DES - Security Through Obscurity Final Writeup

Brandon Crane, Monica Singh, George Wood

10 December 2017

1 Plaintext

The plaintext for the project was 'security' in ASCII characters, or '7365637572697479' in hexadecimal.

2 Ciphertext

The ciphertext for this project was 'EE4934F25C1D28F2' in hexadecimal.

3 Found Key 'K'

Key has not been found as of 12/10/17 at 3:00PM. Several difficulties contributed to this, which we will detail in later sections of this paper. But we are going to keep our instances running until Tuesday, at the time of presentation.

4 Time Taken

Our instances were launched on 11/27 at 11:15PM, and the programs began executing one half hour later at 11:45PM. So, as of 12/10/17 at 3:00PM, we have been executing for roughly one week, five days, and sixteen hours.

5 Challenges Faced & Solutions Adopted

5.1 AWS & DigitalOcean

The one major challenge that we faced was getting our program to run on virtual servers. We first tried to use the service DigitalOcean, because it offered $50 in student credit if you have a GitHub for Education account. It also offers very powerful virtual machines with multiple CPUs that would be of no cost to us if we stayed under $50. However, this platform does not enforce strong default security practices, and as a result, we created many instances without SSH keys. Some of these got hacked as a result of this, in addition to weak

passwords. The attackers used the instances to launch Denial of Service Attacks which were detected by DigitalOcean, causing them to lock our instances. After we figured out how to add SSH keys to our instances, we created more instances and launched our program again. About a day after launch, a couple of our accounts got locked out with this message emailed to us.

> Hi there,
>
> We reviewed the account and found it matches unusual patterns associated with violations of our Terms of Service and Acceptable Use Policy. While we cannot provide details about the specific flag(s) observed, we have determined that restoring access to this account is not possible. Apologies for any inconvenience caused.
>
> Best,
>
> Trust & Safety DigitalOcean

After this happened, we quickly created AWS accounts and ran our program on these. The program has been running (around 12 times slower than the DigitalOcean instances) ever since.

5.2 Team Member Loss

Fairly early on into the project, we lost two of our group members due to them dropping the class. Initially, this concerned our group greatly, as we had two fewer machines to run our DES crack on, and two fewer participants in brainstorming, coding, and presentation creation. We had to ensure that each of our team members were working efficiently in order to complete our assignments on time, and to know that we were making up for the fewer eyes looking at a problem at any given time. However, in the end each of us viewed our smaller team size as a benefit rather than a deficit. Scheduling meetings became simplified, dealing with conflicting opinions became streamlined, and becoming familiar with each member's style of working happened much more quickly. We all found that we were able to work very effectively together, and even enjoyed many of our meetings as a result.

5.3 Operating System Differences

Each of our three members had one of the three major operating systems. Brandon had macOS, Monica had Windows 10, and George had Ubuntu 16.04. The issues we struggled with included, but were not limited to:

- macOS
  - In the pthread library, an incredibly useful tool is the pthread barrier. However, the macOS version of this library does not work properly with the pthread barrier. As a result, some research had to be done to find a fix for this issue, which ended up being including a header file called "pthread_barrier.h" to remedy the problem. This header only

affected machines running macOS, and had no effect on machines running other operating systems.

- Windows 10
  - Even with a port of the Unix GCC compiler installed (MinGW), Windows does not have access to the include files & libraries that allow for TCP connections. These files include "socket.h", "inet.h" and "in.h", amongst others. These files are crucial for TCP connections, so Monica had to utilize virtual machines, and eventually partition her hard drive to include an Ubuntu instance on her laptop in order to compile and run our program.
  - MinGW also does not, by default, include the pthread library. There are workarounds to this, but none of them proved effective for our purposes, so we elected to utilize Ubuntu as mentioned above.
- Ubuntu 16.04
  - There were overall very few issues with Linux, and the issues that did arise were compatibility issues with other operating systems. Care had to be taken to ensure that any tools that we used with different versions for our different operating systems were behaving as expected.

6 Learning Experiences

The irony of having our instances hacked during a security project has not been lost on us. It very quickly taught us that if you possess resources that could be useful to someone, even if you're just using it for a school project, proper security measures are essential to ensure that no external parties can take advantage of those resources. In our case, our DigitalOcean servers were useful to some unethical hackers, and DigitalOcean's default security measures are much less than those of AWS, so they exploited the weaknesses that were left open to launch attacks from our instances.

Another essential lesson we learned is that ensuring strict adherence to proper implementation is critical. Originally, our DES cipher was able to encrypt plaintext and decrypt the output ciphertext back to the original plaintext, but the output ciphertext was not identical to other DES ciphers we found, which all had identical output to each other. This meant that, if we tried to start executing our DES crack without fixing that, we would likely eventually find *some* key, but it would not be the key that was actually used to encrypt the message – it would be the key that *our* implementation would have used to get the given ciphertext from the provided plaintext. As a result of this, we had to spend considerable time finding where our program was encrypting differently – in our case, it was one row of an S-box that wasn't exactly correct – in order to make sure the key we found was the key used by a completely properly implemented DES cipher.

Lastly, we learned that unix-like operating systems are generally easier for collaborative work. If all team members are working on machines that are running an operating system like this, distributing work amongst all members becomes much easier.

7 Team Dynamics

Due to the fact that we had a smaller team, we were able to coordinate and work together more easily. The majority of the work was done cooperatively while we were together, and we rarely had to do any work individually. Any work that was done individually was agreed upon by the group, and was evenly divided. Initially, Brandon focused largely on threading, George focused on TCP connections, and Monica focused on compatibility issues, but towards the end of the project we transitioned from this in to a more cohesive effort to work on individual issues we encountered. Overall, our group worked very well together, and we believe this is evidenced in the quality of our code and the amount of time and effort spent to ensure proper execution – even despite the fact the actual key was never found.

8 Feedback about this class

This class has been very difficult and time-consuming, but extremely rewarding. However, it would be more rewarding if we found the key. We all prefer to have our finals more spread out, so having the final the week before other finals was very nice. However, it would have been nice to have this listed on the syllabus, so we could prepare for this in our schedules ahead of time.