty advisor. Students can select a CDX-related project such as with SIMPLEFLOW (§6).

C3T The C3T is a student-driven academic club which competes primarily in national and international CTF competitions. Some members of the C3T participate in the CDX,

and many of the skills developed during CTF competitions also apply well to the CDX. Most of the students who work to solve the CDX's forensic challenges come from the C3T.

Earlier classes There are a number of other classes which contribute to our students' preparation for the CDX. Our networking, network services, operating systems, cyber-security engineering, and forensic classes all develop the knowledge and skills required for success. Not every CDX participant must take all of these classes, but we find enough of them take each to produce a well-rounded team.

Students participate in the CDX itself under the auspices of a systems design course (CS/IT401–XE402) or an advanced independent study (CS489).

CS/IT401, Systems Design The core of the CDX team is built around students enrolled in CS/IT401. This course is the first of a two-course sequence which focuses on drawing from the skills and education acquired from our various disciplines while working on an interdisciplinary team tasked with completing a sophisticated project. Contributing to the CDX effort is one of the 17 projects our department offers. Around six students participate in the CDX project.

XE402, Integrative System Design XE402 is the follow-on class to CS/IT401. During this second semester course, students complete their research projects and provide completed deliverables and lessons learned to the CS489 students.

CS489, Advanced Individual Study The majority of our CDX team participates under the auspices of an independent study. Around 20 students research, design, and implement their part of the USMA exercise network before coming together as a team (joined by our 401–402 students) to defend the network during the CDX.

Figure 1 depicts the complete CDX team. Each box represents a role filled by one or two students, and each role covers a major task prescribed by the exercise documents.

The course structure we describe is not without its challenges. Most notably, it is rare during the spring semester that the CS489 and XE402 students which make up the CDX team formally meet in the same classroom. This is primary due to scheduling considerations (e.g., all seniors must take XE402, but the XE402 CDX project is limited to six students; this makes it impossible for CS489 and XE402 to meet at the same time), but is not entirely negative. Having the CDX span multiple sections simulates the realities found in large industry projects, and it seems to contribute to the sense of importance surrounding the CDX. Figure 1, our team's organizational chart, indicates XE402 students using a solid border and CS489 students using a dotted border. It falls on our faculty coaches to ensure communication among teammates takes place. Splitting the team leadership between the two courses also helps facilitate communication.

5 Implementation strategy

There are many noteworthy lessons we have learned regarding how to assist students as they build a complex design project like the CDX exercise network. Fostering the right environment in academic projects of this magnitude is challenging. Different portions of the project culminate at different times, and it is difficult to keep each student actively engaged. Similarly, some tasks are prerequisites for other tasks. A student who fails at a task or does not complete something satisfactorily might put other students' work at risk. These situations tempt an educator to step in and personally fix issues to prevent further setbacks. Our strategy provided flexibility and fostered student involvement without stifling innovation.

The key to our strategy was to lay the foundation with a small group of students in the fall semester and then incorporate the larger group in the spring. As mentioned earlier, six students formed the 401–402 portion of the team. Their focus was establishing the core network infrastructure and some key services. Having this infrastructure in place allowed the larger squad to focus on the details. The large squad focused on designing and implementing application-level services, planning incident handling procedures, and functionality testing.

While it might be intuitive that more time leads to higher quality work, there are deeper effects from investing in a year-long model. In a time-compressed implementation, one must make design compromises to get the network available, and this leads to vulnerabilities. Instead, we set out to impart in our students the possibility of removing some security vulnerabilities outright through sound designs. We reviewed the performances of previous years, exercised a deliberate engineering design process, and dissected potential threats via attack-tree analysis.

Indeed, many of the small 401–402 group's projects resulted from an analysis of our performance in previous competitions. We challenged our small squad to categorically remove the vulnerabilities which resulted in compromises the year before. We describe a number of the resulting student projects next.

6 Student projects

In previous years, the CDX served as the capstone project for a single USMA class, CS482: Cyber-Security Engineering. Under that model, students spent the first half of a semester receiving traditional lectures and performing hands-on assignments, and they spent the second half of the semester preparing for the CDX. While students under this approach had a positive educational experience, the shorter period typically resulted in a more chaotic network development, hasty decisions, and overlooked vulnerabilities. As we described above, the new model affords our six 401–402 students much more time to conduct research and to develop thoughtful, innovative solutions. Here we describe the named research projects of the small squad, and show that the depth of student learning can increase along with the additional time allocated.

SIMPLEFLOW Our first project exemplifies a deep student understanding of both system architecture and the nature of

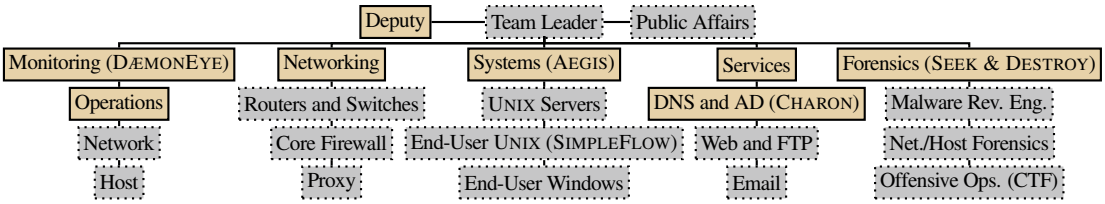

Figure 1: Team organization; solid border indicates CS/IT401–XE402 small-squad, and dotted border indicates CS489 large-squad

Fall	Spring	
Honors program (two CDX students*)		CDX
Cadet Competitive Cyber Team (six CDX students*)		
CS/IT401, three credits (six CDX students*)	**XE402**, three credits (six CDX students*)	
	CS489, three credits (twenty students)	

* Select students from these courses or teams participate in the CDX, while others participate in other projects.

Figure 2: The placement of the CDX-related courses within the USMA curricula along with credit and student counts

the threat model found in the CDX. SIMPLEFLOW consists of two components: the SIMPLEFLOW access control system and a custom network filter. Together, these components prevent certain malicious exfiltration traffic from leaving the exercise network. Recall that one of the aims of the red cell is to discover small files which represent confidential documents. Each academy aims to detect and stop the exfiltration of tokens from their exercise network, especially exfiltration by automated procedures.

The SIMPLEFLOW access control system is a mandatory access control system which implements an information-flow model inspired by—but much more rudimentary than—HiStar [15]. Two honors-programs cadets along with a faculty advisor wrote SIMPLEFLOW as a Linux Security Module. In SIMPLEFLOW, processes are either untainted or tainted, and system objects such as files, pipes, and UNIX sockets are either confidential or not. Opening a confidential object for reading taints the process, and subsequent writes by that process cause the receiving objects to themselves become confidential. Another consequence of a process becoming tainted is that the kernel will mark part of the header (the "evil bit field" [5]) of any IP packet a tainted process writes to the network. An administrator has the option of labeling programs as trusted, and such programs execute as processes unaffected by SIMPLEFLOW's access controls; one example of a trusted program is the NSA-provided scoring agent which manages the tokens placed on a host.

We depict a practical example of SIMPLEFLOW's mediation in Figure 3. Imagine an attacker has installed an automated procedure which occasionally attempts to exfiltrate tokens using the shell pipeline `cat secret | exfil`. Here we assume the command `exfil` sends the token over the network, perhaps hidden in an ICMP, DNS, or HTTP message. SIMPLEFLOW ensures the IP header on such a message bears the evil bit in the following way:

❶ `Cat` invokes the `open` system call to open the file `secret` for reading. Since this file is confidential, the kernel taints the process running `cat` when it reads from the file.

❷ The `cat` process forks a child, executes `exfil`, and writes the contents of the secret file over a pipe to `exfil`.

❸ SIMPLEFLOW marks the pipe as confidential because `cat` wrote to it while tainted.

❹ `Exfil` reads the contents of the secret file from the pipe and becomes tainted.

❺ `Exfil` prepares to send the contents of the secret file within an IP packet towards the attacker's command-and-control server. Since `exfil` is tainted, SIMPLEFLOW sets the evil bit on any IP packets `exfil` produces.

SIMPLEFLOW also sends log messages to DÆMONEYE (described below), so the team can monitor the taint status of the processes on the gray workstations.

The second part of SIMPLEFLOW is its network filter. The purposes of the filter are (1) to ensure that packets which contain sensitive data do not leave the network and (2) to extract information about the nature of the attacker's exfiltration attempt. Recall that the attacker might be using HTTP or another protocol on top of TCP as an exfiltration channel. The filter will block evil DNS request datagrams and TCP SYN segments, and so the compromised host might not send the application-layer request that otherwise would follow the initial connection setup. This missing information would be valuable to the defenders, as it reveals the attacker's attempted exfiltration channel. Returning to Figure 3:

❻ The network filter spoofs the DNS server and intended recipient of an evil-bit exfiltration packet, thus responding to DNS requests and completing the three-way handshake.

❼ `Exfil` begins sending its application-layer request. This message is blocked but recorded by the network filter.

SIMPLEFLOW does not impede normal network traffic:

❽ An unrelated process running `wget` is not tainted, and it sends an IP packet.

❾ The network filter routes this benign packet to the Internet.

Thus SIMPLEFLOW forbids the transmission of many exfiltration packets, while leaving other packets unimpeded. We note that SIMPLEFLOW does not stop all exfiltration. However, the remaining channels are slow and error-prone.

SIMPLEFLOW's presence in the Linux kernel makes it difficult for an attacker without elevated privileges to bypass or detect it. From the point of view of an attacker running programs on the compromised host, the exfiltration seems to be working as expected—each system call involved will succeed. There is little evidence as to why the network packets do not

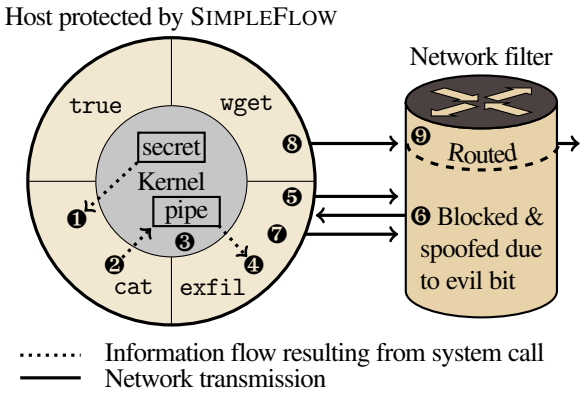

Figure 3: SIMPLEFLOW mediating `cat secret | exfil`

make it back to the attacker's command-and-control host. Only an attacker with the ability to observe the network packets on the USMA exercise network might detect what is amiss. Thus protecting confidentiality becomes equivalent to protecting the privileged account on the compromised host, even while allowing normal users legitimate access to the contents of confidential files. All of this serves to illustrate to our students the value of striving for complete mediation in system designs.

AEGIS One student was responsible for AEGIS, the hardened operating system upon which we built most of our services. Our students had encountered CentOS in other courses, and this, along with the quality of its documentation and its support for SELinux, led to its selection. AEGIS focused on building a minimal CentOS install to which we added security features, primarily in the form of a host firewall, stack-smashing countermeasures, and SELinux policy work. We installed the targeted SELinux policy and made modifications in order to satisfy various servers which we later installed upon AEGIS. SELinux can have a high learning curve, but we teach an SELinux lesson in our operating systems course (most of the rest of the course revolves around development within Pintos [12]). One of our students worked with an instructor to submit a report to Red Hat's bug-tracking system about an incompatibility between the squid proxy and Red Hat's targeted SELinux policy [2].

DÆMONEYE Last year, logging was a mildly-integrated afterthought. This year two students used the longer timeline to build DÆMONEYE, a logging infrastructure which aggregated a number of types of sensor feeds to provide end-to-end situational awareness.

DÆMONEYE uses the open-source Graylog log management system on top of AEGIS. It is capable of processing both host and network based logs and alerts. Some of the key tasks included installing and configuring Graylog log and web servers, installing and configuring syslog-ng on each host on the network to feed its events to Graylog, and securing the associated network traffic using Transport Layer Security (TLS). In addition, the students built a Windows-event-log bridge using NXLog to capture events from the few computers running Windows. Finally, Snort provided alerts, and our firewall provided NetFlow events.

We developed a baseline of DÆMONEYE as our first network service, and this served us well. Having DÆMONEYE in place before we added our user-supporting services such as email ensured that logging was a deliberate component of our network. Students gained ample experience in how to instrument a network for security auditing, how each component in a logging system fits together, the value of scripting repeated administrative tasks, the merit of package management systems, and the workings of TLS and X.509 certificates. Once the system was configured, we shifted the focuse from "build" to "operate." Students learned how to write Graylog searches, filters, and alerts, and how to perform incident detection and incident response leveraging this capability. We found students could visualize how events within their service were related to events elsewhere on the network, and this provided a key advantage during the competition.

A number of anecdotes support that our students achieved high levels of learning in this area:

(1) Our students extracted malware from captured packets, provided the malware to our forensics team for reverse engineering, and blacklisted the command-and-control domains embedded in the malware.
(2) Our students identified Cobalt Strike [1] as the red cell's tool of choice for operations management, and they denied particular command-and-control and exfiltration techniques used by Cobalt Strike. One student partially automated the blacklisting of DNS domains using a script that watched for Cobalt Strike signatures.
(3) Our students recreated malicious events on our gray-cell workstations by studying the notifications provided by the various network sensors and logs they had installed. They did this without persistent, direct access to the grey-cell workstations
(4) Our students learned to integrate and troubleshoot disparate systems. For example, an incompatibility between our firewall and Graylog required rewriting a considerable amount of plug-in code to support NetFlow version 9. We contributed this improvement to the Graylog project.

CHARON The CHARON project categorically denied an attack vector which devastated our team in the previous year. The compromise of our Windows-domain-administrator password during the 2015 CDX was catastrophic.

We set out to implement multi-factor authentication for the administrative accounts on our Windows domain controller. In our last network, the domain controller was our only Windows server; this year we built CHARON using FreeBSD and Samba because the red cell is particularly adept at exploiting Windows. FreeBSD provided another advantage over Windows; our team updated Samba to address the Badlock vulnerability [10] without losing points due to a reboot.

Eventually, time and procurement constraints forced us to abandon the smartcard-based multifactor authentication com-

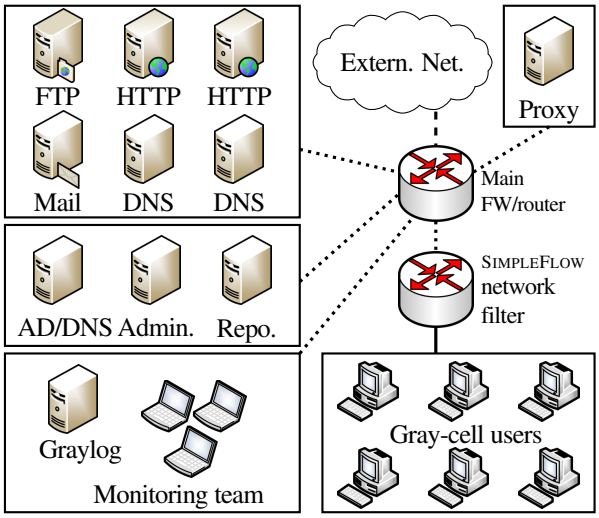

Figure 4: Logical network topology; three different line types represent three different packet capture points

ponent of CHARON. Instead, we implemented strict procedures requiring the use of a special administrative account we named `kamikaze` when performing domain-administrative work on the gray-cell workstations. We changed the `kamikaze` account password after each use and set the account inactive when not in use. To reduce the likelihood of errors, we ensured that a pair of students always worked together when interacting with the gray-cell workstations. Still, humans often make mistakes under stress, and so we hope to resume work on multifactor authentication for next year's competition.

SEEK & DESTROY Our students knew they would receive gray workstations which were already compromised. How would they find the vulnerabilities among billions of bytes? Products such as Tripwire [3] are not helpful, because the gray workstations arrive without a record of pre-compromised file hashes. Thus SEEK & DESTROY became a deep foray into the layering found in modern operating systems. What our students realized is that they could recreate the software present on the workstation using trusted repositories and then compare hashes across the two installs. They focused on categories of decreasing impact: boot loader, kernel, kernel modules, setuid programs, and finally regular programs. They also spent time learning about shared libraries, including how they interact with these categories of software and how to configure the dynamic loader.

Core exercise network The students responsible for the core exercise network designed the use of and configured the network's central firewall, routers, switches, and application-layer proxy as depicted in Figure 4.

This year, we incorporated the core network into the CDX capstone project. Consequently, basic network functionality was achieved by the end of the fall semester including a very detailed and accurate network diagram that was shared with the entire team. This time-phased and heavily-documented approach allowed the networking team to work through the development process without significant negative effects on network availability, resulting in a much less frantic atmosphere in the weeks before the actual exercise.

Category	USMA	Average	Max
Availability	**85.96%**	52.60%	**85.96%**
Conf./Integrity	77.00%	81.63%	100.00%
Gray Cell	65.50%	44.21%	84.53%
Compliance	-4.75%	-11.81%	0%
Capture-the-Flag	71.00%	59.14%	86.00%
Malware Analysis	**100.00%**	38.86%	**100.00%**
Forensics	93.00%	51.14%	100.00%
Weighted Total	**78.14%**	50.68%	**78.14%**

Table 1: Final USMA, average, and maximum CDX scores

In the spring, the networking team focused primarily on network monitoring tasks, access-control-list development, further documentation, and tuning application-layer content filtering at the firewall and proxy servers. This preparation allowed the networking team to quickly develop novel filters to block red-cell exfiltration attempts shortly after the red cell attack window opened.

6.1 Collective training

An important part of building an effective team is *collective training*, training focused on what the team does as a group rather than mastering individual skills. In past years, collective training has taken place during the CDX itself due to the network remaining incomplete up until the competition. This year, we completed much of our network some weeks before the CDX started. This allowed us to complete more deliberate penetration testing and collective training.

Our collective training consisted of faculty coaches generating benign, malicious, and suspect network traffic of the types we expected during the CDX while the students reacted to what they saw. For example, we generated ICMP packets with large payloads which represented the exfiltration of confidential data. We watched to ensure the team detected the strange traffic, discussed what it might represent, and decided how to handle the event. We saw our students develop procedures which crossed squad boundaries: for example, the logging squad might detect an anomalous packet and alert the team leader who would direct the firewall to block a destination IP. We worked through a number of scenarios, and the students gradually began to feel comfortable mitigating these events, generally by manipulating blacklists in our firewall, web proxy, or DNS server. The students formalized a number of these procedures before the CDX began.

7 Results

The USMA team won the overall 2016 CDX competition. Table 1 lists our scores, the average team score, and the maximum score achieved by a team for each scoring category. It is important to point out that each academy approaches the CDX with different resources, numbers of faculty coaches, and curricular designs. Each academy also approaches the

CDX itself in a different way, as evidenced in §3. A detailed analysis would be required to compare across each academy's model. However, our approach clearly produced a positive outcome. Most importantly, we did not win at the expense of education, rather to its benefit.

We used surveys to collect student feedback throughout the exercise in order to assess student growth and the effectiveness of our course design. Figure 5 summarizes the students' introspective assessments with respect to our 11 course objectives. It illustrates growth in all areas—a categorical increase in the mean for every objective and a tightening of the standard deviations. Notably the students overall increased their confidence in defending networks and designing them in a robust manner. We suspect the marginal increase in the engineering design process is due to the disproportional responsibility of the small squad for this task. The larger group, in contrast, was more focused on refinement that required a more myopic design.

The open feedback we solicited from the students was illuminating. We asked the students which aspect they would change about their CDX experience. Figure 6 maps their responses into seven themes. The most frequent was improving team mechanics (communications, drills, practice, etc). This suggests the students can see how their individual component fits into the overall scheme, highlighting their intellectual maturity. Moreover, this realization can only happen after students become individually capable of creating and evaluating within their areas of responsibility.

Only one student stated that the exercise should be more realistic. Throughout the exercise, the scoring system and exercise artificialities did occasionally cause the students to make decisions that were unrealistic. As Eagle has mentioned, these unrealistic scenarios could potentially take away from the educational experience [8]. For instance, on at least one occasion, a compromised image was left on the network to keep "availability" points at the expense of "confidentiality" points. Throughout the exercise, the instructors compared and contrasted student decisions with what they might do in actual practice. The student feedback suggests that the exercise artificialities are not an impediment to their education. Indeed, this dialogue fosters learning, and prompted student-faculty discussions that would not have otherwise taken place. Our results do not invalidate Carlisle et al. and Eagle's claims, but rather they show that the CDX is very beneficial with the right approach.

We have published online a data set named 2016-CDX-USMA [11] which contains many of the 300,000,000 log messages DÆMONEYE captured during the course of implementing and defending the CDX network. The data set also contains our gray-cell workstation images and 489 GB of captured packets and related data. We depict in Figure 4 the span of our packet captures: our sensors captured packets from outside of our core firewall, from our inside subnets except the end-user subnet, and from the end-user subnet. We have released all of these data to the public domain, and are in the process of labeling and curating the data.

8 Conclusion

The CDX challenges our students to design, build, and defend the most sophisticated network they encounter during their time at the United States Military Academy. Our network this year spanned 27 VMs, and it included servers, management workstations, and the gray-cell user workstations; four network devices; six operating systems; a range of server software; a range of monitoring software; and a number of custom-written tools. Furthermore, the CDX provides students an experience in leading a 26-man team.

We found five decisions especially contributed to the CDX as an educational experience:

- span the exercise software and network development over one year,
- challenge students to remove categories of vulnerabilities by design,
- build the logging/monitoring systems first, and only later add end-user services,
- derive benefits from complimentary efforts such as the USMA C3T, and
- perform collective training scenarios before the CDX.

Despite our team's capable performance, we ultimately tried to leave our students with a sense of dissatisfaction. While we won the 2016 CDX, we did not achieve the goal of fully protecting the confidential information on our network. Indeed, the red cell compromised the tokens within every functioning academy network with ease. What then, does this say of the future? The final thought we imparted to our students is that today's state-of-the art computer systems have achieved a high degree of reliability, but not robustness—they lack the ability to maintain confidentiality, integrity, and availability in the presence of an intelligent adversary. Perhaps this is the best lesson of the CDX: that despite our best efforts, we could not stay in front of the red-cell. This reinforces the observation that professionals struggle to defend their systems; the news media is full of stories that demonstrate this.

Modern society places its trust in protocols and systems which are not trustworthy, and our students during the CDX experience the effects of this fact. Vulnerabilities are introduced across a myriad of sources; security researchers must address robustness during development, administration, and end-use. The next generation of computer scientists and information technology professionals must vastly improve on our current systems. The experience of designing, implementing, and defending a network against the NSA red cell provides one small piece of an education that we hope will well serve the next generation as they produce the more robust computer systems our society requires. Defense is the Achilles' heel; our nation is exposed, and our students should be encouraged to confront the challenge, not run away from it.

Acknowledgments

This material is based upon work supported by the US National Science Foundation under grant CNS-1464121.

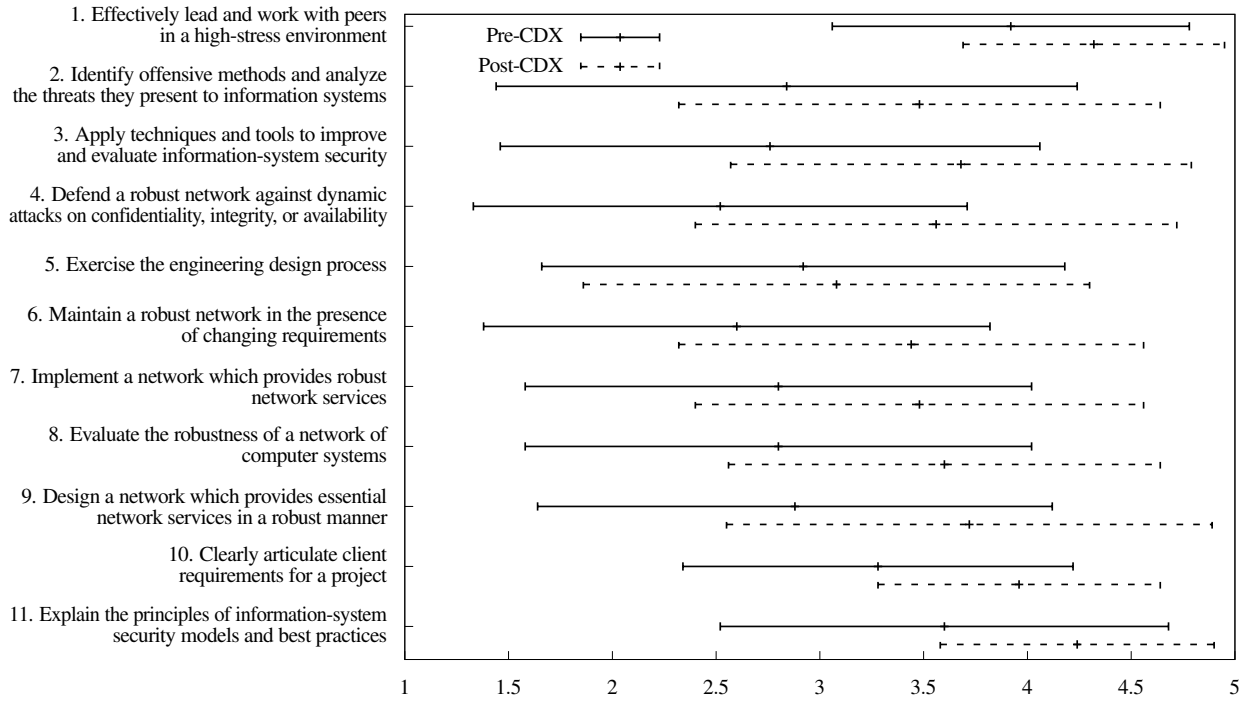

Figure 5: Student self-assessment: each student rated themselves over a scale of 1–5 before and after the CDX

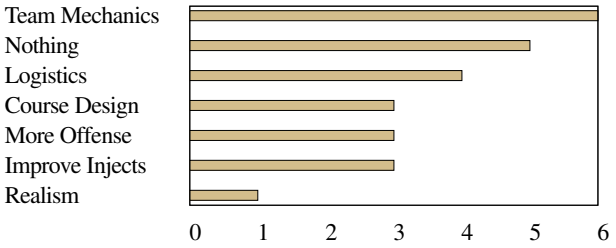

Figure 6: Broad categories of responses when students asked about improvements to the CDX (longer lines indicate more students identified a category as ripe for improvement)

David Raymond, the previous USMA CDX coach, handed us the reigns to a healthy CDX program. We also thank the cadets of the 2015 and 2016 USMA CDX teams, the competitors at our fellow academies, the CDX team at the NSA, and Michael Lanham of the USMA Cyber Research Center.

References

[1] Cobalt Strike. `https://www.cobaltstrike.com/` [Accessed Apr 19, 2016].

[2] SELinux policy breaks squid's `ssl_crtd`. `https://bugzilla.redhat.com/show_bug.cgi?id=1325527/` [Accessed Apr 21, 2016].

[3] Tripwire. `http://www.tripwire.org/` [Accessed Jan 28, 2016].

[4] ADAMS, W. J., GAVAS, E., LACEY, T., AND LEBLANC, S. P. Collective views of the NSA/CSS Cyber Defense Exercise on curricula and learning objectives. In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET 2009)* (August 2009).

[5] BELLOVIN, S. RFC 3514: The security flag in the IPv4 header. `https://www.ietf.org/rfc/rfc3514.txt` [Accessed Jan 20, 2016], Apr. 2003. Status: INFORMATIONAL.

[6] CARLISLE, M., CHIARAMONTE, M., AND CASWELL, D. Using CTFs for an undergraduate cyber education. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)* (Washington, D.C., Aug. 2015), USENIX Association.

[7] EAGLE, C. Computer security competitions: Expanding educational outcomes. *IEEE Security & Privacy*, 4 (2013), 69–72.

[8] EAGLE, C. From CTF to CAE, 2013. `http://infiltratecon.com/chriseagle.html` [Accessed Apr 27, 2016].

[9] MULLINS, B. E., LACEY, T. H., MILLS, R. F., TRECHTER, J. E., AND BASS, S. D. How the Cyber Defense Exercise shaped an information-assurance curriculum. *IEEE Security & Privacy 5*, 5 (2007), 40–49.

[10] NIST NATIONAL VULNERABILITY DATABASE. CVE-2016-2118. `https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2118`, Apr. 2016. [Accessed Jun 14, 2016].

[11] PETULLO, W. M., MOSES, K., KLIMKOWSKI, B., HAND, R., AND OLSON, K. 2016-CDX-USMA data set and exercise documents, 2016. `https://www.flyn.org/CDX/` [Accessed Apr 27, 2016].

[12] PFAFF, B., ROMANO, A., AND BACK, G. The Pintos instructional operating system kernel. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2009), SIGCSE '09, ACM, pp. 453–457.

[13] SANGSTER, B., O'CONNOR, T. J., COOK, T., FANELLI, R., DEAN, E., ADAMS, W. J., MORRELL, C., AND CONTI, G. Toward instrumenting network warfare competitions to generate labeled datasets. In *Proceedings of the 2nd Conference on Cyber Security Experimentation and Test* (Berkeley, CA, USA, 2009), CSET'09, USENIX Association, pp. 9–14.

[14] WELCH, D., RAGSDALE, D., AND SCHEPENS, W. Training for information assurance. *Computer 35*, 4 (2002), 30–37.

[15] ZELDOVICH, N., BOYD-WICKIZER, S., KOHLER, E., AND MAZIRES, D. Making information flow explicit in HiStar. In *Symposium on Operating System Design and Implementation (OSDI)* (Seattle, Washington, Nov. 2006).