The EDURange Framework and a Movie-themed Exercise in Network Reconnaissance

Richard Weiss
The Evergreen State College

Jens Mache
Lewis & Clark College

Michael E. Locasto
SRI

Abstract

Total Recon is a hands-on cybersecurity exercise designed to teach about network reconnaissance, using a movie theme to make it more exciting for students. Students use `nmap` and netcat (`nc`) to investigate hosts on a large network. The multiple levels of the game provide scaffolding that allows students with a wide range of preparation to play the game. The exercise is implemented in the EDURange framework, and according to our surveys, both students and faculty have found the exercises to be very engaging. This short paper describes both the EDURange framework and the Total Recon exercise.

1 Introduction

Total Recon addresses learning goals at multiple cognitive levels:

- It teaches skills such as how to use `nmap`, `nc`, and `tcpdump`, important network reconnaissance tools.
- It teaches knowledge of networking protocols such as TCP, UDP and IP, including CIDR notation.
- It teaches analysis abilities that are required for understanding how to efficiently map a large IP range, and how to use protocols in ways for which they were not intended.

The motivation for creating exercises such as Total Recon was to create exercises that were not too prescriptive but would provide enough guidance so that students could succeed without a comprehensive, encyclopedic knowledge of computer science and cybersecurity. In addition, we wanted to teach analysis abilities and the principles of the *Hacker Curriculum* [1, 2]. *Analytical abilities* include reasoning about large, complex, and opaque data and systems.

2 Total Recon

Total Recon is a multilevel exercise where the student must login to successive hosts, and the required information is obtained by solving puzzles. To better engage students in the learning activity, we design our levels so that they are woven into the plot of the movie "Total Recall". There are eight levels and as levels are solved, students progress through parts of the story. This mechanism attempts to provide students with an additional level of engagement and motivation for solving the challenges. Total Recon is based primarily on the Linux utility `nmap`. This is a powerful tool that allows users to send packets to a variety of ports with different protocols.

One challenge of Total Recon is the size of the IP range that students must search. It is 16K addresses and could take more than 15 minutes with default `nmap` options. Another major challenge is that some of the hosts do not respond to ping. With the default options, `nmap` will first try to ping an IP address. If it doesn't receive a response, it will assume that there is no active host at that address and will not check TCP or UDP ports. The option `-Pn` must be used skip this host discovery step. Even in this case `nmap` only scans 1000 common ports. The first scan is used to detect if a host is alive, but then one needs to use the option `-p-` to scan all of the ports.

In level 4, if one scans the subnet, it takes more than 15 min. However, the challenges are scaffolded, and for the lower level challenges, the IP range is smaller so that students can gradually become familiar with `nmap`. For example, in one of the challenges, there is a constraint on the IP address, so that one can reduce the time from 15 min to 2 seconds by applying that constraint.

At the next level, a firewall is introduced that can block traffic in and out of the network, restricting the information that `nmap` can obtain. However, there is one IP address that is not being filtered and the student must search through the data to identify it based on the difference in the response.

In one of the levels, `nmap` is not available and the student must use `nc` to get information. However, `nc` produces a significantly more data which needs to be searched, for example using regular expressions in `grep`. In addition to becoming familiar with another tool, students develop the ability to search and analyze data. This is where the movie theme can be important. Typically, students are not highly motivated to search through data

unless the results are meaningful to them and they are engaged.

In additional levels, students are introduced to the advantage of using stealthy scans which can result in response messages which give extra information. `ssh` is normally on port 22, but can be running on any port. Students can run version scans, to find if a port is running `ssh`.

The scenario culminates in a final challenge that requires additional Linux command line skills to "turn on" the reactor and win the game. Many of these levels of the scenario are challenging, but often hints are provided so that students are less likely to get stuck. In many of the tasks, students must find a balance between finishing faster and spending more time to get more thorough information.

3 EDURange Framework

The EDURange framework provides the infrastructure for crafting cybersecurity scenarios on multiple VMs in a virtual network [14, 12, 15] . It is implemented on top of Amazon's EC2 using AWS. EDURange currently provides other *scenarios* besides Total Recon for exploring particular security concepts and tools.

EDURange addresses both the student and instructor perspectives:

- *Engaging* for students. Students from a variety of backgrounds can learn practical security concepts, tools, and skills in puzzle-like scenarios involving realistic challenges. They are not too prescriptive and allow for creative solutions.
- *Scaffolding* and *assessment* to support students to achieve learning objectives.
- *Ease-of-use* for students and instructors. Scenarios run on VMs that are created in the Amazon Cloud using scripts. Students don't need special software and it can be used anywhere with Internet service. Instructors register their class members, often grouping them into teams of students with accounts on the same VM, facilitating scenarios involving collaboration. The EDURange system collects data to make assessment easier. Faculty can sign up on our website (`http://www.edurange.org`) and their classes can use Total Recon.
- *Flexibility* for instructors to use simple scripts to specify exercises at a high level and create variations. This enables them to tailor exercises to their specific classes and student backgrounds and continue to modify them in order to minimize risk of students finding the answers online.

We have held several tutorials for faculty, and the recent one at SIGCSE'17 was attended by about 30 faculty. The participants we spoke with were enthusiastic about using EDURange. We have used EDURange exercises in many classes, and surveys indicate that students also enjoyed them [16]. We gave a survey of student interest in an undergraduate security class in fall 2016. The results are shown in Table 1. During the semester, the students were assigned many cybersecurity exercises from a variety of sources, and two of them were from EDURange. Students who took the survey were asked to score the exercises on a scale from 0-10 on how worthwhile they felt they were. The two EDURange exercises, Total Recon and ssh inception, were ranked 3^{rd} and 4^{th} based on their scores. The top two were the time-limited NCL "regular season" competition (a two-part event in fall 2016) that generated a national ranking, but other parts of the NCL were ranked below EDURange. This suggests that the excitement of the national competition may have contributed to students evaluation. The EDURange exercises are not competitive, although they could be adapted to that mode.

Exercise	Average score	Rank
NCL regular season game1	9	1
NCL regular season game2	9	1
EDURange Total Recon	8.7	3
EDURange ssh inception	8.1	4
NCL pre-season	8	5
NCL post-season	7.8	6
vulnerable banking web app	7.1	7
crypto CTF [5]	7	8
deterlab "intro to UNIX"	6.6	9
class project	6	10
SEED Android repackaging	5.8	11
SEED buffer overflow	5.8	11
firewall simulation	5.7	13
NCL labs (in the "gym")	5.6	14

Table 1: Student survey results: how worthwhile each exercise was, on a scale from 0-10

The Hacker Curriculum and security mindset

The scenarios in EDURange are inspired by the *Hacker Curriculum*, which is based on a collection of papers from hacker conferences and blog posts that describe how software and hardware really works (or doesn't). They describe boundary cases, failure modes, API implementations, debugger implementations, linkers and loaders, and tools for observing and analyzing software and hardware artifacts. The abstractions students are commonly taught in early courses as a way to simplify understanding of sys-

tems are an impediment when it comes to understanding security; exploits are often based on knowledge of system details that cuts across these artificial boundaries. The hacker curriculum encourages a *security mindset* [10] that focuses on vulnerabilities in systems, how to exploit them, and how to guard against them.

One of the primary goals of EDURange's exercises is to nurture *analytical abilities* that support the security mindset — reasoning about large, complex, and opaque data and systems. For example, in the Total Recon exercise, it would take a long time to scan the entire network using the default options. This requires students to develop analytical abilities that enable them to understand the protocols used and ways to decompose the search space. These are precisely the kinds of skills or abilities that we believe are useful in many cybersecurity scenarios, from security policy design to reverse engineering to vulnerability analysis. We use the phrase *analytical abilities* in keeping with the terminology of knowledge, skills, and abilities (KSA) that are used to describe the requirements for different security roles. Knowledge is what we think of as information, such as the fact that TCP uses a three-way handshake. Skills are measured by performing tasks, for example which options for `nmap` will use a SYN scan and which will perform a full TCP scan. Mastering the skill of using `nmap` is important, but one still needs to be able to analyze the results. For example, searching through a large network to find a host with an open ssh port, not on the standard port, and is not being filtered by a firewall.

The learning goals for the Total Recon scenario also include specifics of the TCP protocol, such as the 3-way handshake. We would also like students to understand issues with predictable sequence numbers and other implementation weaknesses that can potentially be exploited, but the first step is to become familiar with the basic structure.

Assessment in EDURange

One of the features of EDURange is to provide instructors with assessment tools to see in real time how their students are doing and help them identify students who are missing pieces of the required background. EDURange supports assessment by providing the instructor with the `bash` histories of each of the students, so that it is possible to identify misconceptions early on in the exercise. EDURange has been *instrumented* to track user activities in such a way that they are more easy to analyze. This information, including timestamps and exit status for all commands, can be accessed by the instructor at any time.

We are experimenting with visualizations of the `bash` history data that allow an instructor to quickly determine how far students have gotten in a scenario and whether they might need guidance.

When students are spending hours or even days on a cybersecurity exercise or challenge, it is our responsibility as instructors to give them feedback when they need it. That can be a problem when there are many students and assessing student progress is complex. Having the right tools can make a big difference. this is also true for programming, where there are a few online systems such as zylabs in ZyBooks[1] or Codelab in Turingscraft[2] in which students type or upload their solutions and the system compiles and runs the code on test cases provided by the content creator. These systems are good for very simple coding problems but can be overly rigid because it is difficult to specify multiple solutions. We are still a long way from where we want to be. Cybersecurity exercises can be open-ended and are even more difficult to assess.

On one end of the spectrum are prescriptive exercises, in which students follow step-by- step instructions to run scripted exploits, perform penetration testing, do security audits, etc. On the other end of the spectrum are open-ended exercises and many capture-the-flag (CTF) activities, where little guidance is given on how to proceed. In addition, an important aspect of educational exercises is the question of how to assess learning.

We try to find a balance between the two extremes for guidance in the context of one of the suite of cybersecurity exercises that we have developed [15]. The particular exercise that we present teaches students about dynamic analysis. We have found that students are most successful in these exercises when they are given the right amount of prerequisite knowledge and guidance as well as some opportunity to find creative solutions. Our scenarios are specifically designed to develop analytical abilities and the security mindset in students and to complement the theoretical aspects

Other Scenarios

The Total Recon scenario is just one of several currently provided by EDURange. Others include:

- *ssh inception*: Students use `ssh` and `nmap` on the command line to find IP addresses of hosts and connect to them as part of a level-based game. This exercise was designed and implemented by our students, who saw that many of their peers were struggling with the command line and navigation around the network.
- *strace*: This is based on the Linux utility `strace`. It allows users to see the system calls associated with a process and thus discover possible malicious activity, e.g. reading or writing files, forking processes,

[1] `http://www.zybooks.com`
[2] `turingcraft.com`

or opening network connections. Exercises included in this scenario involve determining what a mystery executable does and determining how `strace` was invoked to yield a given transcript.

- *scapy hunt*: Students sniff packets to understand which hosts are communicating and the protocols they are using.
- *ELF infection*: This is a reverse engineering challenge, where students are given infected binaries and they must locate the malicious code.
- *treasure hunt*: Students must find the contents of the secret files of sixteen faux users in a Linux system using exploits like code injection, PATH exploits, directory traversal attacks, elevation of privilege, and password cracking. This builds on students' understanding of the command line and Linux utilities.

We are working on new scenarios for password cracking, forensics, cryptography, firewalls, and buffer overflows. We encourage readers to submit their ideas at `http://www.edurange.org`.

4 Related and Future Work

There is a growing number of engaging hands-on exercises, but EDURange remains distinct in that it both focuses on analytical abilities and is easy to use in the classroom. Scenarios can be used as modules in a wide range of courses, including computer security, networking, and operating system courses. Research shows that some hands-on exercises increase student interest in cybersecurity [13, 17].

DETERLab [8, 7] has a variety of exercises contributed by several schools, from code injection to DDoS, but modifying existing exercises has a steep learning curve. Security Injections [11] focuses on secure coding rather than analytical abilities. NICE Challenge Project provides a set of goal-oriented, unguided, open-ended online exercises[3], while EDURange tries to provide more scaffolding. SecKnitKit provides a set of exercises that can be integrated with Networking, OS, Database, and Software Engineering courses, but requires downloading and installing VMs. Some instructors prefer to have students become familiar with this process, and it is a useful skill, but it can be a distraction, especially when students have laptops which do not meet the resource requirements. The SEED Project [4] has a large number of advanced exercises, including buffer overflows and malware analysis, but requires downloading VMs. GENI was developed as an environement for network research, but it has also been used for some cybersecurity exercises with a heavy emaphasis on the networking aspects, for example denial of service attack [6]. Only DETERLab and EDURange provide an environment for instructors to create or modify exercises.

There are also several collegiate competitions that are aimed at attracting students to cybersecurity, e.g. NCL[4], CCDC[5], and CSAW[6]. NCL holds a couple of competitions each year and has made some of their challenges available outside of the competitions, so that students can practice inside and outside of the classroom. However, the competition and training periods are still very limited, and the questions are not well integrated with tutorials and courses. We and others have observed that many Capture-the-Flag contests (CTFs) and other competitions are not geared toward novices [9] and can be frustrating for novices because they assume knowledge that is often not taught in courses [3]. EDURange-style scenarios enable students to develop skills and abilities that can prepare them for such competitions.

Although EDURange allows students to explore topics in cybersecurity at their own pace, it does not eliminate the instructor. Nor do the exercises by themselves ensure that students will learn analytical abilities and the security mindset. Rather they are designed to allow for instructors to open that discussion, and they provide tools that can help the instructor with assessment and giving rapid feedback to the students. Instructors can also tweak instructional material and assignments for the class to match their level of understanding. Students can use the feedback to reflect on what they have learned or still have questions about.

EDURange has the potential to foster a learning community and encourage collaborative work among students and faculty. The fact that our students were able to create new scenarios is a testimony both to their creativity and the ease of use of the system. Hosting the exercises on Amazons EC2 is consonant with this goal. EDURange is free for faculty and their classes through an educational grant from Amazon and a grant from the National Science Foundation. Instructors can sign up by filling out a form at `http://www.edurange.org`. We encourage faculty to try the exercises and contribute their ideas.

Assessment of student learning is one of the most important jobs that we have as instructors, and for cybersecurity it is a difficult task because of the range of possible solutions. EDURange is beginning to provide instructors with tools that can help them identify problems that students are having and give then feedback.

[3] `https:www.nice-challenge.com`

[4] `http:www.nationalcyberleague.org`

[5] `http:nationalccdc.org`

[6] `https://csaw.engineering.nyu.edu`

Acknowledgments

We thank the EDURange team, especially Erik Nilsen and students Stefan Boesen, Mark Grossman, Kahea Hendrickson, Jeff Ladish, Yesha Maggi, Nick Stephens, and Noah Weiner. This material is based upon work supported by the National Science Foundation under grants 1516100 and 1516730, and by an educational grant from Amazon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] BRATUS, S. What hackers learn that the rest of us don't: Notes on hacker curriculum. *IEEE Security and Privacy 5* (2007), 72–75.

[2] BRATUS, S., SHUBINA, A., AND LOCASTO, M. E. Teaching the principles of the hacker curriculum to undergraduates. In *Proceedings of the 41st ACM technical symposium on Computer science education* (New York, NY, USA, 2010), SIGCSE '10, ACM, pp. 122–126.

[3] CHUNG, K., AND COHEN, J. Learning obstacles in the capture the flag model. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)* (2014).

[4] DU, W., AND WANG, R. Seed: A suite of instructional laboratories for computer security education. *Journal on Educational Resources in Computing (JERIC) 8*, 1 (2008), 3.

[5] FENG, W. A "divergent"-themed CTF and urban race for introducing security and cryptography. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)* (Austin, TX, 2016), USENIX Association.

[6] LEDFORD, H., MOUNTROUIDOU, X., AND LI, X. Denial of service lab for experiential cybersecurity learning in primarily undergraduate institutions. *Journal of Computing Sciences in Colleges 32*, 2 (2016), 158–164.

[7] MIRKOVIC, J., BENZEL, T., FABER, T., BRADEN, R., WROCLAWSKI, J., AND SCHWAB, S. The deter project: Advancing the science of cyber security experimentation and test. In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on* (2010), pp. 1–7.

[8] PETERSON, P. A., AND REIHER, P. L. Security exercises for the online classroom with deter. *Proc. of the 3rd USENIX CSET* (2010).

[9] PUSEY, P., GONDREE, M., AND PETERSON, Z. The outcomes of cybersecurity competitions and implications for underrepresented populations. *IEEE Security & Privacy 14*, 6 (2016), 90–95.

[10] SCHNEIER, B. Inside the twisted mind of the security professional. *Wired Magazine 16*, 3 (Mar. 2008).

[11] TAYLOR, B., AND KAZA, S. Security injections: modules to help students remember, understand, and apply secure coding techniques. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (2011), ACM, pp. 3–7.

[12] WEISS, R., LOCASTO, M. E., AND MACHE, J. A reflective approach to assessing student performance in cybersecurity exercises. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (New York, NY, USA, 2016), SIGCSE '16, ACM, pp. 597–602.

[13] WEISS, R., MACHE, J., AND NILSEN, E. Top 10 hands-on cybersecurity exercises. *Journal of Computing Sciences in Colleges 29*, 1 (2013), 140–147.

[14] WEISS, R., TURBAK, F., MACHE, J., AND LOCASTO, M. E. Cybersecurity education and assessment in edurange. *IEEE Security & Privacy 15*, 3 (2017), 90–95.

[15] WEISS, R., TURBAK, F., MACHE, J., NILSEN, E., AND LOCASTO, M. E. Finding the balance between guidance and independence in cybersecurity exercises. In *2016 USENIX Workshop on Advances in Security Education (ASE 16)* (Austin, TX, 2016), USENIX Association.

[16] WEISS, R. S., BOESEN, S., SULLIVAN, J. F., LOCASTO, M. E., MACHE, J., AND NILSEN, E. Teaching cybersecurity analysis skills in the cloud. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (2015), ACM, pp. 332–337.

[17] YUAN, X., HERNANDEZ, J., AND WADDELL, I. Hands-on laboratory exercises for teaching software security. In *Proceedings of the 16th Colloquium for Information Systems Security Education* (2012).