Grenze International Journal of Engineering and Technology, Jan Issue

Dictionary List Creation using Web Scrapping and Password Cracking

Mr. S.Giridharan[1], Mr.E.Rajkumar[2], Dr.M.Senthil Kumar[3], A.Selva Muthukumaran[4], R.R.Swarna Krishna[5] and M Balaji[6]
[1-6]SRM Valliammai Engineering College/Department of Cyber Security, Chengalpattu, India
Email: {mailtoogiri, raju.becs39, msen1982, smkaravindan, rrsk261002, gurubalaji2002}@gmail.com

***Abstract*— Recovering a password from encrypted or hashed form is known as password cracking and attackers often use open source intelligence as assistance. The information gathered can vary from usernames, email addresses to personal information such as birth dates. Attackers use this information to create custom wordlists and apply different techniques such as brute force attacks, dictionary attacks and social engineering attacks for password cracking. This paper gives an outline of the password cracking process via web crawling and stresses the significance of implementing strong password policies and making users aware of the hazards of password reuse as well as weak passwords. Moreover, website owners should secure their databases and restrict access to personal information. By comprehending the methods used by attackers, individuals and organizations can defend themselves against password cracking attempts.**

***Index Terms*— web crawling, custom wordlist, open source intelligence (OSINT), social engineering.**

I. INTRODUCTION

Network security heavily relies on passwords which act as a barrier against unauthorized entry and protect the privacy of sensitive data. However, with the ever-increasing demand for passwords across numerous online platforms, it can be arduous to keep track of them all. Over time, users often generalize their passwords, making way for potential security breaches. Hackers can perform password cracking by using software to test different character combinations until the password is exposed. For this, they search for user-specific information available on various channels like social media, forums, or the internet. During brute force attacks, every possible combination of letters and digits are tried, which can be time-consuming, but fruitful if the password is weak. Dictionary attacks are also commonly used, where hackers use pre-compiled lists of commonly used passwords or names to guess the password successfully.

II. LITERATURE SURVEY

[1] It proposes a new approach for generating password dictionaries, which can be used for cracking passwords based on contextual information. The proposed methodology takes into account the common patterns and structures used in passwords, as well as the specific context in which the password was created. This approach is more effective than traditional dictionary generation methods that do not consider contextual information. To test

Grenze ID: 01.GIJET.10.1.391
© Grenze Scientific Society, 2024

their approach, the authors conducted experiments using real-world password datasets. The results showed that their methodology outperformed traditional dictionary-based password cracking.
[2] It explores the effect of using "leet" or "1337" substitutions in passwords on their strength and security. Leet is a technique where letters are replaced with numbers or symbols that resemble them, such as "3" for "E" or "@" for "A". The authors conducted a study using a large dataset of leaked passwords and found that leet substitutions are commonly used by users to create passwords, with the most popular substitution being "3" for "E". However, they also found that such substitutions do not significantly improve the strength of passwords, and can even make them weaker if the substitution is too predictable. The paper concludes that while leet substitutions may make passwords more memorable for users, they do not significantly improve password security and can even decrease it if the substitution is too predictable.
[3] It proposes a novel method for constructing a sentiment dictionary using web crawling techniques. The sentiment dictionary can be used for sentiment analysis, which is a vital component in many natural language processing applications. The proposed method focuses on three main aspects of sentiment analysis: (1) collecting a diverse set of opinionated text from the web, (2) filtering out irrelevant text, and (3) labeling the remaining text with sentiment scores. The method uses a multi-stage approach that includes crawling the web, filtering out irrelevant web pages, extracting opinionated text, and labeling the text with sentiment scores. The sentiment scores are calculated using a supervised learning approach based on a set of seed words. Overall, the paper presents a novel and effective method for sentiment dictionary construction that can be used to improve the accuracy of sentiment analysis in various natural language processing applications.
[4] It proposes a novel approach for attacking machine learning-based systems used in cybersecurity applications. The authors highlight the growing use of machine learning algorithms in intrusion detection systems, and the need to evaluate their robustness against potential attacks. The proposed approach is a brute-force black-box method that involves generating a large number of adversarial samples and testing them against the target machine learning model. The method is based on the assumption that the attacker has no knowledge of the target model's parameters or training data. The authors demonstrate the effectiveness of the approach on two popular intrusion detection systems, Snort and Suricata. The results show that the proposed approach is able to successfully attack the models, even when they are trained with state-of-the-art techniques for robustness. Overall, the paper provides valuable insights into the vulnerabilities of machine learning-based systems used in cybersecurity.
[5] It presents a method to crack the password protection of PDF 2.0 documents using GPU acceleration. The authors use a tool called Hashcat to perform the password cracking and find that the use of GPUs significantly speeds up the process compared to using CPUs. The authors first discuss the PDF 2.0 specification and the various types of password protection it supports, including encryption and access permissions. They then describe the Hashcat tool and its use for password cracking. The authors find that using GPUs can greatly reduce the time required to crack PDF 2.0 passwords. They test various password lengths and complexity levels and find that longer and more complex passwords take longer to crack, but still benefit from GPU acceleration. The authors also compare the performance of different GPU models and find that newer models with higher memory bandwidth perform better. The paper concludes by highlighting the importance of strong password protection and the need for users to use longer and more complex passwords to prevent password cracking attacks.

III. Proposed Work

Data Mining

- Open-source intelligence (OSINT) can be enhanced through the use of data mining techniques to sift through publicly available information from sources like the internet and media. By leveraging big data analysis methods, patterns and insights can be extracted to aid in intelligence gathering.
- Among the approaches commonly employed for such extraction are web scraping and text analysis, respectively used to gather relevant data on entities or to decipher key themes and sentiments.
- OSINT data mining can prove useful in various fields, from tracking security threats to gauging public opinion or gaining strategic insights. It is of utmost importance, however, that all data mining undertakings in this realm adhere to ethical and legal norms.
- The code starts by importing several libraries that will be used throughout the program, including requests, BeautifulSoup, re, random, BytesIO and pdfminer.high_level for PDF processing, and spacy for natural language processing.
- The spacy library is used to load the English language model for NLP processing.

- The extract_text(url) function is defined to extract text from a given URL. The function checks if the URL ends with .pdf and uses the pdfminer library to extract text from the PDF.
- If the URL does not end with .pdf, the function uses requests and BeautifulSoup to extract text from the HTML content. The function then removes all non-alphanumeric characters and converts the text to lowercase.

```
def extract_text(url):
    if url.endswith('.pdf'):
        response = requests.get(url)
        with BytesIO(response.content) as data:
            # Extract text from all pages of the PDF
            text = ""
            with BytesIO() as output:
                extract_text_to_fp(data, output)
                text = output.getvalue().decode()
    else:
        response = requests.get(url)
        soup = BeautifulSoup(response.text, 'html.parser')
        text = soup.get_text()
    text = text.replace(u'\xa0', u' ')
    text = re.sub(r'\W+', ' ', text).lower()
    return text
```

- The get_important_words(text, n=4) function is defined to get the most important words from a given text. The function first splits the text into individual words and removes common words that are not useful as passwords.
- It then uses spacy to extract named entities from the text, specifically people's names. The function then combines the remaining words and the named entities into a list of unique words and randomly selects n words from the list.
- The save_password(password) function is defined to save the selected password to a file named "password.txt".

```
def save_password(password):
    with open("password.txt", "w") as f:
        f.write(' '.join(password))
```

- The main() function is defined to run the program. The function prompts the user to enter a list of website URLs separated by commas. The function then uses the extract_text() function to extract text from each website URL, combines all the extracted text into a single string, and uses the get_important_words() function to select the most important words for the password.
- Finally, the main() function is called to run the program.In summary, the program extracts text from website URLs, processes the text to extract the most important words,and uses those words to generate a secure password. The program is designed to be user-friendly and easy to use, and it has potential applications in the field of cybersecurity for generating strong and secure passwords.

IV. MODULE

A. Module 1: Converting To Exe File

- An application called auto-py-to-exe is utilized in te conversion of webscrapping in the following ways
- Converting a Python script that involves dictionary creation into an executable (.exe) file, you can follow a similar process using tools like PyInstaller or cx_Freeze
- Install PyInstaller:
- Open your terminal or command prompt and run the following command to install PyInstaller:
- pip install pyinstaller

Navigate to your script's directory:
Use the cd command to go to the directory where your Python script is located.
cd path/to/your/script
Create the executable:
Run the following command to create a standalone executable file:

- pyinstaller --onefile your_script.py

Replace "your_script.py" with the actual name of your Python script. The --onefile option bundles everything into a single executable file.
Locate the executable:
PyInstaller will create a dist directory in your script's directory. Inside the dist directory, you'll find the executable file.

- Python script that involves dictionary creation (example_script.py):

example_script.py

```
my_dict = {
    'key1': 'value1',
    'key2': 'value2',
    'key3': 'value3'
}
for key, value in my_dict.items()
 print(f'{key}: {value}')
```

After running PyInstaller, find the executable in the dist directory.

```
cd path/to/your/script
pyinstaller --onefile example_script.py
```

The resulting executable will be in the dist directory, and you can distribute it as needed.

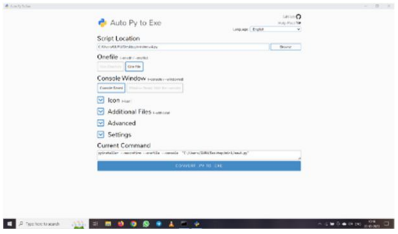

Figure 1: Screenshot of auto-py-to-exe

METASPLOITABLE 2

- The research employs Metasploitable 2, a unique Linux operating system. This machine is used for penetration testing
- Download the Metasploitable 2 virtual machine image from the Offensive Security website or from another reliable source. The file will be in compressed format, so extract it to a location of your choice.
- Install a virtualization software such as VirtualBox or VMware Workstation on your host operating system. These applications allow you to create and run virtual machines on your computer.
- Launch the virtualization software and click on the "Import" or "New VM" button to create a new virtual machine. Choose the option to import the Metasploitable 2 virtual machine image that you downloaded earlier.
- Configure the virtual machine settings, such as the amount of RAM and CPU allocated to the virtual machine. You can also customize other settings, such as the network adapter, hard disk size, and display resolution.
- Start the Metasploitable 2 virtual machine. The first time you start it, the virtual machine will go through a setup process that will take a few minutes. Once the setup is complete, you will be prompted to log in.

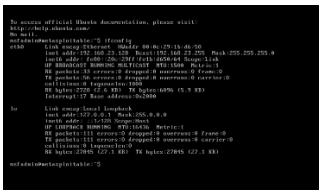

Figure 2: Metasploitable 2 system

B. Module 2: SCANNING NETWORK

- To scan the ports in our network (metasploitable 2) we use a tool called nmap.Nmap can be used to create network maps, track network uptime, and gather information about remote hosts.

- Nmap uses a variety of techniques to determine what hosts and services are running on a network, including TCP/IP fingerprinting, port scanning, and OS detection.
- This software can also identify and exploit known vulnerabilities in network services and operating systems.
- One of the great things about Nmap is that it can be run on various operating systems, including Linux, Windows, and macOS, and can be used either from the command line or through a graphical user interface.
- Whether you are a network administrator, security professional, or someone simply interested in network exploration and management, Nmap is an essential tool that should not be overlooked.

Figure 3: Screenshot showing available options in nmap

PORT STATE SERVICE VERSION

21/tcp open ftp vsftpd 2.3.4
22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp open telnet Linux telnetd
25/tcp open smtp Postfix smtpd
53/tcp open domain ISC BIND 9.4.2
111/tcp open rpcbind 2 (RPC #100000)
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp open exec netkit-rsh rexecd
513/tcp open login OpenBSD or Solaris rlogind
514/tcp open tcpwrapped
1099/tcp open java-rmi GNU Classpath grmiregistry
2049/tcp open nfs 2-4 (RPC #100003) 2121/tcp open ftp ProFTPD 1.3.13306/tcp open mysql MySQL 5.0.51a-3ubuntu5
5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5
After conducting an analysis of Metasploitable2 using Nmap, possible system weaknesses were identified that could be exploited by unauthorized users. These weaknesses were listed in a report that included details about open ports, active services and vulnerabilities that could jeopardize the security of the system.

GPU's for Password Cracking

The latest graphics card from NVIDIA is an impressive computing tool equipped with advanced architecture and cutting-edge technology. Its high-speed memory and massive parallel processing power enable it to perform brute force attacks at an unprecedented speed, potentially cracking even the most complex passwords. It's essential to ensure the security of your online accounts by using strong and unique passwords and taking additional security measures to avoid potential attacks

C. Module 3: CRACKING THE PASSWORD

- The password collected from data mining process is used in a password cracking tool called hydra.This open-source password cracking tool can test a system's security by employing a range of at-tacks against various protocols and services. Additionally, Hydra offers support for a wide variety of authentication methods, making it a versa-tile option for testing different systems and applications.
- In order to use Hydra, a list of usernames and passwords is employed to send login requests to a target system. While Hydra can be a time saver, it must only be used for authorized testing purposes with the system owner's permission. Using Hydra illegally can bring severe consequences.
- The password has been retrieved by Hydra, concluding the search and enabling access to the secured account

Figure 4: Screenshot showing available options in hydra

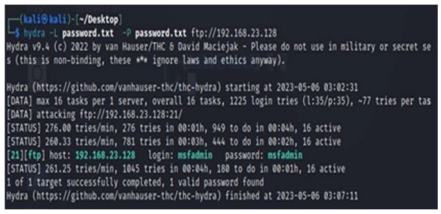

Figure 5: Cracking the password using hydra

IV. Conclusion

In this paper we had analysed the password cracking methods from gathering information from web scrapping and using GPU that will increase the probability to bypass the security. Once the information has obtained it can be used to launch password cracking attack. It involves the use of various techniques to gain unauthorized access to user accounts by guessing or discovering their passwords. Attackers can gather information from websites using various methods, including social engineering, phishing, and data breaches. Once they have obtained this information, they can use it to launch password cracking attacks. The contribution of this paper lies in its innovative approach to dictionary generation for password cracking, which considers the contextual information that is often associated with password creation. These provided methodologies has the potential to be useful for security professionals and law enforcement agencies in the fight against cybercrime, as it provides a more accurate and efficient means of cracking passwords. Overall, this paper provides valuable information about the development of novel techniques for password cracking and highlights the importance of considering contextual information in the generation of password dictionaries.

References

[1] Kanta, I. Coisel and M. Scanlon, "A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking," in IEEE Access, vol. 10, pp. 59178-59188, 2022, doi: 10.1109/ACCESS.2022.3179701.

[2] W. Li and J. Zeng, "Leet Usage and Its Effect on Password Security," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2130-2143, 2021, doi: 10.1109/TIFS.2021.3050066.

[3] W. On, J. -Y. Jo, H. Shin, J. Gim, G. S. Choi and S. -M. Jung, "Efficient Sentiment-Aware Web Crawling Methods for Constructing Sentiment Dictionary," in IEEE Access, vol. 9, pp. 161208-161223, 2021, doi: 10.1109/ACCESS.2021.3129187.

[4] S. Zhang, X. Xie and Y. Xu, "A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity," in IEEE Access, vol. 8, pp. 128250-128263, 2020, doi: 10.1109/ACCESS.2020.3008433.

[5] F. Yu and H. Yin, "Password Cracking of PDF 2.0 documents on GPU," 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu, China, 2021, pp. 721-725, doi: 10.1109/ICCCS52626.2021.9449188.

[6] A. Kanta, I. Coisel and M. Scanlon, "Smarter Password Guessing Techniques Leveraging Contextual Information and OSINT," 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, 2020, pp. 1-2, doi: 10.1109/CyberSecurity49315.2020.9138870.

[7] S. Salamatian, W. Huleihel, A. Beirami, A. Cohen and M. Médard, "Centralized vs Decentralized Targeted Brute-Force Attacks: Guessing with Side-Information," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 37493759,2020, doi:10.1109/TIFS.2020.2998949.

[8] W. Jia, "Analysis on Password Attack Model and Password Generation," 2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Ottawa, ON, Canada, 2022, pp. 145-149, doi:10.1109/CIPAE55637.2022.00038.

[9] S. Schechter, Y. Tian and C. Herley, "StopGuessing: Using Guessed Passwords to Thwart Online Guessing," 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 2019, pp. 576-589, doi: 10.1109/EuroSP.2019.00048.

[10] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol and G. Martínez Pérez, "The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends," in IEEE Access, vol. 8, pp. 10282-10304, 2020, doi: 10.1109/ACCESS.2020.2965257.