Journey to Becoming a Hacker: From Zero to Cybersecurity Ninja

Yoshuam A. Alicea Casillas
Master in Computer Science
Advisor: Dr. Jeffrey Duffany
Electrical and Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract — *Time to time technology changes and the need for cyber security experts increases. While is true that lots of universities prepare students on this career, somehow students face trouble understanding how to acquire the necessary knowledge to perform well, once they land a job in cyber security field. Also, it seems that they need to separate the theory from practice to not only understand why things happens in this field but to know how they happened and the causes that made them. Is for this that we will walk you through the journey of how to start a training that will take you from zero knowledge in the topic to skillful hacker in a short period of time using knowing tools to find the knowledge and exposing yourself to competitive environments that will put in practice what you learned in your degree and what was self-taught.*

Key Terms — *Capture the Flag, CTF, Cyber Security, Cyber Training, Hacking.*

INTRODUCTION

When people think to start a cyber security career, they often expect that pursuing a degree will get them ready to the professional world by having the necessary skills that a cyber security expert needs to overcome any challenge that will present in their jobs. But the problem these newcomers face is that hackers out there have been doing their bad deeds for long time, and even worse they have been doing it in real world scenarios. Some of them are trained by illegal hacktivist organizations and they have all the time in the world to achieve their objectives and to gather the day to day knowledge to become even better. In summary, these bad guys have more experience than any newcomer in the cyber security field. There is no university out there and no certification that will put you under the fire that the company that you will work for will be having. This doesn't mean that your university path is wrong, it means that is unrealistic to be able to train someone at the extent of being competitive against a real threat in the work environment. While your degree and university program will help you grasp all the cyber security concepts and theories behind it, we need to go further and find the experiences that will simulate the cyber war battlefield that you will be exposed once to start in your cyber security career. This is because is not easy for any curriculum out there to cover all the attacks, vulnerabilities and exploitation techniques that exists. So, we need to do this part by ourselves.

The internet is filled with information and we can use it to learn about everything related to cyber security. Problem is since we are talking about cyber security the information may be censured or hard to find so you can actually learn how to exploit something. Is more probable that you will learn how to defend and harden your system instead of how you can penetrate it. There is no problem on knowing how to defend and harden a system, the problem is understanding from what you are safeguarding your system. Knowing how a system can be exploited helps on how you can safe guard it. Is for this that we will take you to a journey of how we became skillful in the cyber security field while going through our cyber security degree. This journey will go from zero knowledge in cyber security to super skilled cyber security ninja.

TRAINING

When you first start in this career you probably have some computer knowledges like programming languages, database management, web application development, network administration, etc. Maybe you know how to use encryption to protect your data, the reason why you would like to use a virtual private network (VPN), and some other reasons that will point you to keep your information safe. Basically, you learned through the time how to protect your information even though you are not sure of how it works and if it really is protecting your system and information the right way. You just trust that it is well done, and your system is protected because of faith on the anti-virus and the firewall software they sold you at X company.

There are many paths to take to start a cyber security training and we will show you the one we took to become skilled in short amount of time. Our first step was looking for a cyber security competition that will force us to go through the topics they cover in those competitions. At first, we did not know what a cyber security competition was and how to play it. After a short research we learned that these competitions are called Capture the Flag (CTF) and they are divided into two different styles [1]:

- **Jeopardy style:** In this kind of CTF you are given tasks by categories, think of it as problems to solve, where you need to exploit, find vulnerability, reverse engineer or find intelligence on the given task. They are divided in different categories like, Web application, password cracking, enumeration and exploitation, reverse engineering, computer forensics, open source intelligence, cryptography, steganography, wireless exploitation, network analysis and log analysis. The aim of this type of CTF is to teach the competitor to identify vulnerabilities, look up for information, learn about software mechanics, identify network traffic, secure web applications, understand passwords weaknesses and correct use of cryptography and finally reversing techniques. All of this are essential for an individual to learn to think “out of the box” like an attacker will do to achieve his goal.
- **Attack and Defense style:** In this type of CTF, a server is given to each team and they have to protect it from the other teams, while at the same time you try to hack into other team’s server. This style is the closest that gets you to a real-world experience. In this style of CTF you will have to play by role like you will do in an IT department.

Once we learned what CTF were and their types we research about what are the skills needed for it. When going through our findings we discovered that there was a common factor in all of the different topics covered in both styles of CTFs. This common factor was system administration skills.

System Administration

We started training ourselves in system administration stuff like understanding our operating systems (Windows, Linux, MacOS) and how to use tools that comes with those operating systems. We learned how to navigate through the command line and how to make powerful search, parsing of that search and learned how to use basic tools like cat, grep, find, strings, chmod, chown, netcat, ssh, openssl, cron, ps, cd, ls, man, vim, etc. The main reason of starting on system administration training was because you need to understand how to administrate a system, where to locate those system’s important files and their directory structure. Once you have this knowledge you will be able to look for password’s dump, where server’s software is located, how to add and remove privileges, opening and closing ports, connecting to another machine through secure tunnel and so on. With the previous mentioned knowledge, you can basically start looking into a system and navigate through it without any problem. You will know what to look up and possible locations to hide backdoors and exploits once you got into a system remotely. As you can see now you can actually start understanding how hackers may get in into your system because you know how to do it as well.

This first step in our training was important because in CTF competitions you may have to connect into a server remotely and exploit it. Also, you may have to run programs that you will need to analyze using system administration tools. If you don’t have these skills, you will experience struggle not only in CTF but in real life scenario as well. Another thing to mention is that we went straight for the command line and not software because mostly of the servers, if not all, does not have a Graphical User Interface (GUI) that you can use to move through it as you are used to. But after mastering the command line we actually tested tools with GUI that could perform same purpose as the command line tools even though we barely used them.

After training ourselves in system administration we were ready to move on into the next steps.

Web Application Exploitation

Our second step was understanding how web application works and understanding the browsers. While going through this training we came up with a web page called OverTheWire.org [2] that will go through baby steps on how to start exploiting web applications. It will also create awareness of best practices that you as a developer should follow in order to avoid being vulnerable. It also taught us how to use the browser's developer tools to debug, tamper data, access and change cookies, and inspect the source code of the web page.

We learned how to make requests to a web application using JQuery and the JavaScript console to send values that were valid and will allow us to change the behavior of the responses given from the server. We also learned how to use famous google hacks [3] to navigate the directory of the server if this one was not well protected in the server rules (ex. .htaccess file). We learned how to inject code to change the queries the web application will do to the database using sql injection attacks. Furthermore, we learned how to upload a backdoor using php script that will be read as a picture exploiting file uploads functionality on the server when this one was not well protected.

After beating each of the OverTheWire.org website challenges we felt pretty confident on how to approach a web site exploitation. Even though these skills still not at the level of a hacker, now that we have the knowledge coming up with new ways of attack and thinking "out of the box" becomes even easier which will boost up in skills every day.

Cryptography

Our third training was based on cryptography. We started by understanding really basic and very old cryptography ciphers and what their purpose was and the reason of their existence. After learning basic ciphers like Caesar, Vigenere, PlayFair, and Railfence ciphers [4] we not only understood why they were so vulnerable due to their simplicity and limits, but we learned how to brute force them to find the message as well as how to perform dictionary [5], known-plaintext [6] and frequency analysis [7] attacks to decipher the encrypted message. Once we understood the simple ciphers, we went for the complex ones like RC4, RSA, DES, AES and many others. For the DES we learned how to use differential cryptanalysis attacks [8] to break variants of DES with up to 15 rounds. Also, we learned how RSA works what makes it work, the factorization problem that it faces and how to exploit it when misused using Wieners' attack [9] or using factorDB.com [10] to find factors of its modulus when their prime numbers are too small. We developed tools to run crib drag attacks on One Time Pads when used multiple times with the same key. We also developed tools for brute force attacks and were able to make an RSA exploit tool to break any RSA that was wiener's vulnerable or had small modulus. After, this training we became really good at cryptography and learned how to make good use of the different encryption algorithms that exist out there and when to use them.

Reverse Engineering

Next stop was the reverse engineering training. This was by far the most difficult training to overcome because we needed to understand basics of assembly language at least at the level of reading it and knowing what is going on. Also, we learned how to ease the process of disassembly a binary with objdump, binary ninja [11] and IDA pro [12] tools. We learned how to identify a buffer overflow, what to do to make it happen and how to safeguard using good programming practices to avoid them. We found two website that will challenge us to get more skilled in this field called ringzer0ctf.com [13] and pwnable.kr [14]. In these websites you will get tested in a hardcore manner in reverse engineering techniques. We also found a library written in python called pwntools [15] that will help us to write shellcodes to exploit binaries once the vulnerability was identified.

We never became experts at this field of cyber security, but we understood the very basics of it. Having reverse engineering skills will allow us to take a malware for example and analyze it in order to find out how it works. Once you know how it works you will understand what vulnerability the malware is taking advantage of and how to protect your system with the knowledge gained from the analysis.

Network Analysis

The final part of the training and we would say the most important one was network analysis. We said is the most important one because this is the real war zone where everything happens. The network is something we need to understand in order to be able to detect from where the attack is coming and how it is being performed. With network analysis skills you can trace the packets being sent over a network in real time or by capturing network traffic that occurred in a specific time. For this training we learned how to use a tool called Wireshark [16]. This tool is so powerful that we could follow streams and analyze at molecular level each of the packets sent over the network. We learned about DNS, TCP, ICMP, HTTP protocols and how do they work. Using Wireshark, we could trace this kind of packets and identify the source and destination of a communication between two IP addresses, we could capture content sent over http protocol and many other powerful things. Also, we could inspect wireless traffic that came from a router and using another tool called aircrack-ng [17] we could break the WEP [18] passwords and WPA [19] through dictionary attack. Basically, this training allowed us to perform forensics in the network.

After going through all of this training that took us over 6 months to feel comfortable with each topic we started competing on the National Cyber League [20] which is a Capture the Flag competition that is very competitive and will put your brain to work in every challenge they give and in CTFTime.org [21] which is a website that hosts many CTF competitions during the year. These competitions are worldwide and very complex.

National Cyber League

The National Cyber League is a defensive and offensive puzzle-based, capture the flag style cybersecurity competition. Its virtual training grounds helps high school and college students prepare and test themselves against cybersecurity challenges that they will likely face in the workforce.

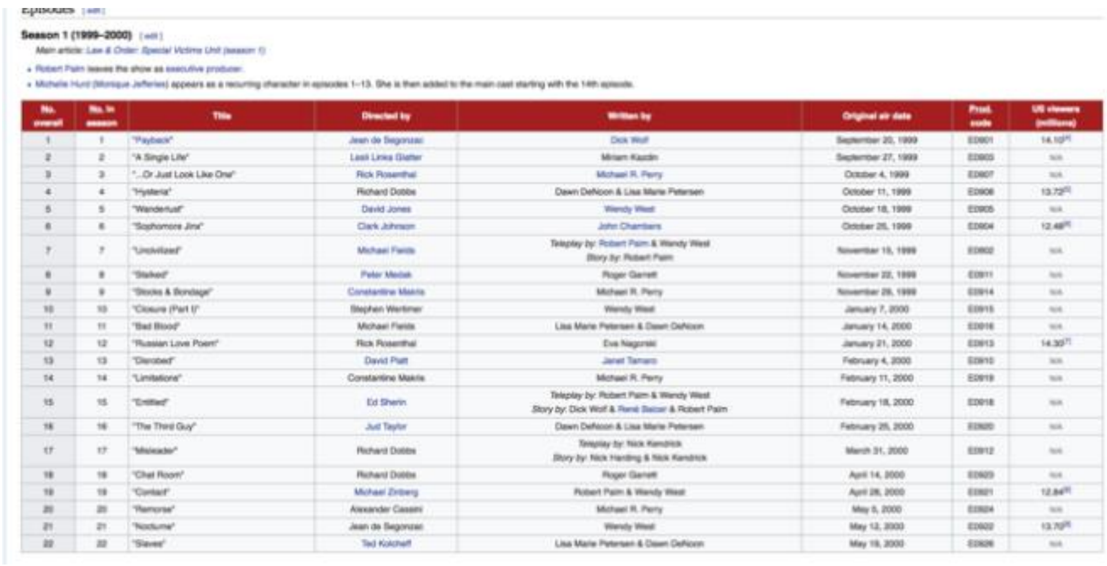
We competed in these competitions and we made it to the gold bracket each time. To understand the effectiveness of our training we did a study [22] along we several students that were starting in the cyber security field and the results were outstanding. In each participation we were able to get closer and closer to top 10 as individual and also as team.

From this competition we were able to learn about tools like nmap [23] for port scanning and dirbuster [24] for directory scanning. Also, for password cracking we learn how to use hashcat [25] and ophcrack to find the plaintext of a hash. We took it even further by writing our own scripts that will perform this kind of attacks to passwords. We also got better at writing scripts that will act as bots to solve problems with ease. Team work is also encouraged in this competition when it reaches the post season.

One good thing about NCL is that it measures the accuracy of your performance. This helps to make the competitor think twice and have a deep analysis on the problem to come up with the right solution else your accuracy will drop and your position in the leaderboard will be affected.

Work Done

We are going to share here some highlights of the work we did in NCL after our intensive training. We also going to include a high-level explanation of how each challenge was solved.



Episodes [edit]

Season 1 (1999–2000) [edit]

Main article: Law & Order: Special Victims Unit (season 1)

- Robert Palm leaves the show as executive producer.
- Michelle Hurd (Monique Jefferies) appears as a recurring character in episodes 1–13. She is then added to the main cast starting with the 14th episode.

No. overall	No. in season	Title	Directed by	Written by	Original air date	Prod. code	US viewers (millions)
1	1	"Payback"	Jean de Segonzac	Dick Wolf	September 20, 1999	E0801	14.10[4]
2	2	"A Single Life"	Lesli Linka Glatter	Miriam Kazdin	September 27, 1999	E0803	N/A
3	3	"...Or Just Look Like One"	Rick Rosenthal	Michael R. Perry	October 4, 1999	E0807	N/A
4	4	"Hysteria"	Richard Dobbs	Dawn DeNoon & Lisa Marie Petersen	October 11, 1999	E0806	13.72[5]
5	5	"Wanderlust"	David Jones	Wendy West	October 18, 1999	E0805	N/A
6	6	"Sophomore Jinx"	Clark Johnson	John Chambers	October 25, 1999	E0804	12.48[6]
7	7	"Uncivilized"	Michael Fields	*Teleplay by:* Robert Palm & Wendy West *Story by:* Robert Palm	November 15, 1999	E0802	N/A
8	8	"Stalked"	Peter Medak	Roger Garrett	November 22, 1999	E0811	N/A
9	9	"Stocks & Bondage"	Constantine Makris	Michael R. Perry	November 29, 1999	E0814	N/A
10	10	"Closure (Part I)"	Stephen Wertimer	Wendy West	January 7, 2000	E0815	N/A
11	11	"Bad Blood"	Michael Fields	Lisa Marie Petersen & Dawn DeNoon	January 14, 2000	E0816	N/A
12	12	"Russian Love Poem"	Rick Rosenthal	Eva Nagorski	January 21, 2000	E0813	14.30[7]
13	13	"Disrobed"	David Platt	Janet Tamaro	February 4, 2000	E0810	N/A
14	14	"Limitations"	Constantine Makris	Michael R. Perry	February 11, 2000	E0819	N/A
15	15	"Entitled"	Ed Sherin	*Teleplay by:* Robert Palm & Wendy West *Story by:* Dick Wolf & René Balcer & Robert Palm	February 18, 2000	E0818	N/A
16	16	"The Third Guy"	Jud Taylor	Dawn DeNoon & Lisa Marie Petersen	February 25, 2000	E0820	N/A
17	17	"Misleader"	Richard Dobbs	*Teleplay by:* Nick Kendrick *Story by:* Nick Harding & Nick Kendrick	March 31, 2000	E0812	N/A
18	18	"Chat Room"	Richard Dobbs	Roger Garrett	April 14, 2000	E0823	N/A
19	19	"Contact"	Michael Zinberg	Robert Palm & Wendy West	April 28, 2000	E0821	12.84[8]
20	20	"Remorse"	Alexander Cassini	Michael R. Perry	May 5, 2000	E0824	N/A
21	21	"Nocturne"	Jean de Segonzac	Wendy West	May 12, 2000	E0822	13.70[9]
22	22	"Slaves"	Ted Kotcheff	Lisa Marie Petersen & Dawn DeNoon	May 19, 2000	E0826	N/A

Figure 1

Law and Order: SVU Episodes Table

Password Cracking

In this NCL exercise we were given a list of usernames along with their password's hashes. See Table 1 below.

Table 1

User and Password Hash Table

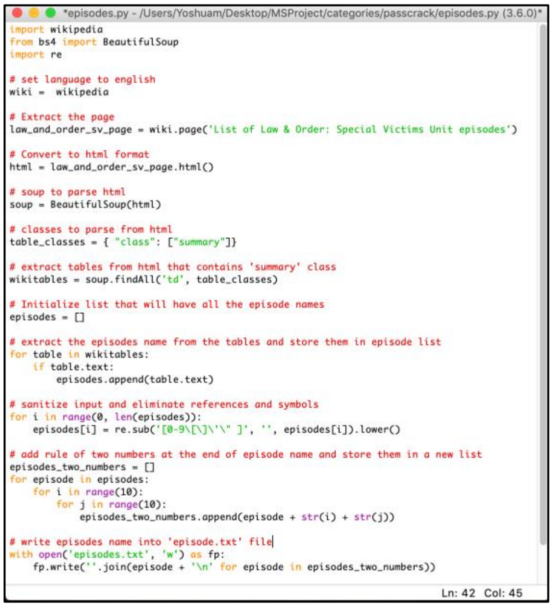
User	Password Hash
Justen	c6ffca47b477506eb331930cc6ae6292
Tom	9594e0f07b4e6e280c6131ce48dbf80d
Rachel	8609c7cc715dea6500e08db180b16f51
Eve	315d1cc9faafa74129769751fdd92ea3
Elliot	247e8adf7ede165ad0bd6032e4c0dfc6

The hint was that each password was a name of an episode of the series *Law and Order: SVU* followed by 2 digits. So, an example of this would be '*payback77*'. To solve it we need to do few steps:

1. Reconnaissance – Find the names of all the episodes in the series *Law and Order: SVU.*
2. Pre-Attack – Build a dictionary with the name of the episodes, each name containing 2 digits at the end of the episode's name.
3. Attack – Finally attack the passwords dump file and hope that we get the plaintext of each hash in the list.

In the first step we went to the source of all information in the internet the biggest encyclopedia called *Wikipedia* [26] to see if they had a list of all the episodes of *Law and Order: SVU* series. After, searching we find the following: See Figure 1.

It happened that Wikipedia had a list of all the episodes tabulated by seasons. That was really good but now the challenge was to extract all those episodes since this series had about 23 seasons with 22-23 episodes each.



episodes.py - /Users/Yoshuam/Desktop/MSProject/categories/passcrack/episodes.py (3.6.0)

```
import wikipedia
from bs4 import BeautifulSoup
import re

# set language to english
wiki =  wikipedia

# Extract the page
law_and_order_sv_page = wiki.page('List of Law & Order: Special Victims Unit episodes')

# Convert to html format
html = law_and_order_sv_page.html()

# soup to parse html
soup = BeautifulSoup(html)

# classes to parse from html
table_classes = { "class": ["summary"]}

# extract tables from html that contains 'summary' class
wikitables = soup.findAll('td', table_classes)

# Initialize list that will have all the episode names
episodes = []

# extract the episodes name from the tables and store them in episode list
for table in wikitables:
    if table.text:
        episodes.append(table.text)

# sanitize input and eliminate references and symbols
for i in range(0, len(episodes)):
    episodes[i] = re.sub('[0-9\[\]\'\" ]', '', episodes[i]).lower()

# add rule of two numbers at the end of episode name and store them in a new list
episodes_two_numbers = []
for episode in episodes:
    for i in range(10):
        for j in range(10):
            episodes_two_numbers.append(episode + str(i) + str(j))

# write episodes name into 'episode.txt' file
with open('episodes.txt', 'w') as fp:
    fp.write(''.join(episode + '\n' for episode in episodes_two_numbers))
```

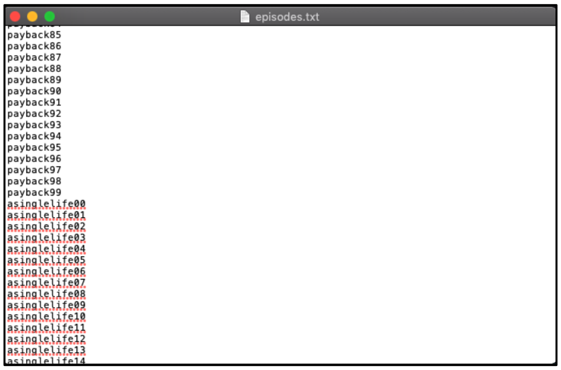
Ln: 42 Col: 45

Figure 2

episodes.py – Script for Scrapping all the Episodes Name from *Law and Order: SVU* Series

We decided to put in practice our programming skills and write a python script, see Figure 2, that will query Wikipedia page by title. After getting the wiki page content in html format, the script scrapped all the episodes name as a text from each table. Then, since some of the episodes contained links to more information, we had to remove all that because we only cared about the name. For this, we ran a regular expression that will remove all the special characters like, brackets, single quotes, double quotes, spaces and numbers to sanitize the episodes name. Then it will get each episode's name and will append all the possible combinations of 2 digits after its name and finally will store every value into a file.

After running the python script, we had a dictionary saved into a file named as episodes.txt containing all the episodes of Law & Order: SVU with two digits at the end, see Figure 3 below, that we used to crack each of the given hashes.



```
payback85
payback86
payback87
payback88
payback89
payback90
payback91
payback92
payback93
payback94
payback95
payback96
payback97
payback98
payback99
asinglelife00
asinglelife01
asinglelife02
asinglelife03
asinglelife04
asinglelife05
asinglelife06
asinglelife07
asinglelife08
asinglelife09
asinglelife10
asinglelife11
asinglelife12
asinglelife13
asinglelife14
```

Figure 3
episodes.txt – Dictionary Created by the Script

Now that we have the dictionary with all the possible passwords containing the rules given in the hint, we moved to the attack phase using hashcat tool. For this we wrote the following command: _\`hashcat -a 0 -m 0 laworder.hash episode.txt -O -o lawordercracked.txt\`_. What this command does is the following:

1. **-a 0** – tells hashcat to run a straight attack mode which means a normal attack given a hash with a possible password and compare
2. **-m 0** – tells hashcat to use md5 hash algorithm to hash.
3. **laworder.hash** – tells hashcat to use this file that contains the given hashes to crack.
4. **episodes.txt** –This tells hashcat to use episodes.txt as dictionary or wordlist to find the hashes.
5. **-O** – tells hashcat to run in optimized mode for better performance.
6. **-o** – tells hashcat to output the findings or cracked passwords into a file named lawordercracked.txt

After running this command hash cat gave us the following output see Figure 4 below:

```
9594e0f07b4e6e280c6131ce48dbf80d:folly63
315d1cc9faafa74129769751fdd92ea3:rockabye45
c6ffca47b477506eb331930cc6ae6292:annihilated35
247e8adf7ede165ad0bd6032e4c0dfc6:lunacy86
8609c7cc715dea6500e08db180b16f51:conned94
```

Figure 4
lawordercracked.txt – File Containing the Cracked Passwords

Finally, after we ran hashcat we could crack all the given passwords hashes and solve the challenge.

CTF TIME

This website is what we call the real test for our acquired skills. This website challenges are of very high complexity and you will be competing against the best teams in the world. Do perform well in this website and you for sure will be ready for any situation that presents in the work environment.

This website will take your skills to another level. Also, after each competition ends, participants are allowed to make write-ups about certain challenges allowing you to see how other people solve the problems using very detail explanation that will cover the why's and how's. This allows to create an even more competitive scenario since less experienced players can get ideas on how to approach similar problems in later competitions.

In addition, participating in this website will keep you sharp on every skill that you need to overcome the threats. They keep creating challenges that are mapped to recent technologies allowing the competitor to stay up to date with new vulnerabilities. To participate as a strong player, you will need to read cyber security related papers to came up with your own solves proving what is written in them. Most of the time you will need to formulate your own attacks, so you need a deep level of understanding on each of the given problems.

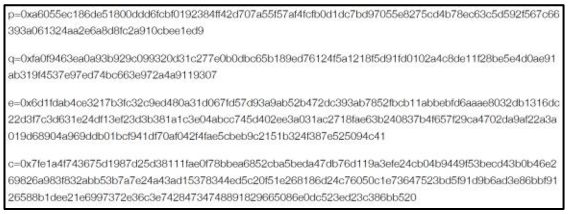
We recommend participating in all of the competitions held during the year because even though there will be competitions that you may not be able to solve a single challenge you will learn a lot from the write-ups and that will make you strengthen your skillset and become a better cyber security professional.

WORK DONE

We are going to share here some highlights of the work we did in CTFTime.org after our intensive training. We also going to include a high-level explanation of how each challenge was solved.

Cryptography

This challenge was given to us in ALEX CTF competition hosted on CTFTime.org webpage. In this challenge a guy named “Fady” send a file to another guy that contained information that at plain sight it would not make sense at all, see Figure 5 below.



```
p=0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e8275cd4b78ec63c5d592f567c66
393a061324aa2e6a8d8fc2a910cbee1ed9

q=0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0102a4c8de11f28be5e4d0ae91
ab319f4537e97ed74bc663e972a4a9119307

e=0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb11abbebfd6aaae8032db1316dc
22d3f7c3d631e24df13ef23d3b381a1c3e04abcc745d402ee3a031ac2718fae63b240837b4f657f29ca4702da9af22a3a
019d68904a969ddb01bcf941df70af042f4fae5cbeb9c2151b324f387e525094c41

c=0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24cb04b9449f53becd43b0b46e2
69826a983f832abb53b7a7e24a43ad15378344ed5c20f51e268186d24c76050c1e73647523bd5f91d9b6ad3e86bbf91
26588b1dee21e6997372e36c3e742847347488918296650 86e0dc523ed23c386bb520
```

Figure 5

Fadymsg.txt – File Containing Weird Information

At first, we tried to convert those hexadecimal numbers into decimal to see if they had any information, but they were just huge numbers. Then we recalled from our crypto training that there is a cipher that uses variables like *p, q, e* and *c* called RSA. Also, we recalled that RSA uses huge prime numbers to accomplish encryption using a simple math formula. Before going into details lets explain first how the RSA algorithm works so you have a better understanding of the math behind it and how we manipulate it to get our solution.

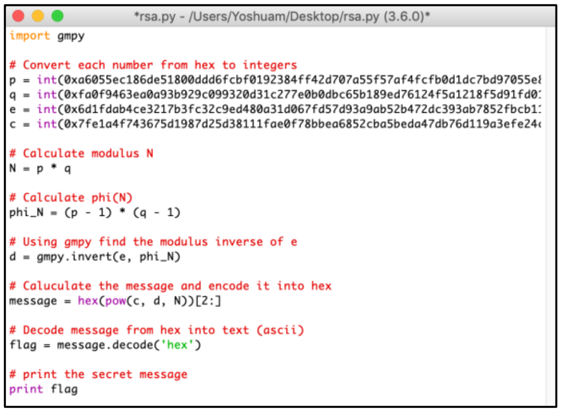
- First, RSA needs two very big prime numbers that are coprime with each other and they call them p and q.
- Second RSA calculates a modulus N by multiplying $p\ times\ q$ leading to the following formula: $N = \ \ p * q$
- After calculating the modulus N then, RSA needs to calculate a function called phi that looks as follow: $\varphi(N) = (p-1)(q-1)$
- Once we have function $\varphi(N)$ then RSA choose a number that meets the following two conditions: $e\begin{cases} \ \ 1 < e < \ \varphi(N) \\ coprime\ w\ N, \varphi(N) \end{cases}$
- Finally, we will calculate a number *d* such that $de\big(mod\ \varphi(N)\big) = 1$.

So now that we explained how RSA works, we need to understand how encryption and decryption works. To encrypt a message *m* using RSA all we need to do is encode message *m* into an integer number and then apply the following: $c = m^e\ mod\ N$ where c is the ciphertext or encrypted message resulting from the calculus. To decrypt a ciphertext c using RSA all we need to do is the following: $m = \ c^d\ mod\ N$ where m after being decoded into text will contain the message that was decrypted.

Now that we understand how we encrypt and decrypt all we need to do is look at the file and see that we have the prime numbers *p* and *q* and we also have the encryption exponent *e* and the ciphertext *c*. Since we know the ciphertext is what contains the message we need to figure out a way to decrypt the it. The problem is that in order to decrypt we need the exponent *d*. Also, we need that in order to obtain d we have to meet the following rule $de\big(mod\ \varphi(N)\big) = 1$.

We could try to brute force that, but the problem is that there are many numbers d that meets that criteria. Thankfully, from our training we knew that having the prime numbers p and q we could use inverse modulus math to calculate the exponent d. This works as follow: $e * d\ mod\ \varphi(N) = 1$ where d is any value that goes from 0 through $\varphi(N) - 1$. You may think that this method is quite inefficient but the extended version of the Euclidian's algorithm [27] allows faster searching of modulus inverse.

In order for us to solve this we used a python library called gmpy [28] that allows us to perform the inverse modulus of a number. After, we found this library we moved on to write the script that solved the challenge, see Figure 6 below.

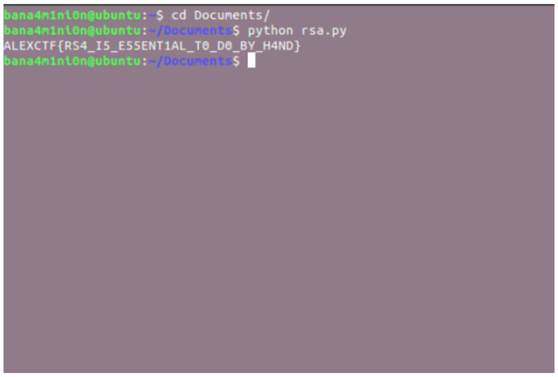


```python
import gmpy

# Convert each number from hex to integers
p = int(0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e
q = int(0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0
e = int(0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb1
c = int(0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24

# Calculate modulus N
N = p * q

# Calculate phi(N)
phi_N = (p - 1) * (q - 1)

# Using gmpy find the modulus inverse of e
d = gmpy.invert(e, phi_N)

# Caluculate the message and encode it into hex
message = hex(pow(c, d, N))[2:]

# Decode message from hex into text (ascii)
flag = message.decode('hex')

# print the secret message
print flag
```

Figure 6

rsa.py – Finding Exponent *d* and Decrypting the Message

After running the script, we got the following output. See figure 7 below.



```
bana4m1ni0n@ubuntu:~$ cd Documents/
bana4m1ni0n@ubuntu:~/Documents$ python rsa.py
ALEXCTF{RS4_I5_E55ENT1AL_T0_D0_BY_H4ND}
bana4m1ni0n@ubuntu:~/Documents$ 
```

Figure 7

rsa.py Output with Secret Message

As you can see above, we were able to decrypt the message and obtain the secret that was "ALEXCTF{RS4_I5_E55ENT1AL_T0_D0_BY_H4ND}".

Conclusion

Following this journey, we assure you that you will become skilled in the cyber security field. Also, if you mix up the theory and practice taught in your curriculum with a set of CTF events you will find yourself thinking like a hacker at the time of knowing what you will need to defend the system and network of your professional environment. We know that it is a long journey and with time the experience will flourish into a better full fledge cyber security ninja.

Always, take a step and learn about tools that you can use to facilitate some tasks to test your network and systems. This way you will learn if your defenses are strong enough to stand the common tools. Then, try to do it your way without the common tools and see if you still strong enough. We need to always remember that in this field of cyber security the training never ends.

Appendix A

List of Scripts, Challenges Solved and Tools

The structure of the package contains the masters project report along with an extended file called appendix.pdf where it contains a set of more challenges and proof of work of the competitions, we participated along with a very detail explanation of how each of them were solved. There are three folders, the references one, which contains a list of tools that we used and also a list of websites where anyone can use to train their knowledge in cybersecurity in a competition like environment. Also, there is a script and tools folder, which contains scripts and tools made by us to solve certain challenges. It also contains a tool called incognito tool that was develop by our team and contain several modules within it that are comprised by a rsa exploit, base converter tool, decoders, directory brute force tool and mathematical and string manipulation tools. Also, there is a Morse code decoder, a one-time pad module and tools to encrypt and decrypt for simple ciphers like vigenere and Caesar. Finally, there is a folder named "utils" that contains some of the props given on the challenges like pcap files, images containing secrets and logs for log analysis.

REFERENCES

[1] M. Hess. (2018, July 30). *How to prepare for capture the flag hacking competition* [Online]. Available: https://www.ctbnuggets.com/blog/2018/07/how-to-prepare-for-a-capture-the-flag-hacking-comeptition/.

[2] S. Van Acker. (2018, Oct 18). *Overthewhire* [Online]. Available: http://overthewire.org/wargames.

[3] J. Jolly. (2007, July 6). *What is Google Hacking (Google Scanning or Engine hacking?)* [Online]. Available: https://searchsecurity.techtarget.com/definition/Google-hacking.

[4] T. Akins. (n. d.). *Cipher Tools* [Online]. Available: http://rumkin.com/tools/cipher/.

[5] J. Ostrowick. (2005, Oct 10). *What is Dictionary Attack?* [Online]. Available: https://searchsecurity.techtarget.com/definition/dictionary-attack.

[6] C. Kowalczyk. (2013, Nov 1). *Known Plaintext Attack.* [Online]. Available: http://www.crypto-it.net/eng/attacks/known-plaintext.html.

[7] C. Kowalczyk. (2013, Nov 1). *Frequency Analysis* [Online]. Available: http://www.crypto-it.net/eng/attacks/frequency-analysis.html.

[8] E. Bilham and A. Shamir. "Differential Cryptanalysis of DES-like Cryptosystems: Advances in Cryptology" in *CRYPTO '90*. Springer-Verlag, 1990, pp. 2–21.

[9] A. Dujella. "A variant of Wieners Attack on RSA" in *Computing 85*, 2009, pp. 77-83.

[10] M. Tervooren (n. d.). *Factorize* [Online]. Available: https://factordb.com/.

[11] Vector 35 (n. d.). *Binary Ninja: A New Kind of Reversing Platform* [Online]. Available: https://binary.ninja/.

[12] Hex Rays (2015, May 27). *IDA PRO* [Online]. Available: https://www.hex-rays.com/products/ida/.

[13] D. Lebron. (2014). *RingZero CTF* [Online]. Available: https://ringzer0ctf.com/challenges.

[14] D. Daehee. (n. d.). *Pwnable.kr* [Online] Available: http://pwnable.kr/play.php.

[15] R. Larsen. (2013, Apr 28). *Pwntools* [Online]. Available: http://docs.pwntools.com/en/stable/.

[16] L. Chappell, *Wireshark Certified Network Analyst Exam Prep Guide*, 2nd ed., Saratoga: PODBOOKS.COM, LLC, 2012, pp.29-153.

[17] Aircrack-ng.org. (2006). *Aircrack-ng* [Online]. Available: https://www.aircrack-ng.org/.

[18] B. Mitchell. (2018). *Why WEP Keys Used to be Cool but Aren't Very Useful Anymore* [Online]. Available: https://www.lifewire.com/what-is-a-wep-key-818305.

[19] Z. Jiang, "Study of Wi-Fi Security Basing on Wireless Security Standards (WEP, WPA and WPA2)," in *Advanced Materials Research*, 2014, pp. 1049-1050, pp.1993-1996.

[20] NCL | National Cyber League | Ethical Hacking and Cyber Security. (2018). *NCL | National Cyber League | Ethical Hacking and Cyber Security* [Online]. Available: https://www.nationalcyberleague.org/.

[21] team, c. (2018). *CTFtime.org / All about CTF (Capture the Flag)* [Online]. Ctftime.org. Available: https://ctftime.org/.

[22] Y. Alicea. (April 25, 2017). *Cybersecurity Competitions as Effective Cybersecurity Teaching Tools* [Online]. Available: http://029e2c6.netsolhost.com/II-Proceedings/2017/IIVC 2017_ALICEA.pdf.

[23] Nmap.org. (n. d.). *Nmap: The Network Mapper - Free Security Scanner* [Online]. Available: https://nmap.org/.

[24] Owasp.org. (n. d.). *Category:OWASP DirBuster Project - OWASP* [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_DirBuster_.

[25] Hashcat.net. (n. d.). *hashcat - advanced password recovery* [Online]. Available: https://hashcat.net/hashcat/.

[26] Wikipedia.org. (n. d.). *Wikipedia* [Online]. Available: https://www.wikipedia.org/.

[27] En.wikipedia.org. (n. d.). *Extended Euclidean algorithm* [Online]. Available: https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm.

[28] PyPI. (n. d.). *GMPY* [Online]. Available: https://pypi.org/project/gmpy/.