CYBRARY

Guerrilla Red Team:
Decentralize the Adversary

Authored by
Christopher Cottrell | *Cybrary Fellow*

Executive Summary

Business Case

One of the main goals of any Cybersecurity program should be to align with the business. The protective measures enacted should embolden the company without interfering with day-to-day operations. Promoting security culture and awareness throughout the organization raises the base difficulty for an attacker to gain an initial foothold. Without security buy-in from those at the front lines, security engineers remain in a state of hyperawareness, waiting to take action on an eventual compromise. Remaining in this state for too long leads to fatigue, burn out, and a lack of empathy for security requirements.

Mature Cybersecurity programs will often employ what is known as a red team, trusted engineers that mimic cyber adversaries, to alleviate their security engineer's hypervigilance. Members of the red team traditionally exist in isolated verticals that test and improve security for the business. When a red team becomes isolated, they provide far less value to the business than they could otherwise. An isolationist red team leads to creative stagnation, distrust from outside staff, and loneliness for the red team operators.

This white paper aims to answer the question of how a red team can expand the reach of the offensive security program outside of their vertical, increase advanced security consciousness throughout the business, and promote grassroots security evangelism.

All can be done with minimal financial resources to promote lean, scalable, and repeatable processes. A lean posture ensures continuity of the program through the most stringent budget cuts.

Methodology

Guerrilla Red Team is a methodology by which a company can grow security consciousness, technical expertise, and security brainpower, resulting in an internal mesh network of trusted ethical hackers. The program requires minimal capital investment from the hosting red team[1]. It achieves its primary goals through weekly group mentorship hosted during a four-hour block,

[1] Capital investment refers to a VIP subscription to Hack the Box, valued at $20 USD per month per student, and two text books valued at $50 USD total

once per week, during the workday[2]. It forms a peer network in which guerrilla operators share ideas and techniques, and ultimately grow technically and professionally as a group. Members of the program come from various technical disciplines, but not necessarily security-focused verticals. The cohort of five to six members follows a nine-week syllabus that takes them from someone with minimal red team experience to autonomous operations. Guerrilla Operators will have a regular cadence of operations, which will require deconfliction from the parent red team to only ensure there are no safety concerns with the proposed target.

Expected outcomes for the nine-week cohort are as follows:

- Guerrilla operators are armed with the skills to continue their red team learning, as well as a support network for challenging tasks

- The parent red team has an expanded network of internal, trusted, and ethical hackers. This strengthens idea generation for campaigns and enables communication through the use of a shared and common technical language. Over time, the Guerrilla Red Team provides a steady flow of trained homegrown red team operators or security analysts

- The company itself benefits by having security-focused mindsets placed throughout technical disciplines, resulting in staff that are poised to ward off attacks by thinking like an attacker, functioning similarly to security-focused Site Reliability Engineers (SRE)

- Provides the company with verification that their security program and infrastructure are as robust as stated through the use of decentralized, independent low-tier actors attacking the network: an Offsec Chaos Monkey

- Provides the guerrilla operators real world, hands-on experience in a career field that is hard to break into outside of the Federal pipeline

Context

The program started as a way to seed assets throughout the company, improving the reach of red team operators conducting security assessments and campaigns. What it evolved into, however, was a program that reached much deeper than technical competencies. Members of the program reported that the opportunity to learn offensive security skills in a trusted judgment-free environment presented them career options they thought impossible: it opened a door for them that they were unaware even existed. It became apparent that the program meant more to the students on a personal level than it did to the red team on an operational level. At the end of the cohort period, most guerrilla operators were conducting training operations on

[2] The amount of time the student can spend during their workweek on the program is negotiated with their respective manager prior to official acceptance into the program

their own, using their newly established mesh network to overcome technical obstacles. Contrast that to nine-weeks prior, where almost half of the students were unaware of how to start the Metasploit framework.

This white paper will detail the methodologies used to establish a Guerrilla Red Team, discuss how the cohort progressed from the start of the program to the finish, share lessons learned, and provide suggestions on how companies can start their guerrilla red team

Origination

The parent red team at the focus of this study operates in a very lean capacity. Members of this red team function as operators, analysts, and developers. Having capacity stretched thin began to take its toll on the mental capacity of the operators. Operations Security (OPSEC) started to take a back seat to execution, degrading the entire reason for having an advanced adversary as part of the security team. It is not an exaggeration to say that the red team lost some of its expert power within the business as a result of this overburden.

Seeking to attack this problem as they would a network, all solutions eventually lead to the same conclusion: the red team needed more people to provide, in programmatic terms, random number generation (RNG) for the brain. The red team did not necessarily seek qualified offensive security professionals, but those within the organization to talk to, share ideas, and provide sanity checking. Moreso, the red team needed those with at least some offensive security exposure to provide the correct type of brain RNG.

With the goal identified, designing the program for maximum success was the next step. Attracting the right kind of people, while reassuring their managers that those in the security team were not poaching their staff, was critical. It was also important to the red team that those selected to join the program felt excitement at the opportunity. The team opted to go for the "all expenses paid" route: whatever the final form of the program turned out to be would be 100% paid for by the cybersecurity team. Having funding for the program that came out of their budget added legitimacy to the initiative: the security team was willing to put their money on the line to go forward with this program.

Program Design

The red team selected a third-party platform to serve as the training lab for the cohort. Access to the paid version of the Hack The Box labs allowed the use of all retired machines: retired machines come with official walkthroughs to ensure that anyone who attempts to hack the particular host will be successful. The Hack the Box labs function as the practical application component of Guerrilla Red Team training.

Supplemental learning came from both required reading, and audio lectures in the form of Cybersecurity podcasts selected to enhance or supplement the intended learning for that week.

The cohort achieved primary learning through live lectures, discourse, group review of operational notes, and practical application.

Live Lectures

Lectures for the Guerrilla Red Team program focused on a specific topic each week, starting with essential learning and gradually increasing in complexity. The first lecture discusses operational notes (opnotes): how to take them, and why they are required. The red team provides students with opnote examples and a tool to generate new opnotes when swapping to a new Hack the Box (HtB) target.

Subsequent lectures review the objective and main points of learning from the previous week and then preview the learning objectives for the current week.

Practical Application

Guerrilla operators achieve the bulk of their learning through practical application via Hack the Box labs. Initial instruction focuses on general operating system knowledge. Each week, MITRE ATT&CK Framework concepts are introduced and reinforced. Finally, the last weeks of the program have students attack current live boxes without write-ups; students will use their mesh peer network to overcome live hosts.

Learning breakdown is as follows:

- Week 1: Intro to Red Teaming
    - Technical Focus: Windows OS
    - Podcast Lecture: Darknet Diaries - MS08-067
    - HtB Target: Legacy (Windows OS)
- Week 2: The "other" OS
    - Technical Focus: Linux OS
    - Podcast Lecture: Darknet Diaries - Just Visiting
    - HtB Target: Lame (Linux OS)
- Week 3: Password Cracking
    - Technical Focus: Password cracking concepts and tools
    - Podcast Lecture: Darknet Diaries - Rockyou
    - HtB Target: Active (Windows OS)
- Week 4: Windows Privesc
    - Technical Focus: Elevation of privileges on Windows
    - Podcast Lecture: Darknet Diaries - Shamoon
    - HtB Target: Optimum (Windows OS)
- Week 5: Linux Privesc
    - Technical Focus: Elevation of privileges on Linux
    - Podcast Lecture: Darknet Diaries - Mini Stories
    - HtB Target: Traceback (Linux OS)
- Weeks 6-9: Live Ops
    - Work Together: Succeed Individually
    - Tiered Ops

        - Complete a specified selection of easy hosts before moving on to medium level hosts
        - Complete all medium hosts to move to hard
    - Group OP
        - Red Team guides the cohort through an operation, with the cohort conducting all offensive actions. Red Team requests information from the students and navigates the group accordingly
    - Students utilize their peer network extensively to research technical obstacles and overcome them
        - Instills the moxie required to function as an effective offensive security engineer

Phase 1: Red Team Development Program

Application Process

With the initial academic groundwork for the program in place, it was time to move on to selecting candidates for the program. The red team was unaware of how popular the program would be and initially budgeted for two or three students max. A Google form was created to serve as both an indication of interest and application. The form requested information about the requestees manager, best contact information, and a single question that required an answer of unspecified length, "Tell us something cool. Take this question seriously. Details are good." An optional file upload added as well in case the applicants wanted to "show, not tell."

Red Team Development Program (Q2 2020)

Interested in Red Teaming? Looking for a way to break into the field? This is it!

What this is: 4 hours a week for 9 weeks where you get to talk shop with the red team, get feedback on performance, and information about tools

What this isn't: An internship

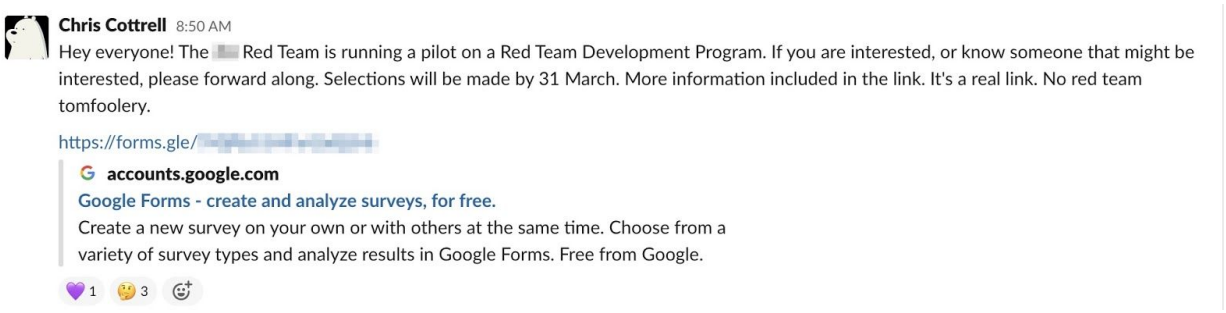
Please fill out the form below to be considered for the program. We have limited positions, and are trying to figure what our true capacity is. Things are definitely subject to change!

***Acceptance into the program is subject to your manager's approval to give us 4 hours of your time.
***Decisions will be made by March 31, 2020

The name, username and photo associated with your Google account will be recorded when you upload files and submit this form

* Required

The red team published one announcement on a shared Cybersecurity slack channel, briefly describing the program, and provided a link to the application. Interestingly, two of the selected candidates were not members of that channel, indicating that word of mouth spread the program beyond its initial scope.



The rationale behind an open-ended question is that the answer would decisively indicate who had the right frame of mind for the program. The deltas between the shortest application and the longest application were quite staggering: 15 words and 1453, respectively. It was quite easy to see what applicants were taking the program seriously and those that were not. The limited amount of time during the workday that the red team would have with the students required that those accepted to the program would have the highest levels of motivation.

In total, the program received 12 applications. Due to the number of quality applications, five candidates were selected with one alternate. All six applicants had submissions of 500 words or higher, and all focused on a technical topic.

Pre-Acceptance Process

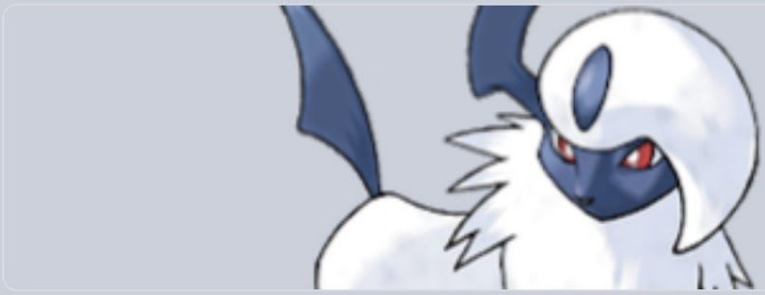
Before informing the applicants that they were accepted into the program, negotiations with their manager were conducted. The last thing we wanted to do was raise the spirits of applicants and deny them later because their parent teams could not afford to lose them. An email was sent to each of the managers of the applicants tentatively accepted to the program. The email discussed the program briefly and included an executive summary presentation. Principal due outs from this phase were approved by their manager to join the program, and an agreed-upon amount of time per week dedicated to the cohort. We asked for 4 hours per week, and cohort members received the requested time.

Once terms were negotiated with the managers, all applicants were informed of their status. Those that were accepted were sent a link to a "pre-flight checklist." Those that were either not accepted or were not allowed to attend were sent a nondescript email asking them to try again on the next cohort.

Selected applicants for the cohort were from the following technical verticals: Cybersecurity, Help Desk, Infrastructure.

Pre-Flight Checklist

An eight-question survey was sent to all accepted applicants to gather metrics and understanding of varying skill levels.



RTDP: ABSOL Cohort Pre-Flight Checklist

Congrats on being selected. The RTDP is using a Pokemon based theme for cohorts. Please fill out the following checklist.

Your email address [redacted] will be recorded when you submit this form. Not you? Switch account

* Required

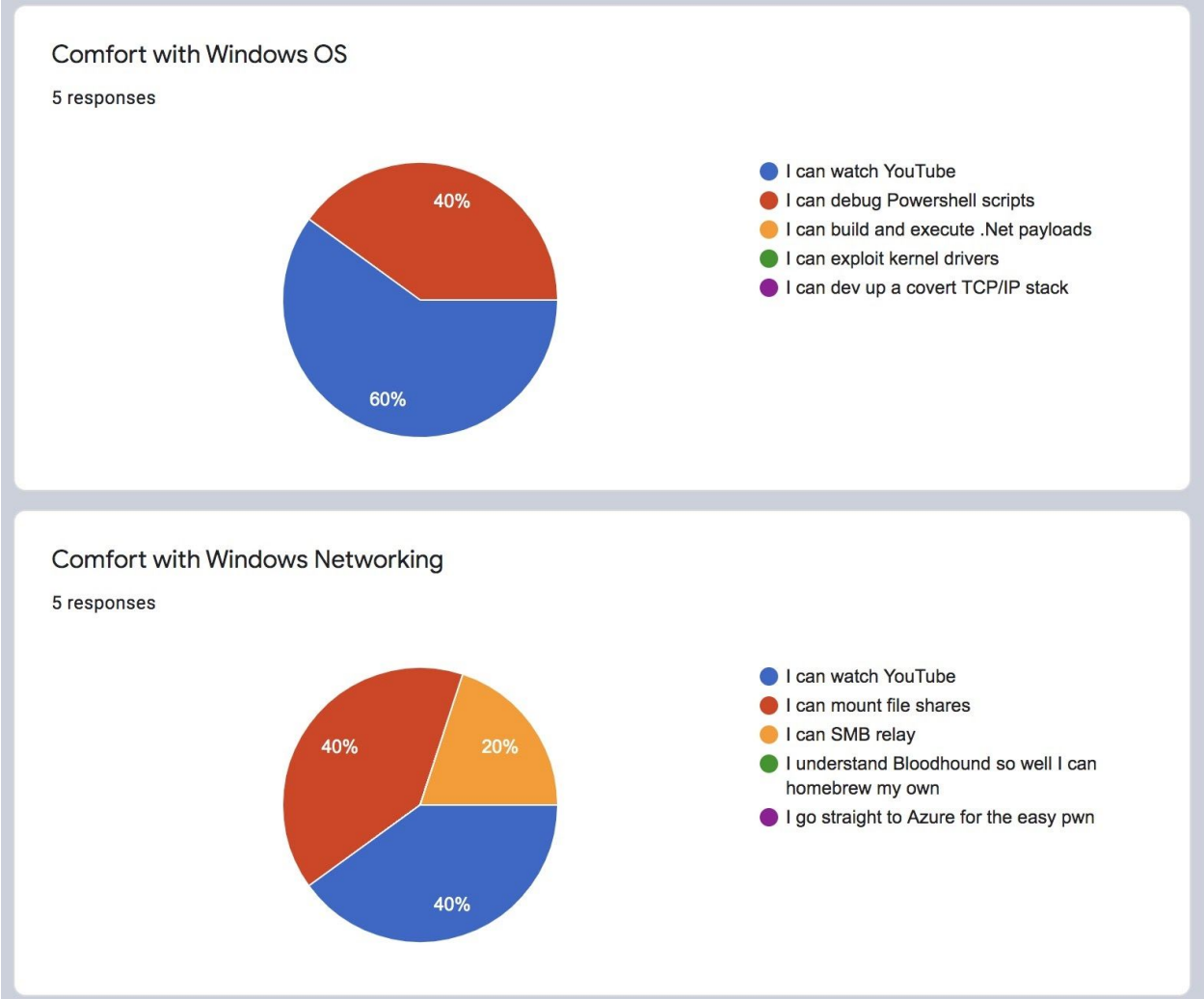
Which virtualization software do you have or can get? *

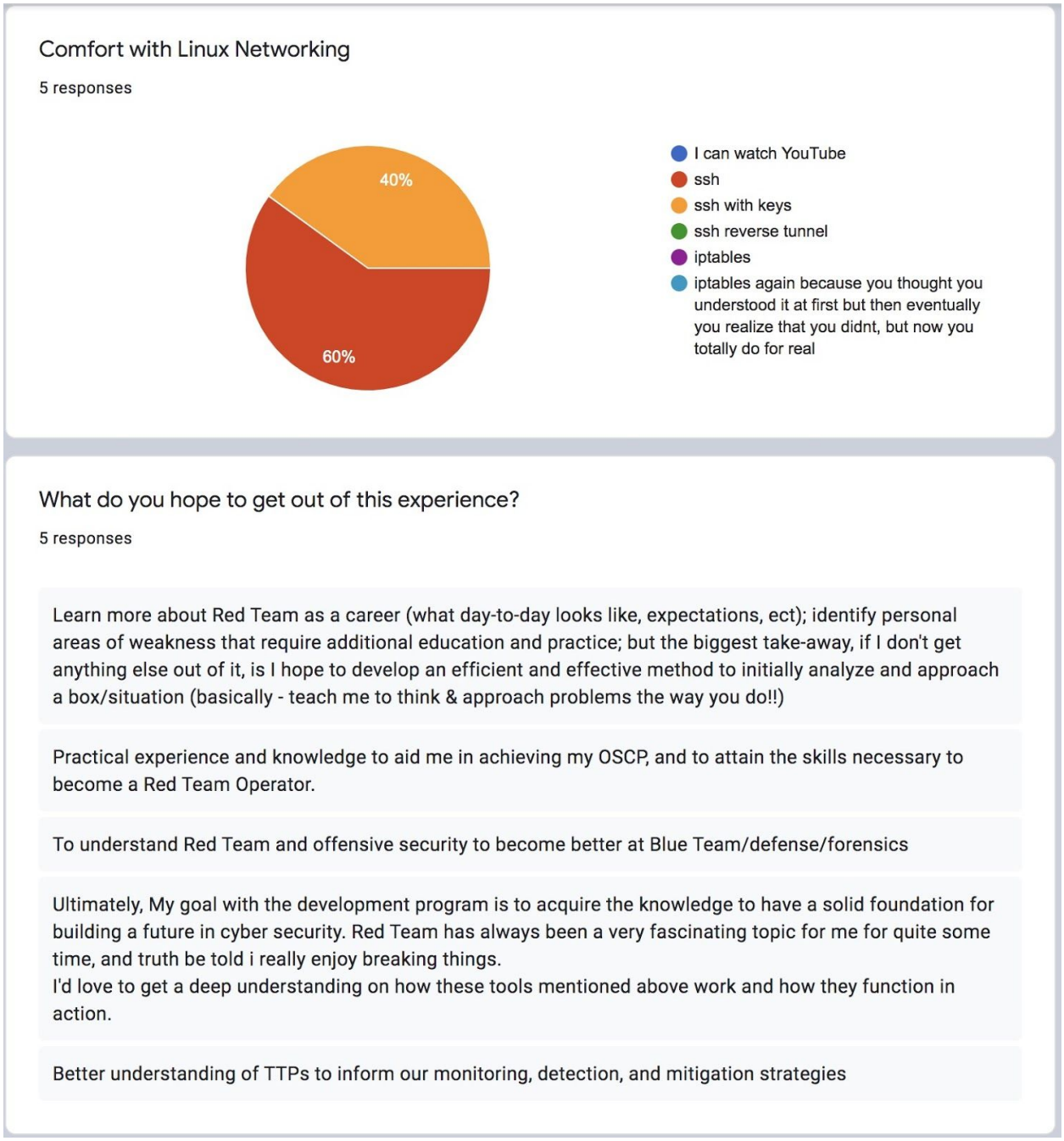
- [ ] VMWare (Fusion or Otherwise)
- [ ] Virtual Box
- [ ] Look man it aint gonna happen so you better set something up for me

What days during the week are best for you to have a solid block of time for instruction? *

- [ ] Monday
- [ ] Tuesday

Questions ranged from familiarity with specific operating systems to networking to expected outcomes of the course.





Program Start

"No plan survives first contact with the enemy" is a quote that best describes week 1 of the program. First lectures discussed what a red team was, what opnotes were, and why they were necessary. Unfortunately, we had severely underestimated how foreign a concept opnotes were to the non-red team world.

Opnotes: The First Roadblock

The forward momentum of the lectures hit an immediate standstill, preventing the hosting red team from moving on to the practical application portion of the program. Detailing what opnotes were and why they were important fell on deaf ears. Students had no concept of how the notes were used in real operations, and thus had numerous questions about the format, description level, and pieces of information to place in the notes. Instructors failed to provide sufficient examples of opnotes during the beginning portion of the program, which resulted in repeated opnote correction and adjustment throughout the first three weeks.

For the ABSOL cohort, the opnote standard that was taught followed that of the notes typically found in US Government red teams. The notes have a timestamped narrative of the operator's thoughts and actions, detailing specific commands when needed. The operator should, in practice, be able to hand their notes to another operator and have them replay the same operation based on the notes alone. Understanding what needs to go in the notes, and what would break up the flow and narrative takes time and practice.

The opnotes the students created during those initial weeks were more in line with raw logs than the narrative format taught. This miscommunication was the fault of the instructors, not the students. The red team created a python script to generate an opnote template based on a JSON file, removing at least some of the variation in the notes. Because the program's outcome was not to make fully trained offensive security engineers, leniency in the standards was allowed. It was much more important to get the students involved in the practical application portion of the course than to remain bogged down on trivial details. In the case of the opnotes, perfection was the enemy of good.

Lesson Learned: Opnotes

The students in the program had no real frame of reference as to what an offensive operation was, let alone what opnotes were. Instructors should provide multiple examples of opnotes used in their organization. Each learning section of the Guerrilla Red Team syllabus should have a corresponding opnote example.

Week 1 Practical Application: A Success

While the opnotes portion of the first lecture was a bit bumpy, the practical application portion went very well. An easy and very "lab-like" target was chosen to give the students a quick win. This target also had a corresponding podcast to help prime the brains of the student.

Lectures took up two hours of that first day, and the cohort broke for lunch to work on labs with the remaining hours.

Students reported that the podcast lecture on MS08-067, coupled with the Legacy lab box, went very well. Throughout that first week, students would message instructors with questions about tools or techniques needed to overcome Legacy. Being contacted outside of the specified training time was a good sign to the instructors that the individuals selected for the program had the right amount of passion and determination to do well as a guerrilla operator.

Deeper Look: Week 1 Questions

Here are a few examples of the questions received during the first week of the program:

- Friday April 17, The day of training, after training had ended
    - "Do I need to finish the (Hack the Box) starting point before I can use the VIP cert?"
    - "What was the Sublime Text package for timestamps?"
    - "Are there any good resources out there based around recon that I can look at on my own time?"
- Monday April 20, Three days after training
    - "I got to here (posted screenshot of mfsconsole). Either my exploit or payload isn't correct, but I think I am on the right path?"
    - "Am I missing something?" (Hours after the previous question when they tried it again)
    - "Okay it worked, you guys are awesome!" (re: the exploit after receiving help from the cohort that they were attacking the wrong network)
- Tuesday, April 21, Four days after training
    - "On top of getting root on Legacy, you wanted us to read chapter 1 of the textbook?"

While the nature of these questions pointed to a lack of structure in the program, it did show that well after the four hours of approved workday training, the students were still very much thinking of and executing on the course materials.

Lessons Learned: Week 1 Practical Application

The first week of the course went surprisingly well. All of the students completed the Legacy challenge, displaying that they had at least the capability to independently research attack paths and vulnerabilities.

This week's biggest hangup was ironing out the reasoning and methodology of the operational notes, or opnotes. Future iterations of the program will need to come prepared with a variety of completed opnotes. The GitHub project, Opnote Generator, will need additional demonstration to ensure the students have a solid template from which to work.

Weeks 2-4

Weeks two through four had their ups and downs. Students completed their assigned labs, but opnotes varied in unexpected ways. The instructors abandoned the original model of going over submitted opnotes in a classroom setting for a more 1-on-1 approach. Swapping to this model was done for several reasons, including:

- Wasting the student's limited dedicated lab time
- Some students requiring more attention than others (technical)
- Time zone differences
- Deciphering poor or cryptic opnotes

Overall, instructors noticed a significant change in the students' technical and research capabilities from the first week. Students were moving from labs that required off the shelf CVEs to finding vulnerabilities using tools such as Impacket.

Success: Week 2 Practical Application

The practical application for this week focused on an easy Linux machine: Lame. Students reported that they enjoyed the easy challenge and that it was nice to see a Linux host.

Instructors decided on Linux for this week's topic to expose students to a different operating system than Windows.

All students completed the challenge for the week.

Lessons Learned: Week 2 Practical Application

Instructors noted that issues with opnote format were still present. Additionally, taking the time to go over opnotes in a classroom format ate up significant amounts of lab time granted to the students. From this week forward, students placed their opnotes in a shared folder so that instructors could review the notes and provide direct feedback ahead of the scheduled classroom time.

Success: Week 3 Practical Application and Supplemental Materials

Password cracking was the focus of instruction for this week. Students reported that the lab machine for the week, Active, combined with the Darknet Diaries episode, Rockyou, meshed together well.

Instructors chose password cracking for this week's learning goal to expose students to generally what to do after a system is compromised.

All students completed the labs.

Lessons Learned: Week 3 Practical Application

The lab machine, Active, did include password cracking concepts. However, Active presented Microsoft Active Directory (AD) concepts more than password cracking concepts. Instructors added AD pentesting links to the syllabus for the week in order to keep students moving forward.

Future iterations of the program will include this host, but further along in the program when AD concepts are the focus. Revisions to the syllabus require an HTB lab machine focused more on password cracking.

Roadblock: Week 4 Practical Application

Week 4 is where the cohort hit its first real roadblock. The learning objectives for the week focused on Windows privilege escalation (privsec) concepts, resulting in leaps of logic for some students. Instructors did not properly screen for these issues.

Root cause analysis for the roadblocks shows that the students were not at fault for any confusion on the learning objectives. Some students had never seen concepts for these learning objectives before and did know how to begin the correct research. Indeed, it was a classic case of not knowing what they did not know.

Instructors provided privsec resources and articles for self-learning toward the end of the week as a result: students will be provided these resources at the start of the week during future iterations.

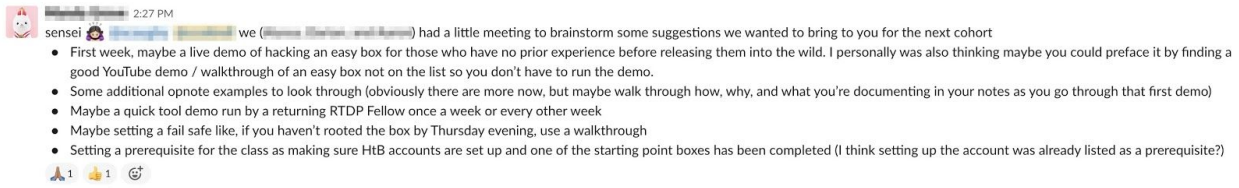
Success: Week 4 Cohort Mesh Network

Week 4's focus was Windows privesec. This topic proved difficult for students. The success for week 4 came in the form of the cohort peer mesh network becoming operational to discuss problems and present solutions.

The students had an independent meeting discussing the issues they were having for the week and then presented those solutions to the instructor. Among the main problems and deficiencies with the program that addressed were:

- Needing live demos of ops and tools.
    - Reading about tools and techniques is okay for conceptual knowledge, but seeing a quick demonstration cements learning
- Additional opnote examples
    - Students desired additional examples of opnotes to help guide what was important to capture for a narrative and was not
- Tool demonstrations
    - Students would like to see demonstrations of conventional attack tools on a recurring basis
- Educational failsafe
    - Students requested time-bound guardrails on objectives for the week. The example given was that if students had not captured the user flag for the assigned objective by Wednesday morning, instructors need to provide additional assistance to ensure that frustration does not impact learning
- Course Prerequisites
    - Students requested that HacktheBox accounts set up before the start of the course, specifically, completing the starting point labs

The fact that students felt comfortable enough to bring concerns to the instructors speaks volumes to the Guerrilla Red Team program. One of the core tenants was to foster a robust learning environment. However, another essential tenant was to form a trusted network of peers that included individuals of varying skill levels both up and down. Instructors hoped that by the end of the program, the students viewed them more as advanced peers than a student/teacher model. Empowering the students to reach out with concerns means that they would also reach out with questions they might find embarrassing due to skill levels. The program was designed to build people up without those barriers, not break them down later.


2:27 PM

sensei [redacted] we ([redacted]) had a little meeting to brainstorm some suggestions we wanted to bring to you for the next cohort

- First week, maybe a live demo of hacking an easy box for those who have no prior experience before releasing them into the wild. I personally was also thinking maybe you could preface it by finding a good YouTube demo / walkthrough of an easy box not on the list so you don't have to run the demo.
- Some additional opnote examples to look through (obviously there are more now, but maybe walk through how, why, and what you're documenting in your notes as you go through that first demo)
- Maybe a quick tool demo run by a returning RTDP Fellow once a week or every other week
- Maybe setting a fail safe like, if you haven't rooted the box by Thursday evening, use a walkthrough
- Setting a prerequisite for the class as making sure HtB accounts are set up and one of the starting point boxes has been completed (I think setting up the account was already listed as a prerequisite?)

🙏 1 👍 1

Week 5: The Turning Point

Week 5 was the turning point of the program because of a conversation between students and instructors at the end of the weekly instruction.

Coming off of the previous week, where the students recommended a slew of improvements, instructors were determined to align learning to student needs. The week's lesson plan was adjusted to meet a requirement from the previous week's suggestions: an operation conducted together as a cohort.

Sherpa Ops

Lesson plan adjustment for week 5 included a "live" operation. Students functioned as a group of operators against the same objective, and the primary instructor served as an advisor guiding the operation: a red team sherpa. The Hack the Box host, Jerry, was selected as the target due to its straightforward and logical attack path. Students worked together to attack Jerry, while the red team sherpa provided guidance and brainstorming. The intent for Sherpa Ops was to conduct an operation against a target the sherpa did not know, providing the students with a view into the mind of how an experienced operator would pursue a new target. The goal was not to fully root the target: the goal was to show how a professional would attack a target.

The sherpa provided critical guidance at times, but overall, the group used each other to overcome obstacles and trigger open-source research. It was indeed a fantastic sight to behold as each of the students reinforced each other.

As a group, the cohort rooted Jerry in one hour.

Success: After the Sherpa Ops

After the Jerry Sherpa Op finished, instructors asked the students what they thought of the program so far. It was about halfway through the nine weeks, and the red team had tried to incorporate some of the changes requested in the previous week. The answers received were resounding, unexpected, and genuinely touching.

- Students reported that their colleagues had begun asking where and when they could apply for the program because of how awesome it sounded and what they saw the students accomplishing
- Students reported that they had attempted to research similar programs to that of the Guerrilla Red Team and could not find anything: not a single program mimicked what they were experiencing
- Students reported that the opportunity to build relationships with someone willing to show them the ropes, provide answers to their "silly" questions, and show them that with

enough persistence they could succeed in the profession was invaluable and borderline life-changing

Receiving feedback of this nature was overwhelming. The course had evolved into something else, something far more significant than intended. It meant more to the students from a personal and professional level than what it meant to the red team on an operational level. Week 5 was the turning point for the program because this was the week in which the red team decided to bring Guerrilla Red Teaming to the world, serving as the genesis for this white paper.

Weeks 6-9: Live Ops

The final portion of the program focused on doing "live" operations on Hack the Box. Instead of focusing on retired machines with easily accessible walkthroughs, students attacked live machines that did not have write-ups. This phase intended to convey struggling with initial access and force the strengthening of the peer mesh network within the cohort.

The Live Ops phase was not an unstructured free-for-all but followed a tiered approach. Students had to root a certain amount of "easy" lab boxes before moving on to "medium" boxes from a host list of targets found in the syllabus. For the week, deliverables were the opnotes for the target machines, whether the box was rooted or not. It was up to the students to seek advice about overcoming roadblocks for each host during lab offtime hours.

```
2020-05-24 14:39:08 team member names to keep in mind from (http://10.10.10.175/about.html) for enumeration of user accounts:

    Fergus Smith
    Shaun Coins
    Sophie Driver
    Bowie Taylor
    Hugo Bear
    Steven Kerb

2020-06-14 spent about an hour reviewing where I left off on this machine
2020-06-14 also ran dirb against 10.10.10.175, but results were the same as previous scan
2020-06-15 23:00:00 researching xss attacks, web attacks used on forms, and sqlmap (tool for web recon), recon-ng, and looking for other possible web
attacks
2020-06-15 23:27:32 kali@kali : ~ $ sqlmap -o -u "http://10.10.10.175/contact.html" --forms

        [ERROR] all tested parameters do not appear to be injectable.

2020-06-15 23:28:02 researching attacks on AD, really got a lot from this talk https://www.youtube.com/watch?v=2Xfd962QfPs
2020-06-15 23:38:10 maybe Impacket will be a good tool to use?
2020-06-16 23:10:06 spent a long time reviewing each page's page source with Google Developer tools to see if I could find anything interesting
2020-06-16 23:15:23 ran an audit with Chrome > Inspect > Audit to try and find problems with the contact page http://10.10.10.175/contact.html
2020-06-16 23:18:44 saved the output as a .html file... nothing jumps out as interesting at first
2020-06-16 23:44:54 continued reviewing page source
2020-06-16 00:00:46 <input class="form-control" type="text" name="Name" placeholder="" required=""> == $0
is this a thing?
2020-06-16 00:00:46
https://stackoverflow.com/questions/36999739/what-does-0-double-equals-dollar-zero-mean-in-chrome-developer-tools

        == $0, It's the last selected DOM node index. Chrome assigns an index to each DOM node you select. So $0 will always point to the last node you
selected, while $1 will point to the node you selected before that. Think of it like a stack of most recently selected nodes.

2020-06-18 21:03:18 while reviewing all the pages again, I do see that all blog posts are made by user "admin"
2020-06-18 21:05:50 the blog page that I found not in the initial scrape, http://10.10.10.175/single.html#
2020-06-18 21:06:17 there's a "search" bar on here, whenever I enter anything get the following error

        405 - HTTP verb used to access this page is not allowed.
        The page you are looking for cannot be displayed because an invalid method (HTTP verb) was used to attempt access.

2020-06-18 21:08:08 same error when I try to submit a comment
2020-06-18 21:09:38 same error when trying to submit anything to the form for http://10.10.10.175/contact.html#
2020-06-18 21:12:54 https://help.catchsoftware.com/pages/viewpage.action?pageId=2031718
2020-06-18 21:12:58 seems to be an error on windows 2008 or win 2008 r2
2020-06-18 21:13:33 maybe this was all a red herring?
2020-06-18 21:16:32 researching ldap anonymous bind
2020-06-18 21:16:40 Anonymous binding is an LDAP server function. Anonymous binding allows a client to connect and search the directory (bind and
search) without logging in because binddn and bindpasswd are not needed.
```

Week 9 Opnote Example for HTB Target, Sauna: Significant improvements compared to week 1

Success: Live Ops

Some students took to the live ops more than others. The less structured environment of this phase of the program enforced the sink or swim mindset: most students swam, and one sank. The students that swam did so at a nice pace. By the end of the program, one student was working on the medium tier hosts, whereas most students were finishing up the last of the easy tier hosts. As a reminder, all of these students have very minimal hacking, red team, or offensive security experience at the program start.

The Guerrilla Red Team program is one of non-judgemental learning, and the student that sank was placed on a learning path that facilitated their current experience. The student was failing because of a lack of experience or exposure to the profession, not because of technical aptitude. The sherpa operator placed this student on a different learning path to reinforce success and keep forward momentum on the program.

Lessons Learned: Live Ops

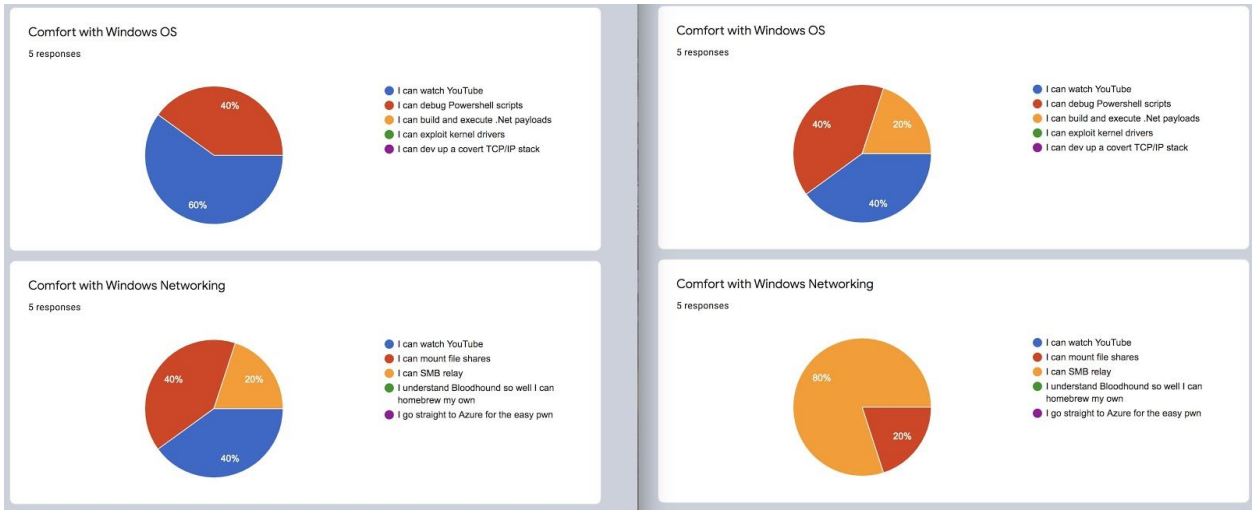
The Live Ops phase was challenging for the students, and it was done so for a reason. Instructors let one student struggle too long before placing them on a different, more structured learning path. While this phase is geared towards working together and succeeding individually, more regular pulse checks are needed in future iterations to ensure that no student succumbs to confusion from isolation.

Cohort End

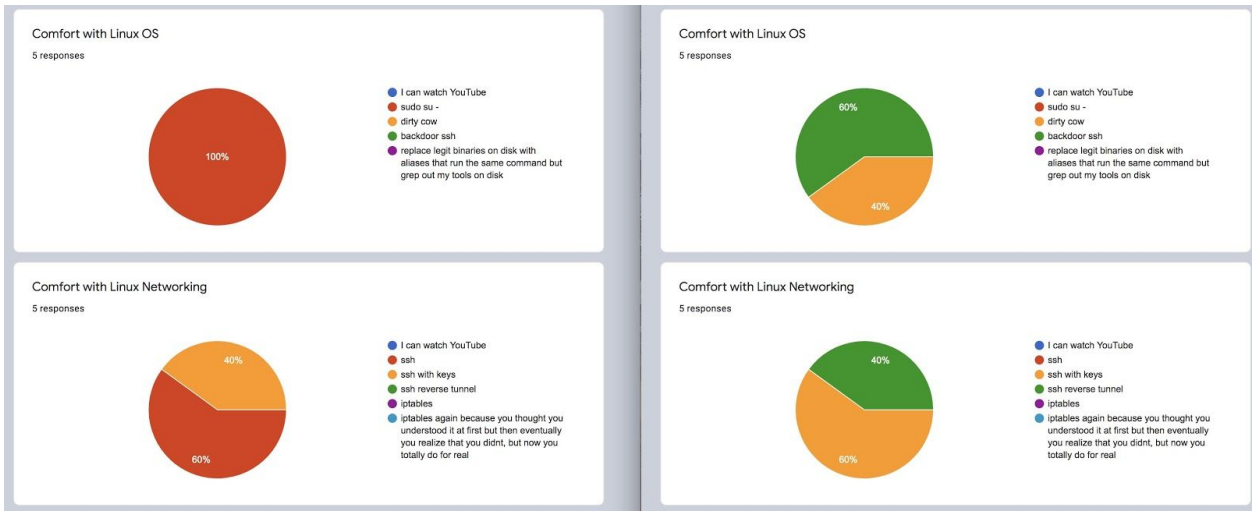
The final tally for live ops is as follows:

- All students completed at least one easy live box
- Three students completed all easy boxes in the syllabus (two students submitted completed opnotes on the final day)
- One student completed one medium level box and was close to completing the second box on the final day of the program

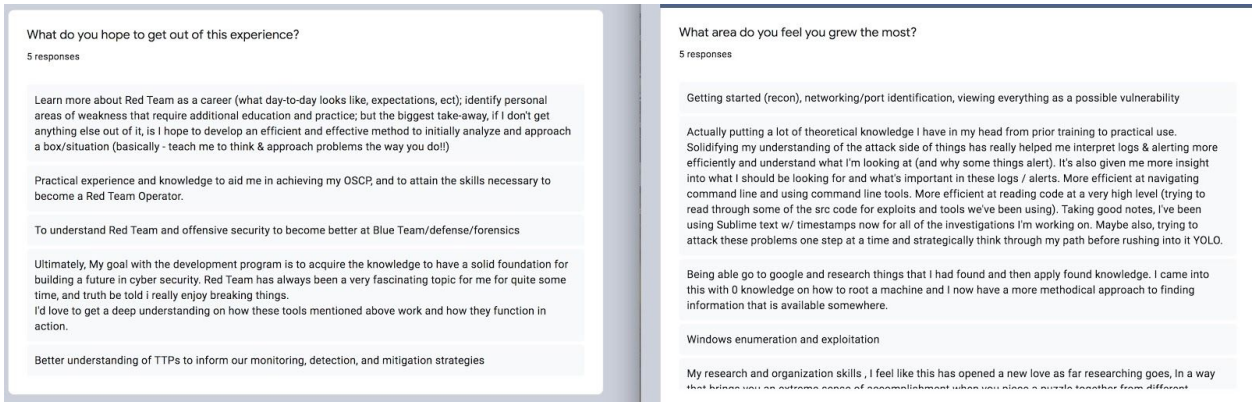
The original version of this program stated that after program completion, students could stay on as Fellows if they met a specific requirement: get and maintain Hacker rank on Hack the Box. Staying on as a Fellow meant retaining the benefits of having their Hack the Box subscription paid for with less stringent requirements. Upon taking exit surveys at the end of the cohort, every member said they planned on staying on as a Fellow. One student was even offered a permanent spot on the red team itself, a completely unintended result.



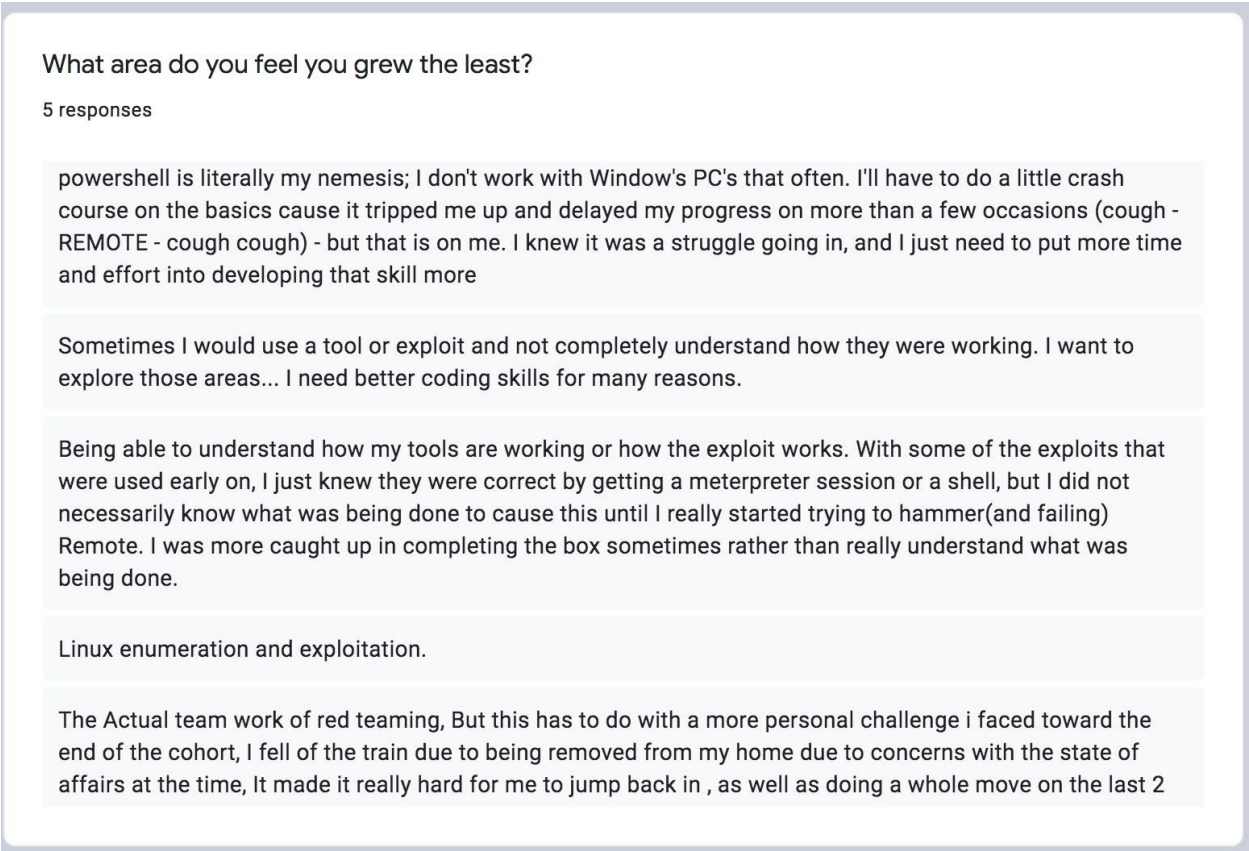
Comparison of Windows knowledge. Left, pre and right, post



Comparison of Linux knowledge. Left, Pre and Right, Post



Expected outcomes versus biggest gains. Left, Pre and Right, Post



Areas in which the students felt they grew the least during the nine-week program

Stakeholder Debrief

Once the cohort was finished, students were asked to complete an exit survey for the program, which included identical questions from the pre-flight surveys at the start of the course. When the data was gathered, a group meeting was scheduled with the direct managers of the cohort members, including any other high-level managers that could benefit from a debrief.

During the meeting, the red team instructors walked the managers through the development program, calling out specific outcomes or notable actions of their direct reports when applicable. Feedback received from the managers about their direct reports was very positive. There were some legitimate concerns and criticisms raised, however.

It was stated that some of the students had seen a dip in productivity of their daily requirements. They would go to bed early so that they could wake up early to work on the program, thus making it harder for them to focus on their assigned tasks. These criticisms also came with the caveat that the managers knew that this will make them better in the long run. From the red team side of the lens, a dip in productivity was to be realistically expected based on the amount of time spent working on the program. To have it called out specifically during the meeting meant that it needed to be corrected for the longevity of the program.

Another concern raised was that this program was a training pipeline for cybersecurity under the guise of a mentorship program. While this concern does raise some overall trust issues in the way the red team is viewed, it is a valid concern nonetheless. Stakeholders all commented similarly that they feel their direct reports would now more favor cybersecurity than other disciplines in the organization. It should be noted that all stakeholders felt the program was a great opportunity for their direct reports and expressed their genuine approval for the training conducted.

Lessons Learned: Stakeholder Debrief

Spending Too Much Time on the Program

If students spend too much time on the program during normal business hours or spend so much time that they lose focus on their daily operations, the red team should provide a proactive correction. Future iterations of the program will have language stated in the acceptance document that regular pulse checks with their managers will be conducted to gauge their productivity. If their manager feels that their productivity is not at a satisfactory level, the student's training in their cohort is to be temporarily paused, resuming once the deficiency is corrected.

Contact with the managers should be as passive as possible, such as through a ticketing system, automated polls via Slack, or scheduled emails going out at a specific cadence.

Training Pipeline Concerns

Stronger language needs to be placed in the acceptance document stating that this program is not a training pipeline. In the instance of the first cohort where one member of the program was placed onto the red team directly, that specific member was part of the security team already. To date, no members outside of the security team have been poached. Outreach to key stakeholders needs to be conducted on a regular basis.

Next Steps

As with many times during the pilot run of the Red Team Development Program, opportunities to do things better, smarter, or leaner presented themselves. Similar to when students viewed the program as a life-changing educational opportunity, the primary instructor also had a

revelation about continuing the program's evolution to enhance not only the student's careers but that of the security team as well.

Phase 2: Decentralize the Adversary

When finalizing the research for the first cohort of this program, it became apparent that the students fell into this gray area of adversarial skill. They indeed were not true beginners anymore, as made evident by the previous nine weeks of successful operations. However, they also could not be classified as full-blown red team operators either. Primarily, the program had created low-tier actors, and low-tier actors are assets when used appropriately.

Low-tier actors have their place in the security world. We always hear phrases like, "X product will defend against low-tier actors, but you need this more advanced product to defend against the REAL threats" or "yes, that is a great idea but will only work against low-tier actors." Okay, time to put that theory to the test. Even an immature security program should be able to handle a few low-tier actors, right? A loosely coupled idea began to form that could accomplish multiple goals at once if constructed correctly:

- The program graduates low-tier actors
- The real danger from a low-tier actor comes from a smash and grab mentality: low tradecraft could equal the destruction of a host
- Low-tier actors generally lack sophisticated tools and techniques
- The program attracted students that were looking for opportunities to gain exposure and experience to the field

Tying all of these thoughts together into a singular purpose, the program's real purpose became clear. Evolving from the original mindset of a single phase:

1. Upskill assets

There existed a real possibility of a second phase that answered the "so what" of the program, having a real impact not only on the business but also on the people:

1. Upskill assets
2. Unleash hell

Guerrillas in the Network

When breaking down the skills and attributes of graduates of the Red Team Development Program, it became apparent that they share similarities with those that practice unconventional warfare: guerrillas. Like guerrillas, the graduates are generally unsophisticated, underfunded, and underequipped. Guerrilla warfare is often effective, and the challenge here was to figure out how to apply that model to both red teaming and cybersecurity in general.

Throughout history, governments have had resounding success in the use of guerrillas and irregular warfare, typically providing the training and arms for the irregulars to conduct their operations. The mental model applied to this scenario follows that where professionals train and equip locals to fight on their own, similar to US Army Green Berets. The Red Team Dev Program provided the training. Following the Special Forces model meant that the red team needed to provide the weapons as well.

Guerrillas conduct operations independently of the main force (red team), with some slight safety checking to ensure they are not attacking a friendly or critical asset. The process of selecting targets without putting the business under any unnecessary risk needs to come from the red team in some capacity. The heavy-lift of constructing the armaments needed for a guerrilla operation comes from the red team, but luckily almost all of that process can be automated.

Program Overview

Phase 2 of the Guerrilla Red Team program focuses on decentralizing the adversary as much as possible. Decentralization can mean several things and will vary from organization to organization. For the examples listed in this whitepaper, decentralization has the two following definitions:

1. Decentralization of tradecraft
2. Decentralization of target selection

Decentralization of tradecraft means that with an ever-growing pool of low-tier assets, tactics, techniques, and procedures (TTP) will significantly vary with each operator. TTP variance improves the security team's detection rate and provides cover and concealment for the red team.

The decentralization of target selection refers to the irrational and irregular mechanisms in which a guerrilla chooses a target. A red team selects targets based on a campaign that supports a specific goal: there is a logic behind the targets engaged. Guerrilla operators will not share that same logic.

The idea behind decentralizing an adversary stems from that of Netflix's Chaos Monkey tool. The question that the Guerrilla Red Team attempts to answer is, "How do we make a Chaos Monkey for red teaming?"

The following steps briefly describe the lifecycle of a Guerrilla Red Team operation, and are detailed in their corresponding sections:

1. Target Selection
2. Target Approval
3. Arming Sequence
4. Scheduling
5. Arms Delivery
6. Op Execution
7. Detach Resources
8. Debrief
9. Blue Team Delivery

Target Selection

Target selection can come from one of two sources: from the guerrilla asset due to their knowledge of the environment, or a list produced from the red team but chosen by the asset. The organization's asset management suite or vulnerability scanning tools can produce a list of hostnames, operating systems, and a list of potential vulnerabilities depending on the Red Team posture and relationship with the organization. It is also very likely that the guerrilla operator will have intimate knowledge of inventory seen in day-to-day operations and may choose to attack a "local" device.

A critical component of this portion of the lifecycle is that the Red Team itself does not select the target; the guerrilla operator provides the target selection. This provides a level of ownership with the guerrilla asset's operation and removes overhead from the red team.

Target Approval

Once a target is selected, the guerrilla will write up an operation plan, or op plan. This plan will detail a general attack path and the intended goals of the attack on this target, which can be as

simple as achieving root or local administrator access. The completed op plan is submitted to the red team for safety checks, verifying that the target is not too critical to the business or too sensitive. An example of a sensitive target is their direct manager's host or the CEO’s laptop.

If the safety checks come back clear, the red team informs the guerrilla that their plan is approved and awaiting scheduling. Then, the red team begins to build the guerrilla's armaments so that they can schedule the op.

Arming Sequence

Arming the guerrilla involves multiple steps:

- obtaining credentials for the op
- creating and hosting an attack platform for the op
- creating ssh keys for the attack host
- securing the keys and credentials
- packaging everything up into a target package
- storing the target package securely for delivery

The examples in this whitepaper use Amazon's Simple Storage Service (S3).

Upon completion of all of the above steps, the red team will notify the guerrilla asset that it is time to schedule the operation. Details about each of the above steps are presented below.

Automation opportunities exist for many of the steps in this phase.

Credential Assignment

The red team will request a certain number of domain accounts to be used specifically for the Guerrilla Red Team. Ideally, these accounts should, at the minimum, have domain user permissions, but having random permissions assigned to each of the accounts is ideal and encompasses the spirit of decentralization. The permissions would not be known to the red team, and enumeration of account permissions serves as a training and tradecraft function for the guerrillas.

Each of the accounts is created and placed on inactive status until they are ready for immediate use.

Attack Platform

The red team constructs an EC2 host for the guerrilla to launch their attacks; this method works when the hosting environment is peered to the target network. If the organization is not peering or lacks peering into the environment, other options are:

- Set up a partnering agreement with a team that does peering. Having infrastructure as code that you can pass off to the team may expedite this process
- Have the guerrillas launch attacks from their own host, joined to a VPN. This is the less desirable solution, as the red team loses positive control over the situation. This risk may be acceptable to different organizations based on security and risk posture.

The operating systems for the attack platform can range anywhere from Kali, Linux, or Windows. It is up to the red team to decide what platform they would like to build for the given operation and forensics training initiatives for the blue team.

The newly constructed attack box is to have its security groups configured to not allow any access over SSH or other ports. When its corresponding operation begins, the box will have SSH or RDP whitelisted for the guerrilla's attacking IP.

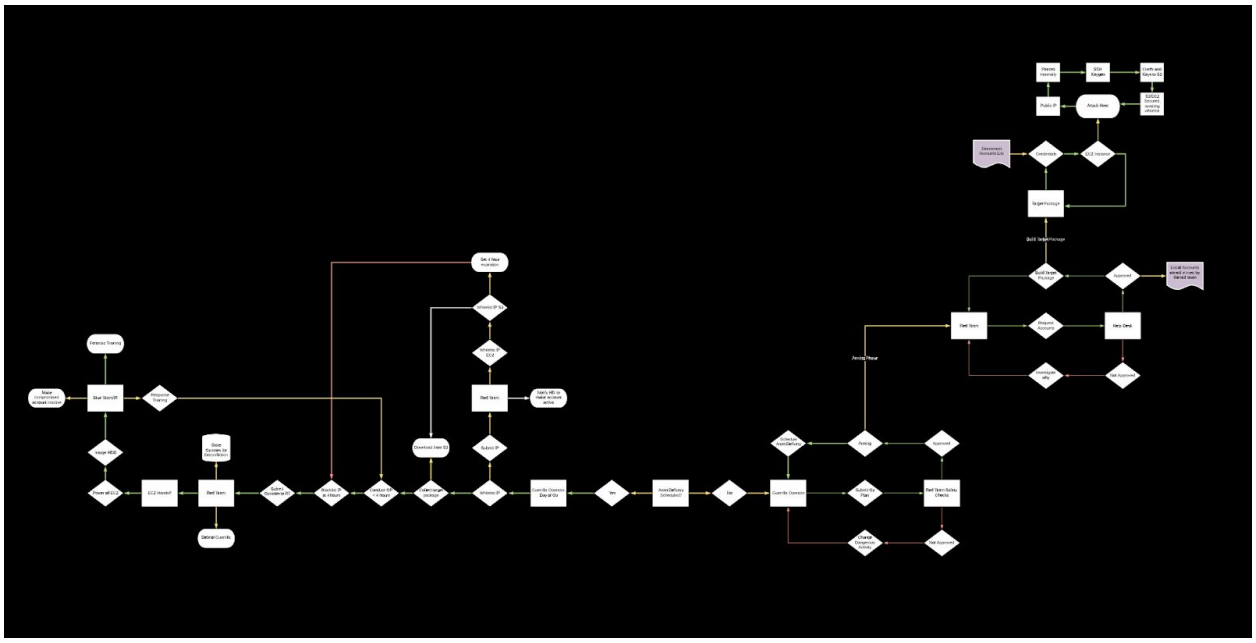
SSH Keys and Domain Credentials

It is recommended to set up SSH keys for each operation. Baking the key generation processes into the build pipeline expedites the arming phase. The private key is to be stored in the target package, which will be secured once the arming sequence is complete.

Domain credentials are also to be placed inside of the target package, preferably on the op plan itself. For the example in this whitepaper, the credentials and target IP address information were placed inside of a JSON file to be used with an opnote generator tool that produces a standard format used in the Red Team Development Program.

Storage and Delivery

The completed target package is to be stored somewhere securely, such as S3, with public access turned off. On the day of the operation, public access is to be enabled so that the guerrilla can pull down the target package. Storing the keys, credentials, and connection information in this manner places the onus and overhead on the guerrilla, freeing up the red team to observe the operation in action.



Multi-step flowchart on the guerrilla arming process

Scheduling

Once the target packet for the operation is secured, the red team will reach out to the guerrilla asset to schedule the operation. The rules surrounding the operation are simple: it cannot last more than four hours. The specific amount of time selected, four hours, was chosen based on environmental characteristics within the organization. Four hours is enough time to conduct an op, but not enough time to become over fatigued and make mistakes.

30 minutes prior to the start of the scheduled operation is when the next phase begins: Arms Delivery.

Arms Delivery

30 minutes prior to the start of the scheduled operation, the guerrilla will submit their public IP address to the red team. The red team will then whitelist the IP to the attack host, enable access to the S3 bucket containing the target package, and inform the account owner (usually Help Desk) to enable the selected domain account.

The red team informs the guerrilla of the S3 link to download their target package, which contains all relevant information to begin the attack.

Automation opportunities exist for much of this phase.

Op Execution

The guerrilla begins the operation by executing the opnote program located on their attack box. It will generate a plan for the operation and present them with credentials to be used. The guerrilla conducts the operation for no more than four hours, at which time their IP address will be removed from the security group allowing SSH or RDP.

During the operation, the red team lightly observes the engagement and stands ready to provide deconfliction support if and when the low-tier actors set off alarms.

The operation is to be conducted without notifying the blue team or any other security teams. Members of the white cell, or a similar group of key stakeholders, are to be advised of the start of the operation for deconfliction purposes.

Detach Resources

At the 4 hour mark from the start of the operation, the red team will remove the guerrilla's IP address from the EC2 security group for the attack host. Public access to the target package S3 bucket will be removed, if not already done so.

Automation opportunities exist for this phase.

Debrief

After the operation, the red team will debrief the guerrilla asset, and the guerrilla will debrief the red team. Op notes are a requirement, as the red team will be walking through the engagement step by step to ensure they have enough information to handle any deconflictions. Once enough data is gathered, the red team stores the opnotes in a repository. The guerrilla asset is released until the next cycle.

Blue Team Delivery

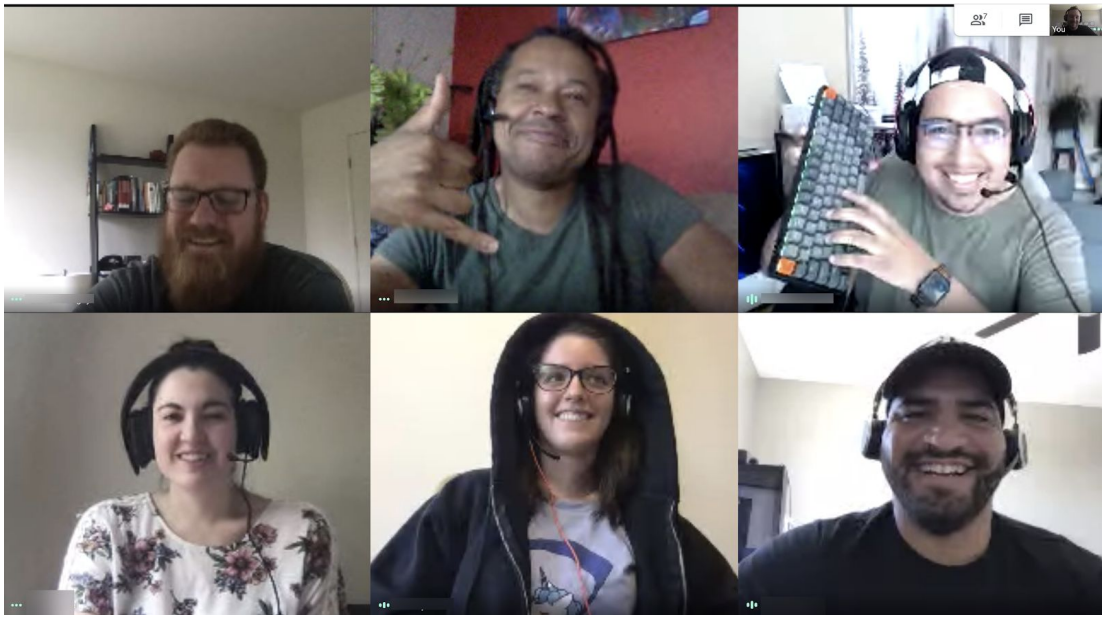
After the debrief, the red team shuts down the attack host and siphons off the hard disk image for the device. The image is delivered to the blue team for deconfliction and training purposes. Proposed training includes:

- Forensics training
- Responder training
- Threat training

Conclusion

Guerrilla Red Team started as a simple upskilling program and blossomed into something beautiful and unexpected. Originally dubbed “Red Team Development Program”, the guerrilla concept came after the first cohort had ended as a way to answer what was next: what was next for the program, what was next for the business, but most importantly, what was next for the students.

At multiple times throughout the execution of the course, students behaved and acted in ways that were surprising to the instructors, forcing the program to grow and adapt to meet their needs. What started out as a way for the red team to selfishly gain assets throughout the enterprise turned into an asset for people trying to better themselves. By providing upskilling opportunities through the nine-week training program, and then constructing a framework to allow graduates to hone their skills further, the business was impacted the most not through its processes or technology, but through its people.



ABSOL Cohort, Q2 2020

www.cybrary.it