SANS Institute
Information Security Reading Room

Capture the flag for education and mentoring

Jerome Radcliffe

Copyright SANS Institute 2021. Author Retains Full Rights.

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Capture the flag for education and mentoring: A case study on the use of competitive games in computer security training.

GCIH Gold Certification

Author: Jerome Radcliffe, jay.radcliffe@gmail.com

Adviser: Richard Wanner

Accepted: October 21st, 2007

Outline

1. Introduction

As a child I often played Capture the Flag, I have fond memories spending hours sneaking in the woods evading my would-be "captors" in search for the elusive enemy flag (Anonymous, 2007). I was recently able to relive that childhood excitement at a SANS conference during a day long "Capture the Flag" event played out over a computer network. The goal of this adult version of the game is to apply the teachings of a computer security course in order to penetrate different servers and gain access to files, analogous to those childhood enemy "flags." This modernized version of Capture the Flag was just as exhilarating as the game I played as a kid. The six hours I spent sneaking around networks and prodding at servers felt like mere minutes. Everyone in the class shared my excitement; the feeling was that this exercise was not only immensely fun, but that it was also beneficial. At the end of the class several people approached the instructor asking how they could run their own Capture the Flag course, or where to get more information on how to set a course themselves. It came as a disappointment to all that there was no easy answer for the students; there were no books or manuals on how to run such an event. When I returned to work, where I serve as a mentor and teacher, I thought this activity would be a useful tool

to advance the knowledge of my co-workers while also prompting team building. I took it upon myself to design two capture the flag "events" based upon my experiences at SANS. It is my goal with this paper to describe my experiences of setting up and running a Capture the Flag event, and, hopefully, help others utilize this as a security teaching tool.

2. Defining CTF

The idea of using games as teaching tools is nothing new; Romans were known to played games with their children to prepare them for future battle (Fowler 2006). Capture the Flag has a very similar history; it can be seen as a way of preparing for battle and learning strategy. In Capture the Flag there is an area to be infiltrated and objects to be captured, in the face of another team's defense (US Scouting Service Project, 2007). Playing this game as a kid there were pre-defined areas for each team to consider their "base" and a flag for each team that they had to protect located within their base. The objective was to sneak into the other team's base and capture their flag and return to your base without getting caught, while also protecting your flag from capture. Adapting this battle strategy game to the computer security field allows for creative adaptation, and the results can take a variety of forms. There are

three basic formats; “Offensive Only” in which each team attacks an established network, “Offensive/Defensive” in which a team either defends or attacks a network, and “Mixed Offensive/Defensive” in which teams are responsible for both defending and attacking a network. In all of these formats there is a moderator who is responsible for the setup, creation of the rules, and adjudication of the actual event.

In an “Offensive Only” format there are several teams trying to reach the same objective, or “flag.” The typical objective is to capture flags located on servers. In the simplest form the flags are simple text files placed in a specific location the player is required to locate. Once all these flags are collected they form a phrase, or a larger, single file. Both network and servers are setup by the moderator who is responsible for scoring, refereeing and presenting the solution of the game upon completion. Typically the moderator also will help teams in the process of the game, giving hints to help all the teams move forward in order to complete the game. This format is the easiest to score, as the moderator tracks the time required by each team to capture each flag. The first team to collect all the flags wins. This format has no defensive or monitoring elements.

In an "Offensive/Defensive" format there are two competing teams. One team's role is to be on the offensive; they would attempt to penetrate the other team's network, infiltrate their servers, and capture "flags." The other team's role is defensive; their job is to create the network, setup the servers for the offensive team to penetrate, and place the "flags" for the other team to locate, additionally they can setup monitoring devices, like burglar alarms, on the network to try and catch the other team in the act. Scoring this format is more complex. There can be points allocated for service availability (for example: keeping a web service up and running), and points for taking down the other team's services. Points can be assigned for catching certain types of attacks, and point can be deducted for getting caught attacking. This format usually has fixed time limits.

The last format is a combination of offensive and defensive elements, or "Mixed Offensive/Defensive" where both teams have to have an offensive element. In this format both teams penetrate another team's network and capture flags while simultaneously defending their own network. Each team is responsible for setting up a network and protecting the flags and monitoring for attacks from the other team. The scoring is very similar to the "Offensive/Defensive" format, where teams are only responsible for

one aspect of the challenge.

All of these formats need a moderator, or judge, who is responsible for creating the rules of the game and enforcement of those rules. Depending on the objectives of the game the moderator can also provide assistance to teams. The moderator is also responsible for verifying that teams meet individual goals and recording the time of completion for scoring purposes.

3. Creating CTF I

After attending the SANS conference, I decided that I wanted to create a Capture the Flag event of my own as an extra-curricular activity for the people at work. The division I work for has a focus on security; as a result I believed that there would be plenty of participants for the event. My managers were exceptionally supportive of this event and offered to provide prizes to the winning team and to purchase dinner for all participants as the event would take place after normal work hours. In order to plan the event I needed to take two aspects of the event into account; the procedural, or rules aspect, and the technical aspect.

Procedural Issues

The procedural and rules section was the first section that

needs to be focused on when planning a Capture the Flag event, as there are a number of little details that need to be taken into consideration. In the case of my work environment, the makeup of the teams was the top consideration as there was the need to give sufficient notice to all the potential participants, in order to create the appropriate number of teams with the right number of participations on each team. For our environment, I settled on two-person teams with a one computer per team limit. The reason for the one computer limit was to "force" the teams to interact, creating a true team environment, and not to have two people working separately on individual computers with no interaction. The next question to consider for my event was to establish a time limit for the event. I did not want the event to be an all-day event like the SANS Capture the Flag or the Def Con International event, as I had taken an informal survey of co-workers who felt strongly that a long (more then five hours) event would be too much. Some said that they felt intimidated, that it would be too technical for them. They also said that if the event was after work, they might not have the mental stamina to compete. I decided that the first Capture the Flag would have to be limited to three hours including time at the end that would allow for discussion of the solution. For my work environment, a three hour limit made for a good balance between being too short,

possibly preventing teams from finding a groove together, or too long, possibly preventing people from participating over concerns of time. The next consideration was selecting a format. In my environment the participants have limited experience in the offensive tactics of the Capture the Flag game but have extensive experience in defensive tactics (as we work for a internet security company). The logical choice was to have an offensive only format so our staff could work on improving the balance of their knowledge. Once I had all the elements determined, I distributed a flyer which outlined the basic concept of the Capture the Flag game including an explanation of the makeup of the teams and the date and time of the event. There was a strong overall interest in the game, and teams began to form.

Technical Creation

The technical details of the event are the second consideration when planning a Capture the Flag event. In my case, a limited timeframe was going to play a significant factor in the design of the game. If there was too many exploits used, then it would go over the time limit. Too few exploits would make for a boring game, which would make it hard to get people interested in the next game. A second factor I had to consider was the limited experience in the game and in offensive attacks. This limited experience might mean that an exercise designed to take 20 minutes might take an hour.

Given those factors, two types of attacks were selected: Password Cracking (Anonymous, 2007) and IIS Unicode Exploit (National Institute of Standards and Technology. 2007). These were selected based on several factors. Password Cracking tools like Crack (Muffett, 2001) or "John the Ripper" (Openwall.com, 2007) are effective at showing the weaknesses of passwords, which is useful to reinforce the need for password security in the workplace.

The IIS Unicode exploit (International Network Services, 2007) is a simple exploit that demonstrates a common theme in the world of exploits known as Directory Traversal. The Microsoft Windows 2000 series of operating systems shipped with a web server called "Internet Information Services" commonly known as IIS. In December of 2000, a CERT advisory was released that outlined a vulnerability in IIS that allowed unauthenticated use of the system which included access to data and the ability to run commands(National Institute of Standards and Technology, 2007). The key to understanding this vulnerability is awareness of two things: how directory traversal works and what make a Unicode character work. Directory Traversal involves the manipulation of a URL or specified location. As an example a specific location in a unix system might look like this: **/usr/local/bin/help.** Using directory traversal concepts you could write the same location in different ways (Imperva, 2007). An example of

that would be to use the "../" notation which signifies going backwards one directory. So one could write **/usr/local/bin/help** as **/usr/local/bin/../bin/help** or **/usr/local/../local/bin/help.** In most applications these "../" notations are filtered out so that they can not be used in a harmful way. This brings in the Unicode aspect to this vulnerability. In computer technology there needs to be representation of every type of character of every language. This generates multiple representations of the same characters. As an example the slash ("/") is represented by "%c0%af" in Unicode. So normally when a user types this as their URL into a web browser: "**http://192.168.1.1//scripts/../../winnt/system32/cmd.exe?/c+dir+c:**" the IIS server would recognize this and not process the request. This particular request would execute the cmd.exe with the command of 'dir c:\' which would print out a list of all the files and directories in the main directory of the server. If the user were to replace one of the "/" with the Unicode representation of the same character so it looks like:

"**http://192.168.1.1//scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:**" IIS would not recognize this as a problem and process the request. This would return the output of the "dir c:\" command to the user. The user could then modify the command to perform harmful tasks or access restricted data without authentication. The IIS Unicode exploit does

not need any special tools to demonstrate, just a web browser, which makes it ideal for the first exercise.

Password cracking would be the second technical concept used in this event. There are several myths in passwords that even technical professionals still carry which this exercise should address. The first is that there is some mathematical formula that one can use to "un-encrypt" a users password, the second is that passwords are not breakable. The concept of cracking a password is rather simple. When a user creates their password, the system encrypts that password and stores the encrypted version of that password. The next time the user attempts to authenticate, the system prompts them for their password and encrypts it in the same way, and then compares the stored encrypted version of the password and the newly encrypted password attempt. If those two pieces of encryption match, then the user is authenticated and allowed access. If they do not match, then the password was not the same, and the user is not allowed access. The concept of password cracking is dependant on the attacker getting access to the stored encrypted version of the passwords. At that point one can start encrypting words/strings/numbers/etc and then comparing the encrypted result against the captured stored encrypted passwords. Once a match is detected, then that user's password is revealed. On modern computer systems over 5000 guesses per second can

be achieved.

In addition to these two attack types the ability to map a network would be a key element of the game. Tools like "nmap" are used quite heavily in network mapping and reconnaissance (Insecure.org, 2007). All three of these have the ability to be utilized on a regular basis for future games, and within the larger workplace. One of the goals of this game is to build a set of skills to enhance individual's skills in the computer security field. The experienced gained with the use of Nmap and John the Ripper would be used extensively in the computer security field.

With the exploits and skills needed selected, a logical setup would be needed to determine how those concepts would be used. The approach used to make those determinations was to map out how teams would ideally complete the event. The first step that teams should take is some sort of reconnaissance of the network. They would be given a network segment that would be attackable, but not told where the hosts are on that network or what operating systems those host were running. This would provide them on opportunity to utilize the Nmap program to not only find the hosts, but to determine their operating system level. Once the hosts have been identified, teams would ideally move on to the next step of using the IIS Unicode exploit. This requires a windows machine running IIS that has not

been patched. Windows 2000 Server from 2002 running on VMware server was chosen to fill this role. A default installation was selected with the IIS/Webserver options selected. Successfully completing this task is deserving of a flag, so one was placed in the "C:\" directory labeled flag1.txt. Several methods could be used to obtain the flag, but for the documented solution the IIS Unicode exploit to run the "type" command to display the flag in the browser window.

The second task needed to utilize password cracking tools such as crack or John the Ripper. The first step in this process is obtaining the encrypted version of the password. This turned out to be quite complicated. The first attempt to do this was to have the students utilize the IIS Unicode exploit to run pwdump2, which would output the encrypted password of the user accounts on that machine. In theory this is quite easy, the reality was it was not. The first problem was getting the pwdump2 program onto the exploited device. While quite do-able, I felt it was a little too much for the first time around on this. I did make a note to myself that I would have a IIS Unicode file transfer section in an upcoming CTF. After getting the pwdump2 program on the device, I attempted to run it with the IIS Unicode exploit (Elcomsoft, 2007). This did not work. It turns out that the IIS user does not have enough privileges to run this program effectively. Elevating the IIS user's privileges was considered, but

another option was selected. During the SANS course there was an example situation given where a hacker stumbled upon the "Windows Recovery" disk on a production server. This of course can be used to acquire the encrypted password for user accounts. The instructor made a point to tell us that we should be careful what we leave on servers, individuals that exploit boxes tend to poke around the device to see what else they could find. Typically there are interesting and valuable data in places like temp directories. I decided that I would utilize this methodology to "plant" the encrypted password. I created a non-administrator account with an easily crack-able password on the IIS server used in the first exploit. Then the pwdump2 program was used to generate the encrypted passwords and planted in the "C:\temp" directory. Utilizing the IIS Unicode exploit teams can use the "dir" command and "type" commands to retrieve the encrypted passwords. Once the encrypted passwords are found teams can use the John program to crack that user's password. A second VMware server needs to be created to house the second flag. The operating system chosen was RedHat 7.3. This older OS has several existing vulnerabilities that can be used in future events, which will make creating those events easier. The installation of this OS is very barebones. The only service that needs to be running is SSH to allow remote logins to the server. For this event though

all services are turned off except for SSH. The next step is to create a user on the RedHat server with the same login and password as the crackable-non-administrator account that was created on the windows server. This account should be created with UserID 0, which is a root or superuser account. There are a couple reasons that this is done. First, this often occurs in the real world. A user might not have privileges on one set of systems, but has administrator or superuser privileges on a different set of systems. The other lesson here is that there can be multiple "root" accounts on unix based systems that are not labeled root. These can be intentionally created by an inexperienced system administrator, or it could be from malware (Anonymous, 2007) or a rootkit (Anonymous, 2007) that got on to the system. The second and final flag is placed on this server in the "/" directory. This flag can be directly accessed once the team tries to login to this server as the user that "cracked" user from the Windows server.

There is one more element that is needed to make the event more challenging. Another server is created to be a decoy. This server does not have any intentional vulnerabilities setup on it, and does not need to be exploited to capture any flags. This is fairly easy to setup, one of the two previously created VMware images can be copied and modified so that the flags are removed, IP addresses

changed and passwords changed. An optional step is to setup a service to make teams more confused. As an example a webservice can be started that just has one page that has a funny picture or taunting language to amuse the teams.

The last technical element that needs to be thought about is what the teams will need to function in the event. Rather then leaving this up to the teams to determine, there should be a recommended Knoppix (KNOPPER.net, 2007) version of Linux to use. This provides several benefits. First, having all the teams use the same distribution of knoppix makes it easier to help demonstrate the solutions. If all the teams were using different versions of tools then little things can be different and the solution designed might not work. The second advantage is that there is no possibility of tampering or changing team's personal computers. These types of events often use tools that are very powerful and can cause irreversible damage to data or functioning of computers. Using knoppix limits that possibility. Finally, teams will get some experience using knoppix, a tool that can serve them well in their careers. The distribution that was chosen to suggest to teams was one called "Backtrack 2" (Remote-Exploit.org, 2007). This distribution is heavily focused on forensics and security, which for our environment is very applicable for future use. Also this

distribution worked on the laptops that our employee's have. This distribution was downloaded as an ISO image and placed on our local network for easy access. One other element that was suggested to the teams is to acquire a USB-Flash drive. This allows teams to have an easy place to write data, and provides the ability for students to bring reference material with them to the event. No internet access should be available during the event. This is to contain the exceptionally powerful tools that will be used in the events. There is potential for mis-typing an IP address and attacking a network that is owned by the public.

This information comprises the entirety of a Capture the Flag event. In summary, there are three servers: one Windows server with IIS installed and running, one RedHat server that is locked down to just SSH and has an additional root/superuser account, and one decoy server.

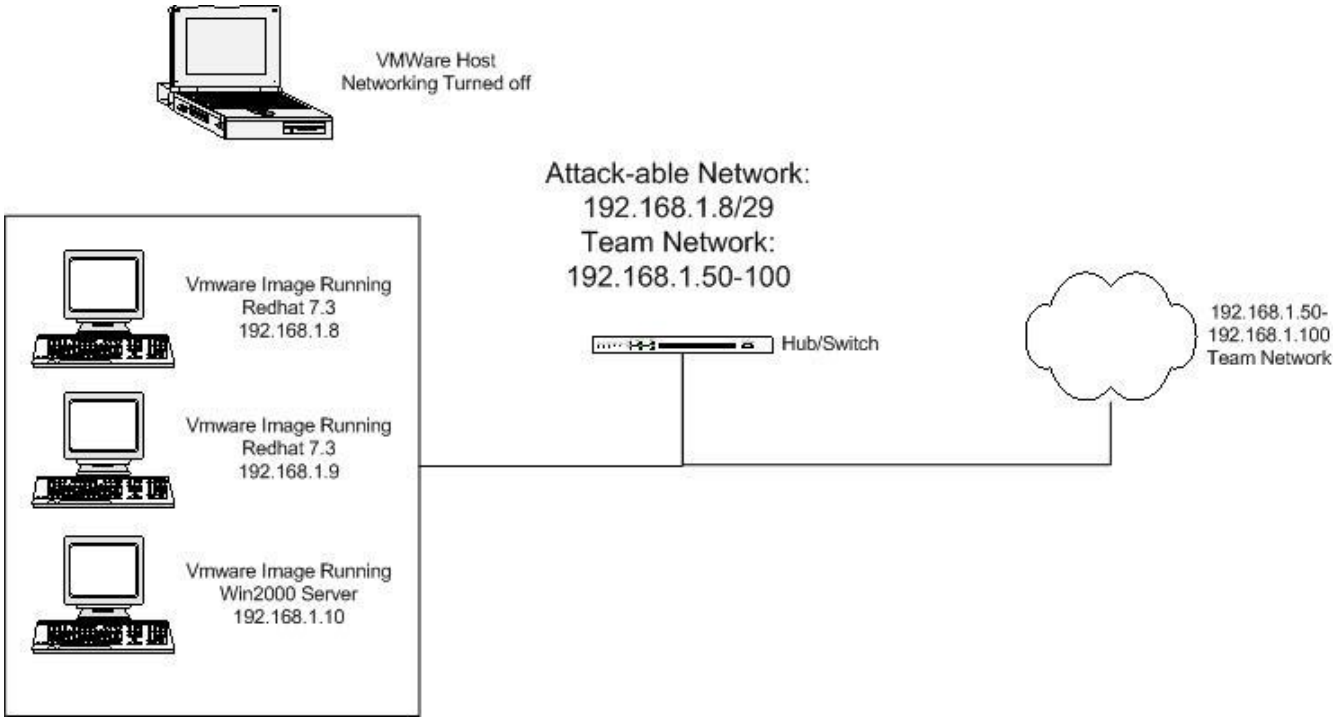
4. The Day of the event

Planning out the first Capture the Flag event was quite a daunting task. There are many indirect details that need to be taken care of before the event begins. The very first thing that needed to be done is to pick a date to have the event which can really only be done by asking your participants what they would prefer. Once that is

determined, a facility needs to be secured for that date and time. The facility needs to have adequate amount of power outlets and enough seating and tablespace for all the teams. Conference rooms are good candidates for this type of activity. The "flyer" for the event should be distributed about 30 days before the event, and should be posted in common locations to alert people to the event. The flyer that was used in our work event can be found in the Reference section. Once the flyers were distributed, our people started to get really excited. They posted the flyer in the break room, the bathroom, and basically anywhere they were allowed. Teams formed and started "playing" with the recommended Knoppix distribution to get familiar with how it functioned.

The day of the event, I loaded up a Laptop with the VMware images (IBM Thinkpad X31, 1.3gHz, 2gb RAM) and allocated the three servers with the IP addresses of 192.168.1.8-10 with a netmask of 255.255.255.0. Since there is no other network involved there is no default gateway or router needed to be setup. Teams are allocated IP addresses starting at 192.168.1.50, with the same 255.255.255.0 netmask. After starting up all three servers I ran through the "solution" to make sure that each of the pieces worked. Nothing would be more embarrassing then having it not work on the day of the event. Once that all aspects were completely verified I let teams

come into the room to set up. A network hub or switch was needed to allow everyone to be connected. Once the first team enters the room, I disconnected the network cable from the laptop housing the VMware servers to keep teams from getting an illegal head start. While the teams are setting up, the "attackable" network is written on a whiteboard, 192.168.1.8/29, along with each team name and time columns for each "flag", which allows everyone to keep track of how teams are progressing. Once all the teams arrive and are on the network, the event can begin.



The time for all the teams to complete the event took a little over 2 hours. The first team took slightly over 45 minutes to

capture both flags. During the event I wondered around the room and watched each team work, providing slight nudges or encouragement where needed. As teams finished I asked them to help the lesser experienced teams that were having trouble. This worked out really well, as it amplified the teamwork aspect of the event. One of the teams struggled due to "over thinking" their laptop situation. They had VMWare installed so they could use both BackTrack 2 and windows at the same time. They spent the first 30 minutes trying to get VMware networking to work properly. The prizes offered were $25 Best Buy gift cards for each of the winning team members. Teams were also offered dinner after the event dinner for all participating teams.

5. Summary

Overall the Capture the Flag event went far better than expected. There were no technical hang-ups, and everyone enjoyed the event. Managers of the teams were very impressed with the teamwork that occurred and encouraged me to make this a regular event. I got similar feedback from the participants as well. They were so excited when they got access through the exploit. There was quite a long discussion of how enlightening it was to see the "other side" of the security field. There were some suggestions and modifications offered to make things better for the next event. The first

suggestion was to make some additional flags for "advanced" teams. Due to a mixed environment of experience, some teams might finish quickly, while some are likely to go slower due to inexperience. This mixed environment issue could be addressed through some "bonus" flags which teams can work on after completing the primary, main, tasks. Such a suggestion could also be accomplished by creating two divisions, one that is easier for inexperienced teams, and one that is harder for advanced teams. Another concern brought up was that one team could become dominate due to experience and always win. A suggested remedy to this was to have the winning team help create/design the next Capture the Flag event and that would exclude them from the next event. This was very well received by both the experienced and inexperienced teams.

After receiving recommendations for changes in the event, several other ideas were discussed among our workers. First, there was interest in competing in larger, more competitive CTF events, like DEFCON in Las Vegas each summer. One idea was to use these smaller, work based events are preparation for creation of a future team set to compete in those larger CTF events. Another idea was to attempt an Offensive and Defensive format using larger teams. This would allow the mixing of a very high level of expertise in defense and the newly discovered knowledge on the offensive side. Our

company has several locations and one idea was to have the different location compete against each other. Another element that individuals were interested in was the addition of security products like Intrusion Detection Sensors on the attack network so that everyone could review the information provided by the sensor while the attacks were occurring.

The use of competitive gaming, such as capture the flag, in computer security can be successful in getting a greater level of participation, and enjoyment, from employees and students. This type of educational experience is hard to replicate in any other format and provides a unique experience for everyone involved. Using the knowledge presented here, others can hopefully develop and hone these types of exercises and offer them to even more individual in the computer field.

6. References

Anonymous. n.d. Password cracking. Retrieved October 15, 2007 from http://en.wikipedia.org/wiki/Password_cracking

Anonymous. n.d. Malware. Retrieved October 15, 2007, from http://en.wikipedia.org/wiki/Malware

Anonymous. n.d. Rootkit. Retrieved October 15, 2007, from http://en.wikipedia.org/wiki/Rootkit

Elcomsoft. n.d. Proactive System Password Recovery. Retrieved October 15, 2007, from http://www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003

Fowler, R. 2006. Gladiatorial training and combat. The Roman familia that travels together stays together. Retrieved October 15, 2007 from http://ancienthistory.suite101.com/article.cfm/gladiatorial_training_and_combat

Imperva. 2007. Directory Traversa. Retrieved October 15, 2007 from http://www.imperva.com/application_defense_center/glossary/directory_traversal.html

Insecure.org. n.d. NMAP-Free Security Scanner. Retrieved October 15, 2007 from http://insecure.org/nmap/

International Network Services. 2001. IIS Unicode exploit. Retrieved October 15, 2007 from http://www.ins.com/WorkArea/showcontent.aspx?id=1126

KNOPPER.NET. n.d. KNOPPIX Linux Live CD. Retrieved October 15, 2007, found http://www.knoppix.org/

Muffett, A. 2001. FAQ for Crack v5.0a. Retrieved October 15, 2007 from http://www.crypticide.com/alecm/security/c50-faq.html

National Institute of Standards and Technology. n.d. Vulnerability Summary CVE-2000-0884. National Vulnerability Database, retrieved October 15, 2007 from http://nvd.nist.gov/nvd.cfm?cvename=CVE-2000-0884

Openwall.com. nd. John the Ripper password cracker. Retrieved October 15, 2007 from http://www.openwall.com/john/

Remote-Exploit.org. n.d. BackTrack. Retrieved October 15, 2007, found http://www.remote-exploit.org/backtrack.html

US Scouting Service Project. n.d. Capture the Flag rules. Retrieved October 15, 2007 from http://www.usscouts.org/usscouts/games/game_cf.asp