A New Method terms of Cyber Security: Fight, Flight, Freeze Effect

Özgecan SİYEZ

ABSTRACT:

This article aims to model the amygdala’s fight, flight, and freeze responses within the context of cybersecurity, to enable more intuitive and rapid responses to cyber threats. The developed model labels each type of cyber attack with numbers attributed to fundamental defense responses. Although the practical effectiveness of the model has not yet been tested, it is theoretically considered to add a new dimension to cybersecurity strategies. The article proposes that this approach could improve the adaptation of cybersecurity systems to the dynamic and continuously changing threat environment and make defense mechanisms more flexible.

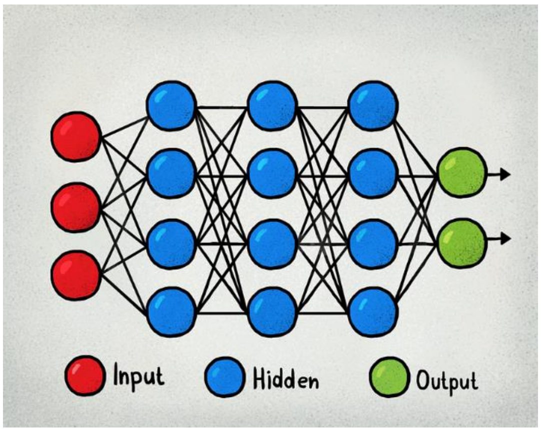
Keywords: Artificial neural networks, Cybersecurity, Amygdala, Fight response, Flight response, Freeze response, Threat modeling, Intuitive defense, Dynamic adaptation, Flexible defense mechanisms

1. INTRODUCTION:

Artificial Neural Networks (ANNs) are designed inspired by the structure of the brain, yet they are not exactly the same as the actual brain structure. Nevertheless, some features and the continuity of ANNs show similarities to the ways the human brain operates. Artificial Neural Networks consist of artificial entities that mimic biological parts. These artificial formations create a neural network by establishing changeable connections. These computers represent the synapses of nerve cells (neurons) in the real brain. Artificial Neural Networks are typically organized in layers. The most basic structure consists of an input layer, hidden layer(s), and an output layer. Transitions between layers allow the neural network to distribute information and reflect these divisions, mirroring the neural action potentials (communication between neurons) in the real brain. ANNs can be used in the field of cybersecurity to detect potential security vulnerabilities by recording network connections and analyzing log data, helping to identify signs of attack by defining normal network structural patterns and abnormalities, with intrusion detection systems (IDS) and intrusion prevention systems (IPS) available. ANNs can detect signs of attack by continuously monitoring network details and system logs and can automatically block attacks and be used to detect and open malicious software. By learning the behaviors and characteristics of malware, they can be effective in detecting unknown malicious software. They can be used for identity verification and access control by monitoring users’ information, modeling normal user operations, and detecting abnormalities to identify unauthorized access attempts. They can be used in the development of security protocols and encryption processes. They can be used to

https://doi.org/10.33774/coe-2024-xszz9 **ORCID:** https://orcid.org/0009-0008-4985-7191 Content not peer-reviewed by Cambridge University Press. **License:** All Rights Reserved

create encryption keys, optimize firewalls, and establish secure communication (Öztemel, E.,2003).



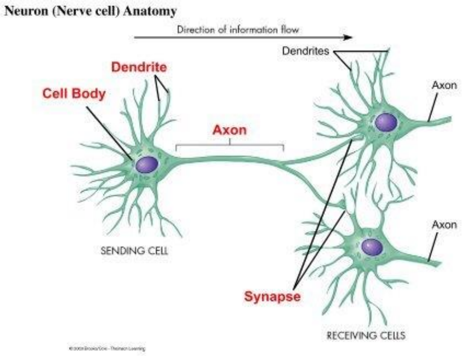


Figure 1: Artificial Neural Networks **Figure 2:** Neuron (Nerve Cell) Anatomy

2. LIMBIC SYSTEM AND AMYGDALA

Dendrites in Artificial Neural Networks (ANNs) represent the input systems that receive data from the external world or another programming unit. This input layer is the starting point of the model and receives inputs in a manner similar to biological dendrites. In ANNs, instead of axons, there are weights or connections. These weights emerge from the input layers.

The amygdala, as a brain structure, is accessible through dendrites, axons, and synapses, and communicates these services with other brain regions. The amygdala has dendrites to receive neurotransmitters at the ends of nerve fibers (axons) coming from various brain regions. Dendrites facilitate the transfer of information from other sections into the amygdala. They particularly process sensory information, especially values that are emotionally significant, such as threat perception. The amygdala is connected to many different structures in the brain through axons. For example, the amygdala is associated with structures such as the hypothalamus, hippocampus, thalamus, and brainstem. These axons are used to transmit treatments from the amygdala to other brain regions or to transport incoming devices to the amygdala. Sections within the amygdala communicate with other sections through synapses. Synapses are small parts that provide the connection between parts. Neurotransmitters reach receptors attached to the dendrites as an extension of the synaptic area's breaks and then form nerve impulses through the dendrites.

The amygdala, through these dendrites, axons, and synapses, communicates with various brain regions to regulate emotional responses and contribute to the development of emotional experiences. It plays a significant role in regulating and memorizing emotional states such as fear, anger, and stress (ÖZTÜRK E. N., & KARŞIDAĞ, Ç., 2023).

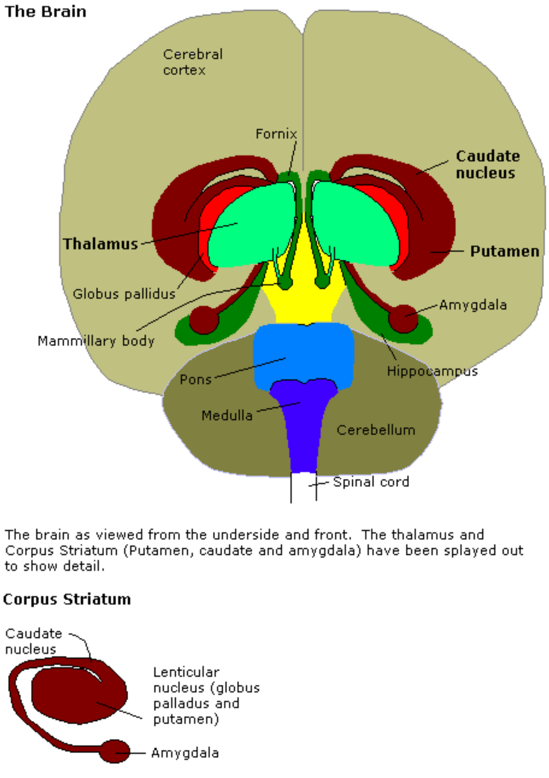

The Brain
Cerebral cortex
Fornix
Caudate nucleus
Thalamus
Putamen
Globus pallidus
Amygdala
Mammillary body
Hippocampus
Pons
Medulla
Cerebellum
Spinal cord
The brain as viewed from the underside and front. The thalamus and Corpus Striatum (Putamen, caudate and amygdala) have been splayed out to show detail.
Corpus Striatum
Caudate nucleus
Lenticular nucleus (globus palladus and putamen)
Amygdala

Figure 3: Amygdala Structure

2.1.The Amygdala's "Fight, Flight, Freeze" Response

From a neuroscientific perspective, the "fight, flight, or freeze" responses are different behavioral responses that occur under the influence of the amygdala. The amygdala plays a critical role in regulating emotional and behavioral responses when encountering dangerous or stressful situations.

1. **Fight**:
   - When danger is perceived, the amygdala spreads signals that trigger aggression and combat behaviors in other areas of the brain.
   - The fight response causes an increase in communication between brain regions that are particularly active when the amygdala is engaged.
   - This response involves the activation of areas such as the hypothalamus and brainstem, along with the expansion of the unit and pumping more blood to the muscles to prepare for action.
2. **Flight**:
   - Upon perceiving danger, the amygdala sends signals that trigger escape and avoidance behaviors from other areas of the brain.
   - The flight reaction changes with the activation of brain regions that facilitate rapid movement. This includes a series of brain hosts located within the hypothalamus and brainstem.
   - This reaction involves preparing for care and hiding to move away from danger.

-

3. **Freeze**:
    - When danger is perceived, the amygdala sends signals that trigger immobility and freezing behaviors in other areas of the brain.
    - The freeze response can change with the activation of brain regions that cause immobility and ongoing pauses.
    - This response is considered a strategy for hiding or remaining undetected against danger and is sometimes used to ensure the danger goes unnoticed (KayaY, A., & CoşkuntanE, İ.).

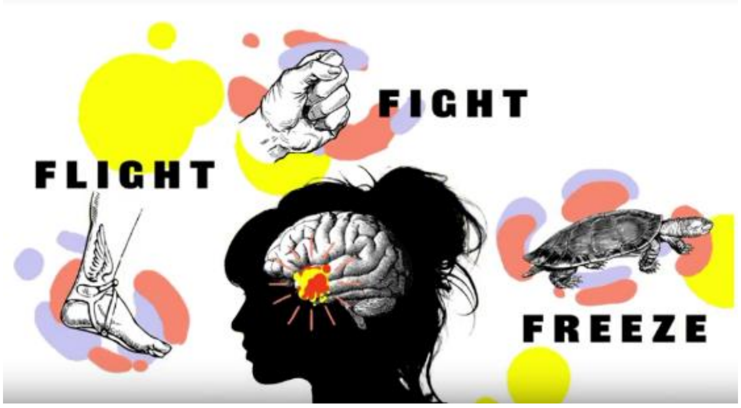


Figure 4: Fight, flight, freeze response

3. CYBERSECURITY

Cybersecurity is a discipline that ensures the protection of computer systems, networks, devices, and data against malicious or unauthorized access, use, alteration, or destruction. Fundamentally, cybersecurity involves various technological, legal, and administrative measures to protect information and systems in the digital world. These measures aim to provide protection against various cyber threats such as hackers, malware, data breaches, and phishing. Cybersecurity also requires continuous monitoring and updating to detect security vulnerabilities, reduce risks, and respond quickly in crisis situations (Hekim, H., & BAŞIBÜYÜK, O., 2013).

3.1.What is Machine Learning and How is it Used in Cybersecurity?

Machine Learning (ML) is a type of algorithm that enables computers to analyze data, learn from past experiences, and make decisions in a manner similar to human behavior. Machine learning algorithms in cybersecurity can automatically detect and analyze security incidents. Some can even respond to threats automatically. Many modern security tools, such as threat intelligence, already utilize machine learning. There are many machine learning algorithms, but most perform one of the following tasks:

- **Regression**: Identifies correlations between different datasets and understands how they relate to each other. You can use regression to predict operating system calls and then identify anomalies by comparing the prediction with an actual call.
- **Clustering**: Determines similarities between datasets and groups them according to their common characteristics. Clustering works directly on new data without considering previous examples.
- **Classification**: Classification algorithms learn from previous observations and try to apply what they have learned to new, unseen data. Classification involves taking artifacts and classifying them under one of several labels. For example, it classifies a binary file under categories such as legitimate software, adware, ransomware, or spyware (Seyyarer, E., & Ayata, F., 2023).

3.2.Cyber Attacks and Their Types

Cyber attacks are malicious actions carried out with the intent to damage or incapacitate computer systems, networks, or digital assets.

- **Neural Fuzzing**: Used to detect software vulnerabilities by testing large amounts of random input data. A threat actor could combine neural fuzzing with neural networks to gather information about a target software or system and learn its weaknesses.

Network Attacks:

- Denial of Service (DoS) Attacks
- Distributed Denial of Service (DDoS) Attacks
- Man-in-the-Middle (MITM) Attacks
- ARP Spoofing Attacks
- DNS Spoofing Attacks
- Port Scanning Attacks

Malware:

- Viruses
- Worms
- Trojan Horses (Trojans)
- Rootkits
- Ransomware
- Adware
- Spyware
- Botnets

Identity Theft and Data Breaches:

- Phishing Attacks
- Identity Theft
- Data Breaches
- Authentication Attacks

Physical Attacks:

- Hardware Misuse
- Hardware Theft
- Electronic Lock Picking
- CCTV System Manipulation

Social Engineering:

- Voice Phishing (Vishing)
- Tailgating
- Information Gathering
- Email Spoofing

Web Application Attacks:

- SQL Injection
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Session Hijacking
- Clickjacking

Password Attacks:

- Brute Force Attacks
- Dictionary Attacks
- Rainbow Table Attacks

IoT (Internet of Things) Attacks:

- Botnet Attacks
- Firmware Attacks
- DDoS Attacks

Ransom Attacks:

- Data Encryption
- Data Deletion
- Data Publication

Internet Protocol (IP) Attacks:

- IP Spoofing
- IP Fragmentation Attacks
- ICMP Attacks

Cybersecurity Awareness and Measures The number and diversity of cyber attacks are continually increasing. Therefore, it is crucial to implement cybersecurity measures and enhance security awareness (ARSLAN, M. E.).

Names for cyber attacks are typically given by various sources such as cybersecurity experts, security companies, governments, and media organizations. The naming often refers to the attack's characteristics, the methods used, the systems affected, or the industry.

Some cyber attack names have been standardized by the cybersecurity community. For example, the Common Vulnerabilities and Exposures (CVE) system is used to assign unique identifiers to known security vulnerabilities and attacks. A CVE identifier defines a specific security vulnerability.

Some attacks may also be referred to by unique names given by hackers or cybercrime organizations themselves, such as the WannaCry ransomware attack or the Stuxnet worm.

Media organizations may also name cyber attacks when reporting on them, often based on the attack's scale, its impacts, or other factors that attract media attention.

However, not every cyber attack is named, and some only remain in the spotlight for a short period. Especially minor or commonly seen attacks are often not referred to by a special name.

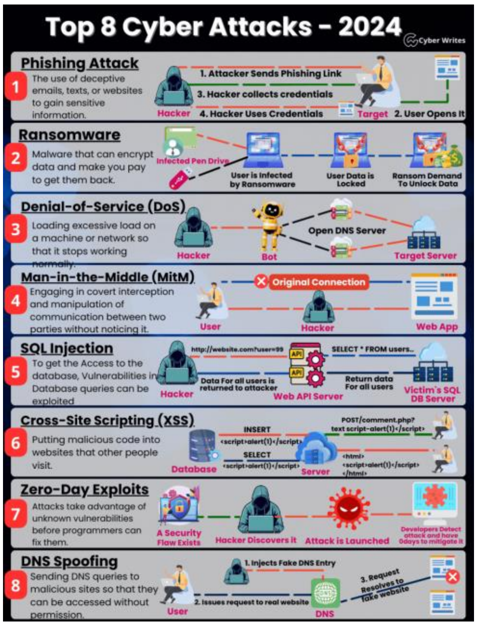

Top 8 Cyber Attacks - 2024
Cyber Writes
Phishing Attack
1
The use of deceptive emails, texts, or websites to gain sensitive information.
1. Attacker Sends Phishing Link
3. Hacker collects credentials
Hacker
4. Hacker Uses Credentials
Target
2. User Opens It
Ransomware
2
Malware that can encrypt data and make you pay to get them back.
Infected Pen Drive
User is Infected by Ransomware
User Data is Locked
Ransom Demand To Unlock Data
Denial-of-Service (DoS)
3
Loading excessive load on a machine or network so that it stops working normally.
Hacker
Bot
Open DNS Server
Target Server
Man-in-the-Middle (MitM)
4
Engaging in covert interception and manipulation of communication between two parties without noticing it.
Original Connection
User
Hacker
Web App
SQL Injection
5
To get the Access to the database, Vulnerabilities in Database queries can be exploited
http://website.com?user=99
SELECT * FROM users..
API
API
Data For all users is returned to attacker
Return data For all users
Hacker
Web API Server
Victim's SQL DB Server
Cross-Site Scripting (XSS)
6
Putting malicious code into websites that other people visit.
INSERT
<script>alert(1)</script>
SELECT
<script>alert(1)</script>
POST/comment.php?
text script-alert(1)</script>
<html>
<script>alert(1)</script>
</html>
Database
Server
Zero-Day Exploits
7
Attacks take advantage of unknown vulnerabilities before programmers can fix them.
A Security Flaw Exists
Hacker Discovers it
Attack is Launched
Developers Detect attack and have 0days to mitigate it
DNS Spoofing
8
Sending DNS queries to malicious sites so that they can be accessed without permission.
1. Injects Fake DNS Entry
2. Issues request to real website
3. Request Resolves to fake website
User
DNS

Figure 5: Brief description of 8 common cyber attacks

The Common Vulnerabilities and Exposures (CVE) is an identification system used in the field of computer security. CVE is utilized to define known security vulnerabilities and threats with a unique identifier. These identifiers, known as CVE numbers, are typically in the format "CVE-YYYY-NNNN", where "YYYY" represents the year and "NNNN" is a number.

The CVE system was established by the cybersecurity community to track, report, and share known security vulnerabilities. CVE numbers allow for the clear definition of a security vulnerability or threat, facilitating different security professionals and organizations to refer to the same vulnerabilities.

CVE numbers are recorded in a database managed by the National Cybersecurity Center (NVD). Security experts can use this database to research, report, and monitor known security

vulnerabilities. Additionally, CVE numbers can also serve as a reference for manufacturers to release security patches or fixes.
A specific security vulnerability or threat corresponding to a CVE number can be learned from resources created for that CVE, security reports, or cybersecurity databases. For example, sources like the CVE Details Database, NIST National Vulnerability Database (NVD), or MITRE's CVE website provide information on security vulnerabilities and threats associated with a particular CVE number.

Each CVE number is expected to correspond to a security vulnerability or threat known to the security community. However, sometimes different CVE numbers may be assigned to the same security vulnerability or threat from different sources, or a single CVE number may describe multiple security vulnerabilities or threats. Therefore, it is important to research the relevant sources for a complete understanding of the security vulnerability or threat corresponding to a CVE number.

Cyber attack types are not usually paired with a specific CVE number because CVE numbers are generally used to identify specific security vulnerabilities or threats. However, there are known security vulnerabilities and threats for certain types of cyber attacks, and CVE numbers may have been assigned to these vulnerabilities or threats. Nonetheless, there may not be a CVE number for every type of cyber attack, and some may have multiple CVE numbers.
For example, there is no specific CVE number for DDoS (Distributed Denial of Service) attacks, a popular type of cyber attack, because DDoS attacks are typically designed to overload network resources rather than exploit a security vulnerability. However, some tools or methods used to carry out DDoS attacks may have specific security vulnerabilities, and CVE numbers may have been assigned to these vulnerabilities.

Similarly, other types of cyber attacks, such as ransomware attacks, may not have specific CVE numbers. Nonetheless, ransomware attacks usually involve malicious software and the security vulnerabilities they exploit. Therefore, there may be specific CVE numbers for the malicious software used in ransomware attacks or the vulnerabilities they exploit.

Pairing cyber attack types with specific CVE numbers is a complex issue, and there is no definitive rule for each type of attack. However, security vulnerabilities or threats are usually identified for known cyber attacks by the security community, and one or more CVE numbers are assigned to these vulnerabilities or threats. Therefore, it is important to research relevant security sources to determine whether a specific type of cyber attack can be associated with a specific CVE number.

The CVE system is an important tool for promoting collaboration in the field of cybersecurity and better managing security vulnerabilities. However, CVE numbers only identify known security vulnerabilities and do not cover all cybersecurity risks. Therefore, CVE numbers should be used only as one component and evaluated in conjunction with other security measures (Akram, J., and Ping, L., 2020).
Here are some examples of CVE numbers and a cybersecurity measure that can be taken against a vulnerability:

CVE Examples:

- **CVE-2021-3456**: A vulnerability in the sudo program in Linux operating systems that allows an attacker to execute malicious code via a specially crafted package.

- **CVE-2020-0601**: A vulnerability in Windows operating systems that allows an attacker to use digital certificates for spoofing.
- **CVE-2019-0708**: A vulnerability in Microsoft's RDP (Remote Desktop Protocol) service that allows an attacker to perform remote code execution attacks.
- **CVE-2018-11776**: A vulnerability in the Apache Struts web application framework that allows an attacker to perform remote code execution attacks on web applications.
- **CVE-2017-0144**: A vulnerability in Microsoft's SMB (Server Message Block) protocol that was used for an infiltration attack known as EternalBlue, which was widely used during the WannaCry ransomware attack.

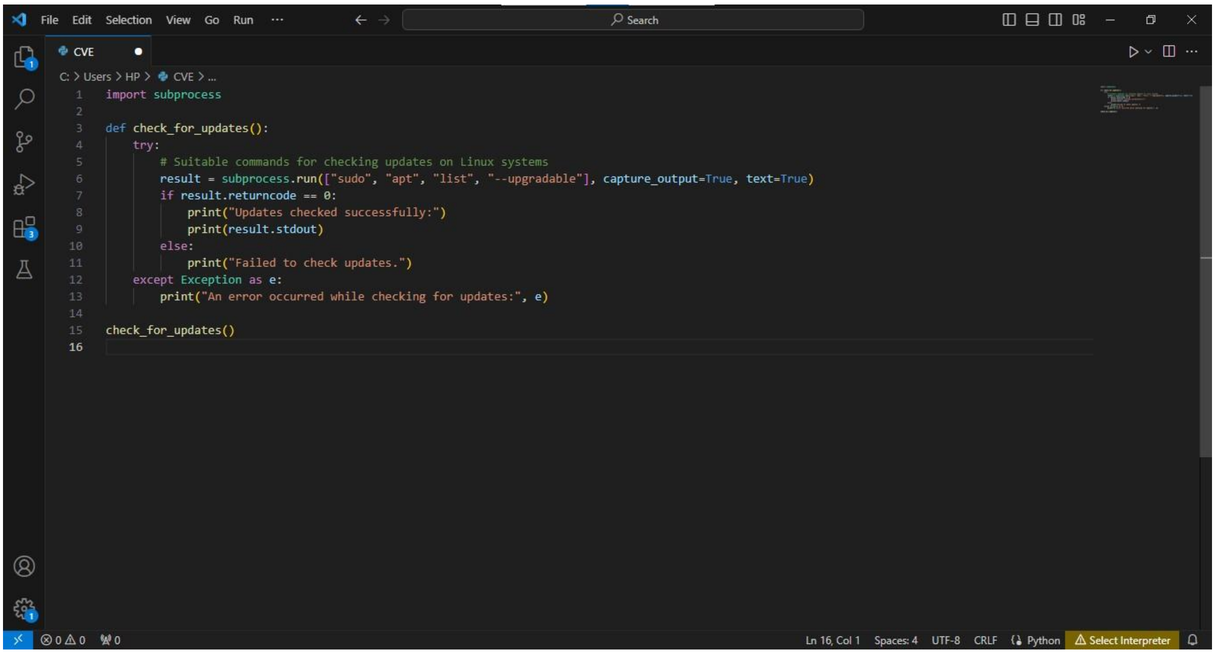
These CVE numbers are examples and each identifies a specific security vulnerability or threat. These numbers are commonly used in the cybersecurity community and industry to denote known security vulnerabilities (www.nvd.nist.gov).

Cybersecurity Measure Example:

Scenario: An organization has identified systems affected by the security vulnerability with the number CVE-2021-3456 in the sudo program on Linux operating systems.

Cybersecurity Measure: A measure that can be taken against this vulnerability is to update the sudo program on the affected systems and ensure that the patch is applied.

Example Python Code (Update Check):



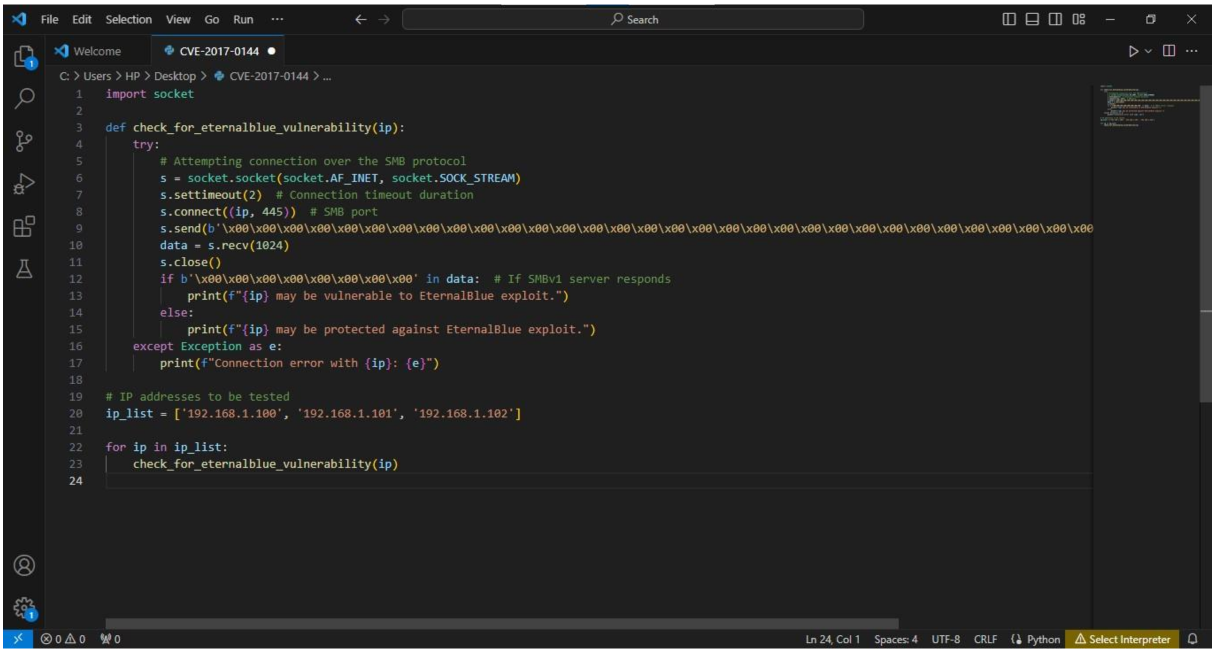
```python
import subprocess

def check_for_updates():
    try:
        # Suitable commands for checking updates on Linux systems
        result = subprocess.run(["sudo", "apt", "list", "--upgradable"], capture_output=True, text=True)
        if result.returncode == 0:
            print("Updates checked successfully:")
            print(result.stdout)
        else:
            print("Failed to check updates.")
    except Exception as e:
        print("An error occurred while checking for updates:", e)

check_for_updates()
```

```
import subprocess


def check_for_updates():

    try:

        # Suitable commands for checking updates on Linux systems
        result = subprocess.run(["sudo", "apt", "list", "--upgradable"], capture_output=True,
text=True)
        if result.returncode == 0:
            print("Updates checked successfully:")
            print(result.stdout)
        else:
            print("Failed to check updates.")
    except Exception as e:
        print("An error occurred while checking for updates:", e)


check_for_updates()
```

The Python code provided can indeed be used to check the update status on Linux systems. If an update is available, a system administrator can then apply the necessary updates. This example demonstrates how cybersecurity measures can be taken based on CVE numbers. Similar measures can be taken for other security vulnerabilities and threats, which can help organizations improve their security posture.

For our next example, we will use Python to detect attacks against a security vulnerability known as CVE-2017-0144, also referred to as EternalBlue. This vulnerability, due to a flaw in the SMB protocol, was exploited during the WannaCry ransomware attack.

“We can perform a network scan in Python to detect affected systems. Here is an example:”


File Edit Selection View Go Run ···
Search
Welcome
CVE-2017-0144
C: > Users > HP > Desktop > CVE-2017-0144 > ...
Ln 24, Col 1 Spaces: 4 UTF-8 CRLF Python
Select Interpreter

```
import socket

def check_for_eternalblue_vulnerability(ip):
    try:
        # Attempting connection over the SMB protocol
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.settimeout(2)  # Connection timeout duration
        s.connect((ip, 445))  # SMB port
        s.send(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
        data = s.recv(1024)
        s.close()
        if b'\x00\x00\x00\x00\x00\x00\x00\x00' in data:  # If SMBv1 server responds
            print(f"{ip} may be vulnerable to EternalBlue exploit.")
        else:
            print(f"{ip} may be protected against EternalBlue exploit.")
    except Exception as e:
        print(f"Connection error with {ip}: {e}")

# IP addresses to be tested
ip_list = ['192.168.1.100', '192.168.1.101', '192.168.1.102']

for ip in ip_list:
    check_for_eternalblue_vulnerability(ip)
```

```
import socket
```

```
def check_for_eternalblue_vulnerability(ip):
    try:
        # Attempting connection over the SMB protocol
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.settimeout(2)  # Connection timeout duration
        s.connect((ip, 445))  # SMB port

s.send(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00')  # Testing SMBv1 protocol
        data = s.recv(1024)
        s.close()
        if b'\x00\x00\x00\x00\x00\x00\x00\x00' in data:  # If SMBv1 server responds
            print(f"{ip} may be vulnerable to EternalBlue exploit.")
        else:
            print(f"{ip} may be protected against EternalBlue exploit.")
    except Exception as e:
        print(f"Connection error with {ip}: {e}")


# IP addresses to be tested
ip_list = ['192.168.1.100', '192.168.1.101', '192.168.1.102']


for ip in ip_list:
    check_for_eternalblue_vulnerability(ip)
```

This Python code performs a test against a security vulnerability based on CVE-2017-0144 (EternalBlue) by connecting to the SMB protocol on a specific IP address. If the SMBv1 server responds, it indicates that it is vulnerable to the attack. Otherwise, it indicates that it is protected against the attack.

This example demonstrates how Python can be used to scan and detect systems for a security vulnerability based on a specific CVE number (CVE-2017-0144). Such scans can be used by organizations to update their systems and protect against attacks.

3.3.Enumeration and Definition of Types of Cyber Attacks

The atomic number represents the number of protons in the nucleus of an element. The atomic number is a fundamental property that determines the chemical characteristics of an element. Each element has a unique atomic number, which is a primary criterion for the arrangement of elements in the Periodic Table.

The atomic number is usually written as a small number to the upper left or bottom of an element's symbol. For example, hydrogen has an atomic number of 1, so the hydrogen symbol "H" can be written with the atomic number as "1H".

The atomic number indicates the number of protons in the nucleus of the element, and therefore, it also determines the basic electron configuration of the element. For instance, hydrogen has one proton and one electron, hence the atomic number of hydrogen is 1. Carbon has an atomic number of 6 because there are six protons in the nucleus of carbon.

The atomic number is a fundamental property that determines the chemical behavior and placement of an element. For example, in the Periodic Table, elements arranged in increasing atomic number order are grouped in families with similar chemical properties. Therefore, the atomic number is a fundamental tool for the classification of elements and understanding their chemical behaviors.

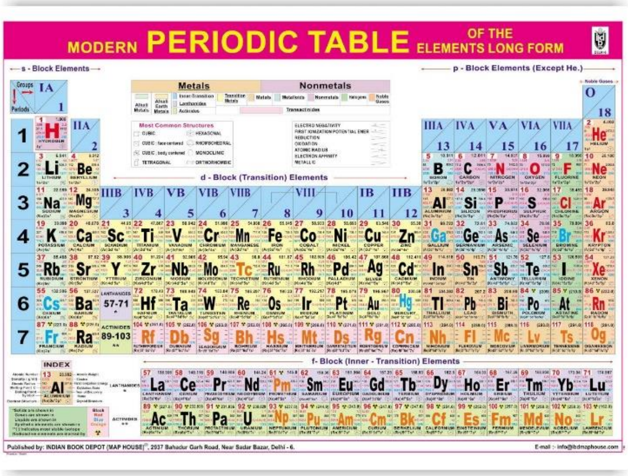

MODERN PERIODIC TABLE OF THE ELEMENTS LONG FORM
s - Block Elements
p - Block Elements (Except He.)
d - Block (Transition) Elements
f- Block (Inner - Transition) Elements
Metals
Nonmetals
Most Common Structures
INDEX
Published by: INDIAN BOOK DEPOT (MAP HOUSE), 2937 Bahadur Garh Road, Near Sadar Bazar, Delhi - 6.
E-mail :- info@ibdmaphouse.com

Table 1: Periodic Table Chart

The abbreviated symbolic names of elements are usually derived from the first letters of their name in English or Latin. These symbols are standardized to represent any language and are the same worldwide. Here are a few examples of how the symbolic names of some elements are derived:

Hydrogen: **H** The first letter of its English name is taken as "H".

Carbon: **C** The first letter of its English name is taken as "C".

Oxygen: **O** The first letter of its English name is taken as "O".

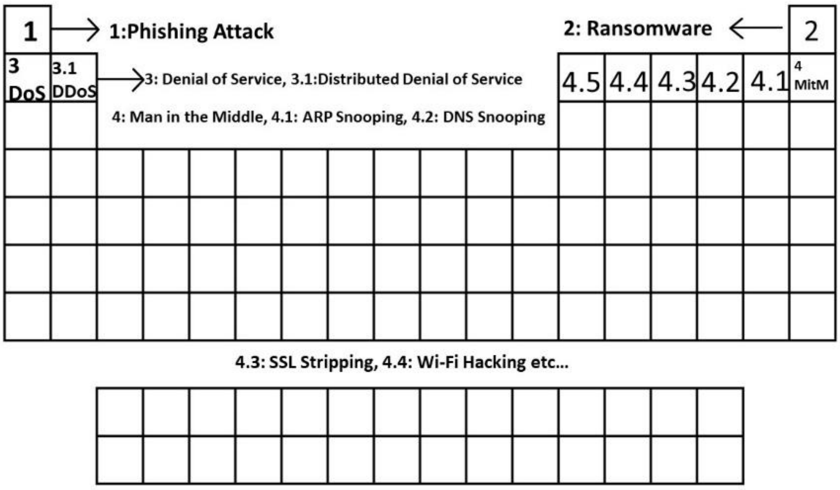

1
1:Phishing Attack
2: Ransomware
2
3
DoS
3.1
DDoS
3: Denial of Service, 3.1:Distributed Denial of Service
4: Man in the Middle, 4.1: ARP Snooping, 4.2: DNS Snooping
4.5
4.4
4.3
4.2
4.1
4
MitM
4.3: SSL Stripping, 4.4: Wi-Fi Hacking etc...

Table 2: Cyber Attack Table

This table is modeled after the chemical periodic table and serves as an example and representation for classifying cyber attacks. Each number corresponds to a different type of cyber attack and is grouped accordingly. It is proposed as an alternative to CVE codes. Considering that cyber security attacks are constantly evolving and mutating, this means that attackers are continually developing new methods to overcome new defense measures. Cyber attackers use advancements in technology and vulnerabilities in security measures to constantly change and adapt their attack strategies. Accordingly, each numerical range and the ongoing types of attack types represent the advanced process of this mutated or evolved type of cyber attack. (e.g., 4: Man in the Middle, 4.1: ARP Snooping, 4.2: DNS Snooping, 4.3: SSL Stripping, 4.4: Wi-Fi Hacking, etc.)

3.4.Cyber Security Perspective: The "Fight, Flight, Freeze" Mechanism Based on the phrase

"A chain is only as strong as its weakest link, and the weakest link is often the human," an action plan has been developed from a security chain perspective. The types of attacks that are numbered and classified will activate the action plan when they exhibit their unique symptoms in the system. Just like in the amygdala, it will implement a defense mechanism in the form of "Fight, Flight, Freeze."

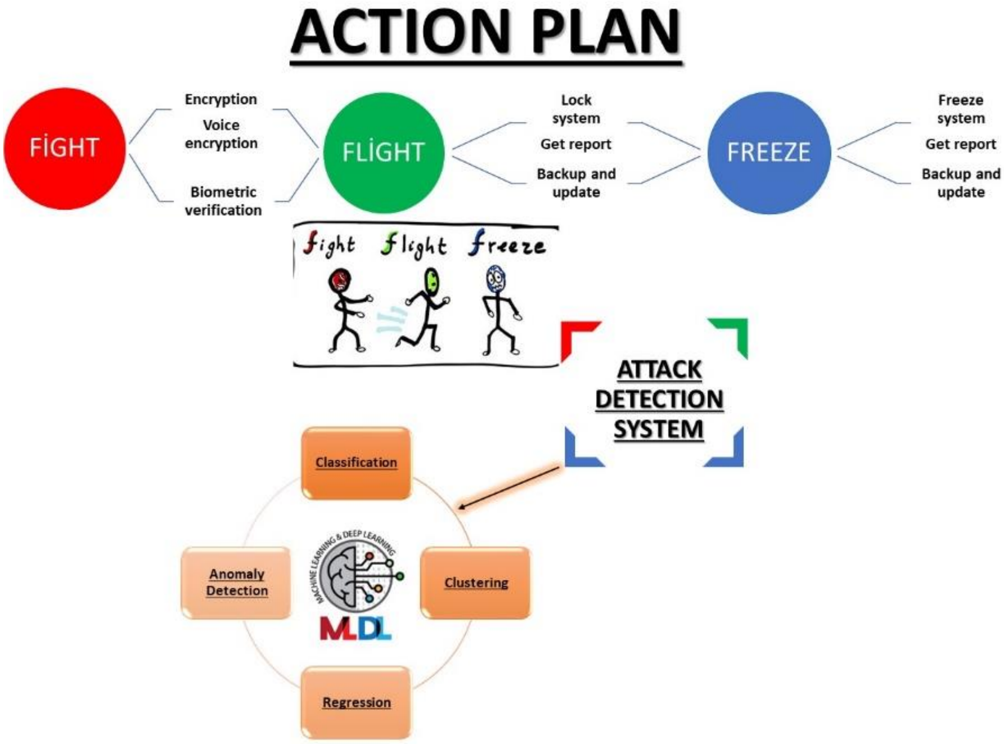


Figure 6: "Fight, Flight, Freeze" interactive action plan in terms of cyber security

"FIGHT"

The "Fight" command ensures security by responding to attacks through encryption, voice-generated encryption, or biometric identification. Encryption is a widely used method to secure communication and protect data. Encrypting data during transmission or storage is crucial for maintaining data integrity and confidentiality.

Voice encryption is a technique used to encrypt audio files in communication to ensure cyber security. This method can be used to protect the privacy of communication and prevent unauthorized access. Here are some details on how voice encryption is used to ensure cyber security:

- **Privacy**: Voice encryption protects communication by encrypting it, preventing third parties from listening or monitoring attempts. This helps safeguard sensitive information and enhances privacy.
- **Secure Data Transfer**: Voice encryption can use encryption software to securely transmit sound insulation. This prevents data from being compromised or manipulated during transfer.
- **Unauthorized Access Prevention**: An encrypted voice connection prevents unauthorized users or systems from being activated. Encrypted data allows access only to those with the correct key or authentication information.
- **Two-Factor Authentication**: Voice encryption can be used alongside additional security measures such as two-factor authentication for encrypting sequential data. This increases security and controls access permissions.

- **Reliability and Integrity**: Encrypted voice transmission ensures the protection and integrity of data. Encryption prevents alterations, ensuring accurate and reliable communication.

Voice encryption plays a significant role in cyber security and increases the volume of communication. However, it should not be used alone to create an effective security strategy. It should be integrated with other security controls. This is an important part of creating a comprehensive security strategy. Voice encryption is a significant tool to increase communication volume, but it is hosted alongside other security layers to provide full-scale protection against cyber threats (Mobayen, S., Volos, C., Çavuşoğlu, Ü., & S. Kaçar, S., 2020).

Biometric identifications are available for restricting access to specific areas or systems. For example, biometric readers in a workplace can provide authorized access only to certain rooms or computers. They can be used to monitor and detect security incidents. For instance, an unexpected biometric recognition failure in a system could indicate a potential unauthorized access attempt. In terms of multi-factor authentication, it involves using multiple methods of authentication. Users can verify their identity using a scan or biometric scan received via SMS after executing their passwords (ŞAMLI, R., & Yüksel, M. E. (2009)).

“FLIGHT”

When the system encounters an unidentified cyber attack, the “FLIGHT” command locks the system. After generating a report, the system and data are backed up and updated. The reporting process involves:

- **Reinforcement Learning**: This machine learning approach allows a computer to learn from experience. The system develops capabilities based on feedback within a specific service. To use reinforcement learning for detecting unknown cyber attacks, the following steps are followed:

    - **Data Collection**: The first step is gathering a broad dataset that includes various cyber attack types and normal traffic patterns. This dataset should contain network disruptions, system logs, and other relevant data from during attacks.
    - **Labeling**: The collected data is labeled as normal or attack. This helps the learning process to recognize the development of different attack types.
    - **Model Training**: The model is personalized using a training system. It learns the features and duration used to identify a specific attack.
    - **Feedback**: The model receives feedback from the dataset and adjusts its behavior accordingly. Feedback is provided to improve the model’s accuracy when spreading attacks or producing false alarms.
    - **Continuous Updating**: The system must be continuously updated to enhance new attack models and capabilities. As new attack types emerge or attackers’ tactics change, the adaptability of the model is crucial.
    - **Performance Evaluation**: The model is continuously evaluated, and metrics such as false positive and false negative rates are used to assess its performance.

  - **Integration and Implementation**: The trained model is integrated in real-time and begins detecting cyber attacks by analyzing live data measurements. The alarms generated by the model are verified for reliability by the cyber security team or automated response systems.
  - **Advanced Analysis and Improvement**: The model's performance is constantly monitored and improved. This includes refining features, enhancing learning, and reducing false alarm rates.

Following this data and the reinforcement learning process, when unidentified attack types exhibit similar characteristics, they can now be classified and added to the table. Additionally, once they are identifiable, they can be addressed with the "FIGHT" command directive in response to the cyber attack.

"FREEZE"

When an unidentified cyber attack is encountered, the "FREEZE" command within the system is activated, freezing the system. After a report is generated, the system and data are backed up and updated. If no clue to the attack is captured in this command flow, the system will undergo a process involving machine learning and deep learning.

In cyber security, artificial intelligence techniques such as machine learning and deep learning are used for threat detection, attack detection, vulnerability analysis, and security incident response. Here are the basic types of machine learning and deep learning used in cyber security and how they work:

- **Classification**: Classification is a machine learning technique used to divide data into different classes. In cyber security, classification algorithms can be used for malware detection, malicious URL detection, and anomaly detection in user behaviors.
- **Clustering**: Clustering is a machine learning technique that groups data based on similar characteristics. In cyber security, clustering algorithms can be used for network traffic analysis, detection of attack clusters, and identification of anomalies.
- **Regression**: Regression is a machine learning technique used to model the relationship between a dependent variable and one or more independent variables. In cyber security, regression analysis can be used to predict the timing of cyber attacks and the impact of security incidents.
- **Anomaly Detection**: Anomaly detection is a machine learning technique used to identify deviations from normal behavior. In cyber security, anomaly detection algorithms can be used to detect abnormalities in network traffic, changes in user behaviors, and identification of security incidents.
- **Deep Learning**: Deep learning is an artificial intelligence technique that uses the multi-layered structure of artificial neural networks to model complex relationships. In cyber security, deep learning algorithms can be used for image-based malware detection, spam filtering, and natural language processing-based attack detection.

These techniques are used in cyber security for data analysis, threat detection, and security incident response. Machine learning and deep learning techniques can identify complex patterns based on large amounts of data and provide more effective protection against cyber threats (Özgür, SB, 2021).

Despite all these techniques, when faced with an unidentified cyber attack, the report taken is forwarded to the relevant institutions, organizations, or communities involved in cyber security.

Additionally, remote monitoring and penetration tests are also of great importance in terms of cyber security. Remote monitoring and tracking are crucial for detecting potential threats and implementing security measures in computer networks and systems. This process can be performed using various methods and tools.

Remote monitoring and tracking can involve recording network connections, analyzing system logs, storing security incidents, and using models of firewalls and other security devices. Advanced technologies such as artificial intelligence and machine learning development are also available for monitoring cyber security events and detecting threats.

Remote monitoring and tracking are part of the preventive, detective, and responsive measures in cyber security. Preventively, it allows for the identification of potential threats and the implementation of measures. As a regulatory measure, it helps identify abnormalities or potential threat indicators in the system. Responsively, it enables a quick response to identified threats and their mitigation.

In conclusion, remote monitoring and tracking are significant elements of cyber security and form an essential part of an organization's cyber security strategy. This process plays a critical role in detecting potential threats, preventing malicious activities, and ensuring the integrity of networks and systems.

Penetration tests are controlled attacks and tests conducted to identify and assess security vulnerabilities in computer systems or network infrastructure. These tests help determine the effectiveness of an organization's cyber security defenses. Here are the types of penetration tests and how they work:

Network Penetration Tests:

Network penetration tests are conducted to identify security vulnerabilities within an organization's network infrastructure. Penetration testers target servers, network devices, and other components within the network to perform vulnerability scans and execute attacks. These tests utilize techniques such as port scanning, vulnerability scanning, network traffic analysis, and attack simulation.

Application Penetration Tests: Application penetration tests aim to detect security vulnerabilities in an organization's web applications, mobile applications, and other software applications. Penetration testers use vulnerability scanning, code review, and attack simulation techniques to identify security weaknesses and protect against malicious attacks.

Physical Penetration Tests: Physical penetration tests evaluate an organization's physical security measures. Testers attempt to bypass security controls to gain physical access to facilities and test physical security measures to identify vulnerabilities.

Social Engineering Tests: Social engineering tests are designed to identify security vulnerabilities related to an organization's human resources. Penetration testers interact with users to try to obtain sensitive information or attempt to violate security policies.

Wireless Network Penetration Tests: Wireless network penetration tests are performed to detect security vulnerabilities in an organization's wireless networks. Penetration testers target wireless access points to conduct vulnerability scans, test encryption security, and analyze network traffic.

Penetration tests are a vital tool for strengthening organizations' cyber security strategies and identifying vulnerabilities. These tests help organizations become more resilient against cyber attacks by detecting security weaknesses.

In the context of the relationship between penetration tests and machine learning and deep learning, these AI techniques can play a significant role in the data analysis and threat detection processes used in penetration tests. Here are some aspects of this relationship:

Data Analysis and Anomaly Detection: Machine learning and deep learning techniques can be used to analyze data obtained from penetration tests and detect anomalies. For example, network traffic data collected during penetration tests can be analyzed using machine learning algorithms to identify potential attacks or abnormalities.

Vulnerability Scanning and Attack Detection: Machine learning and deep learning techniques can be utilized in vulnerability scanning and attack detection processes. Data obtained from penetration tests can be processed using machine learning algorithms to determine security vulnerabilities in the system or detect potential attacks.

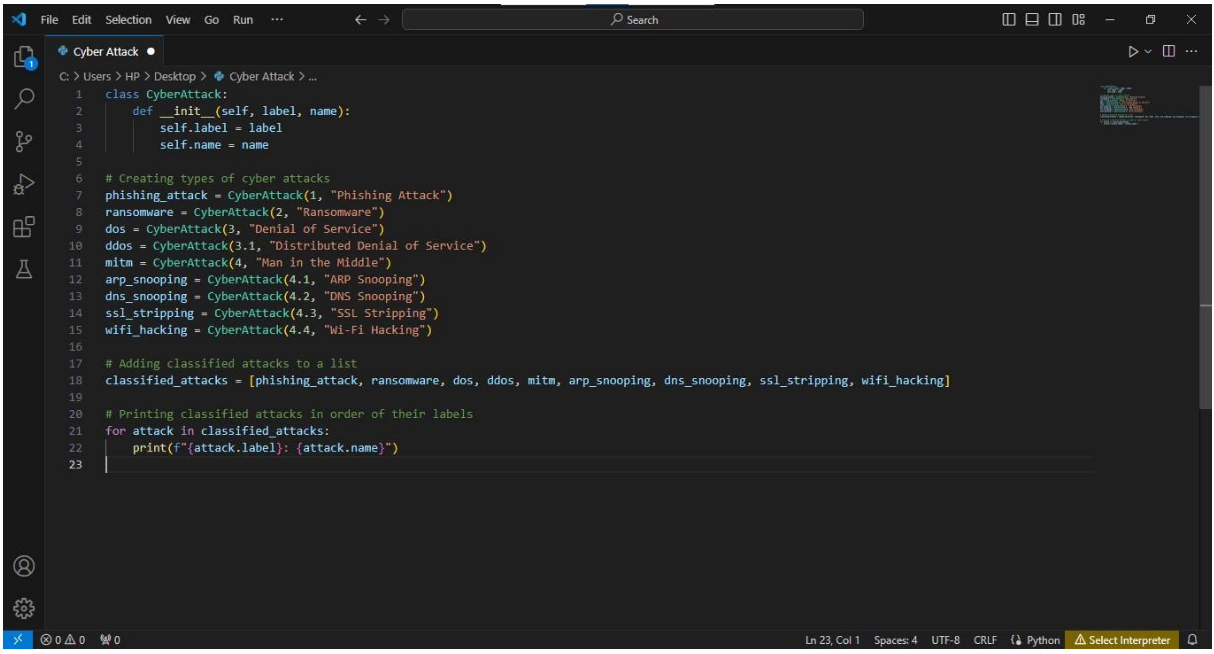
Attack Simulation and Threat Modeling: Machine learning and deep learning techniques can be used for simulating attacks and modeling threats in penetration tests. These techniques can be employed to strengthen organizations' defense mechanisms and prepare for potential attacks.

Security Incident Monitoring and Response: Machine learning and deep learning techniques can be applied to monitor and respond to security incident data obtained from penetration tests. These techniques can assist organizations in quickly and effectively responding to security incidents.

For these reasons, machine learning and deep learning techniques can help make penetration tests more effective and efficient, aiding in the identification of cyber security vulnerabilities. These techniques play a crucial role in data analysis, threat detection, and strengthening defense mechanisms in the field of cyber security (Vural, Y., 2007).

4. Method and Evaluation:

Artificial Neural Networks (ANNs) can detect signs of attacks by continuously monitoring network details and system logs, and they can automatically block attacks and be used to detect and open malicious software. By learning the behaviors and characteristics of malware, they can be effective in detecting unknown malicious software. They can be used for identity verification and access control by monitoring user information, modeling normal user operations, and detecting abnormalities to identify unauthorized access attempts. They can be used in the development of security protocols and encryption processes. They can be used to create encryption keys, optimize firewalls, and establish secure communication. Following this statement, the amygdala's unique "fight, flight, freeze" mechanism is expressed in simplified language as a defense method against cyber attacks.

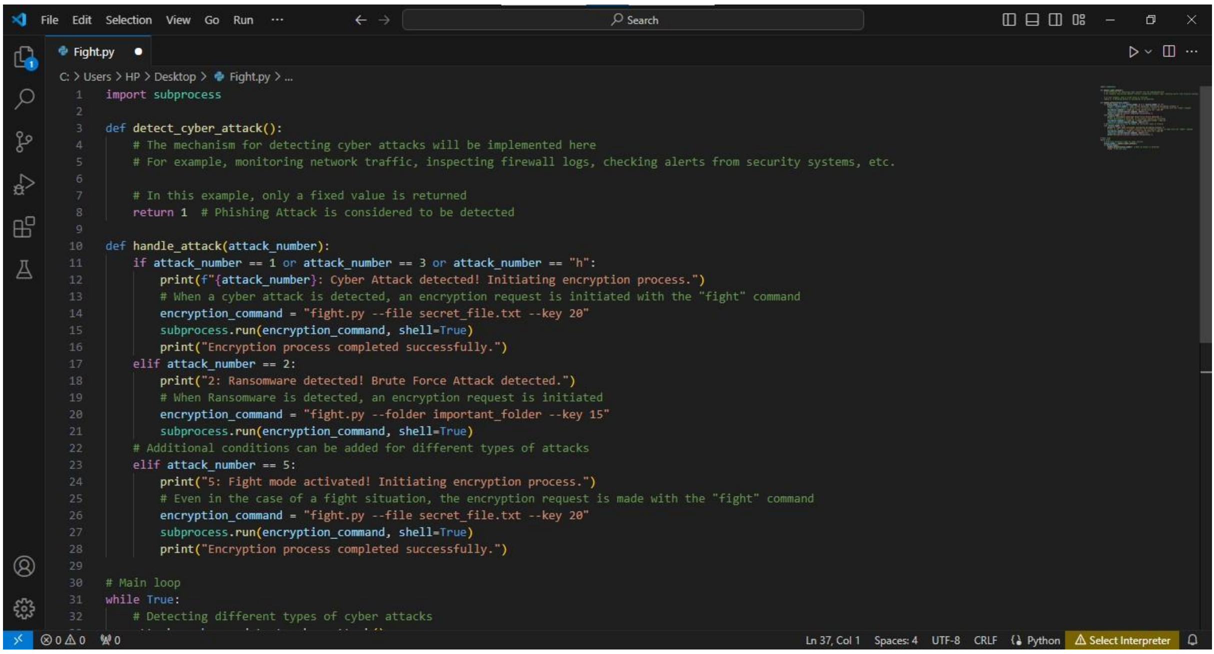


```python
class CyberAttack:
    def __init__(self, label, name):
        self.label = label
        self.name = name

# Creating types of cyber attacks
phishing_attack = CyberAttack(1, "Phishing Attack")
ransomware = CyberAttack(2, "Ransomware")
dos = CyberAttack(3, "Denial of Service")
ddos = CyberAttack(3.1, "Distributed Denial of Service")
mitm = CyberAttack(4, "Man in the Middle")
arp_snooping = CyberAttack(4.1, "ARP Snooping")
dns_snooping = CyberAttack(4.2, "DNS Snooping")
ssl_stripping = CyberAttack(4.3, "SSL Stripping")
wifi_hacking = CyberAttack(4.4, "Wi-Fi Hacking")

# Adding classified attacks to a list
classified_attacks = [phishing_attack, ransomware, dos, ddos, mitm, arp_snooping, dns_snooping, ssl_stripping, wifi_hacking]

# Printing classified attacks in order of their labels
for attack in classified_attacks:
    print(f"{attack.label}: {attack.name}")

```

Utilizing a cyber attack table to label the collected dataset as attacks aids in the recognition of the development of different types of learning. Labeling the dataset collected using the cyber attack table as attacks is a crucial step in training machine learning models and recognizing various types of attacks. This process ensures the model is trained correctly and can more effectively detect new or unknown types of attacks. Labeled data plays a critical role in the learning process of the model and enhances its sensitivity to real-world scenarios. Consequently, cyber security systems can detect attacks more quickly and accurately, enabling the activation of appropriate defense mechanisms.

“FIGHT” Defense Mechanism

The “Fight” command ensures security by responding to attacks through encryption, voice-generated encryption, or biometric identification. Encryption is a commonly used method to secure communication and protect data. Particularly during the transfer or storage of data, encrypting the data is crucial for maintaining data integrity and confidentiality.

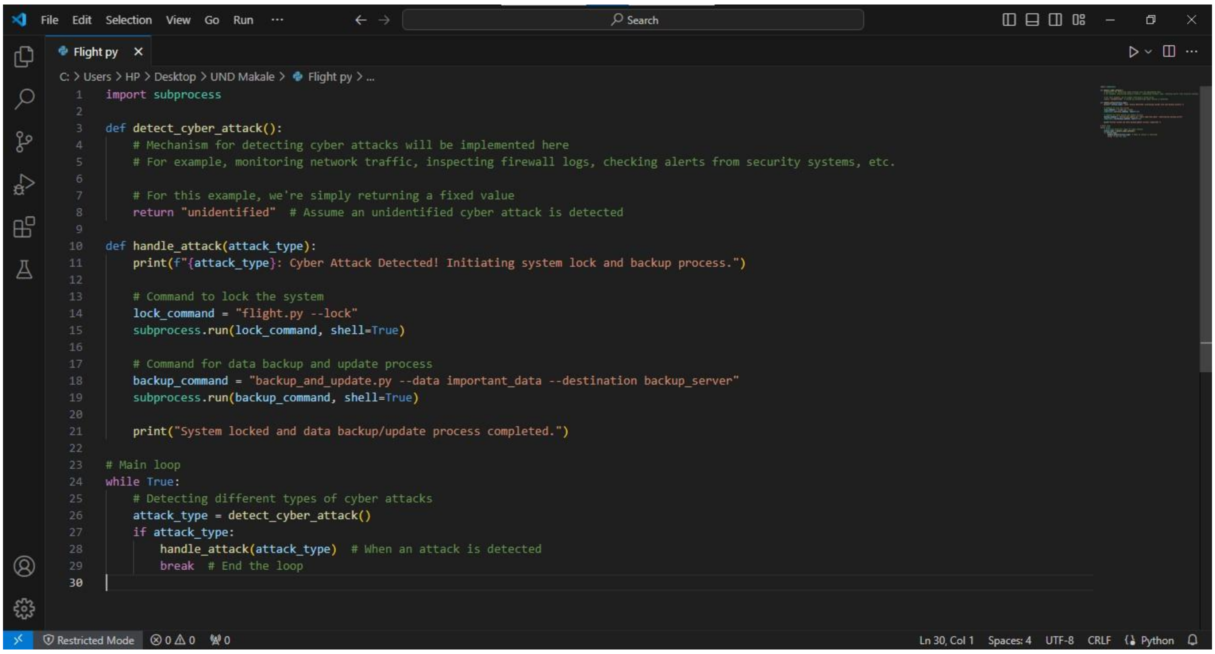


```python
import subprocess

def detect_cyber_attack():
    # The mechanism for detecting cyber attacks will be implemented here
    # For example, monitoring network traffic, inspecting firewall logs, checking alerts from security systems, etc.

    # In this example, only a fixed value is returned
    return 1  # Phishing Attack is considered to be detected

def handle_attack(attack_number):
    if attack_number == 1 or attack_number == 3 or attack_number == "h":
        print(f"{attack_number}: Cyber Attack detected! Initiating encryption process.")
        # When a cyber attack is detected, an encryption request is initiated with the "fight" command
        encryption_command = "fight.py --file secret_file.txt --key 20"
        subprocess.run(encryption_command, shell=True)
        print("Encryption process completed successfully.")
    elif attack_number == 2:
        print("2: Ransomware detected! Brute Force Attack detected.")
        # When Ransomware is detected, an encryption request is initiated
        encryption_command = "fight.py --folder important_folder --key 15"
        subprocess.run(encryption_command, shell=True)
    # Additional conditions can be added for different types of attacks
    elif attack_number == 5:
        print("5: Fight mode activated! Initiating encryption process.")
        # Even in the case of a fight situation, the encryption request is made with the "fight" command
        encryption_command = "fight.py --file secret_file.txt --key 20"
        subprocess.run(encryption_command, shell=True)
        print("Encryption process completed successfully.")

# Main loop
while True:
    # Detecting different types of cyber attacks
```

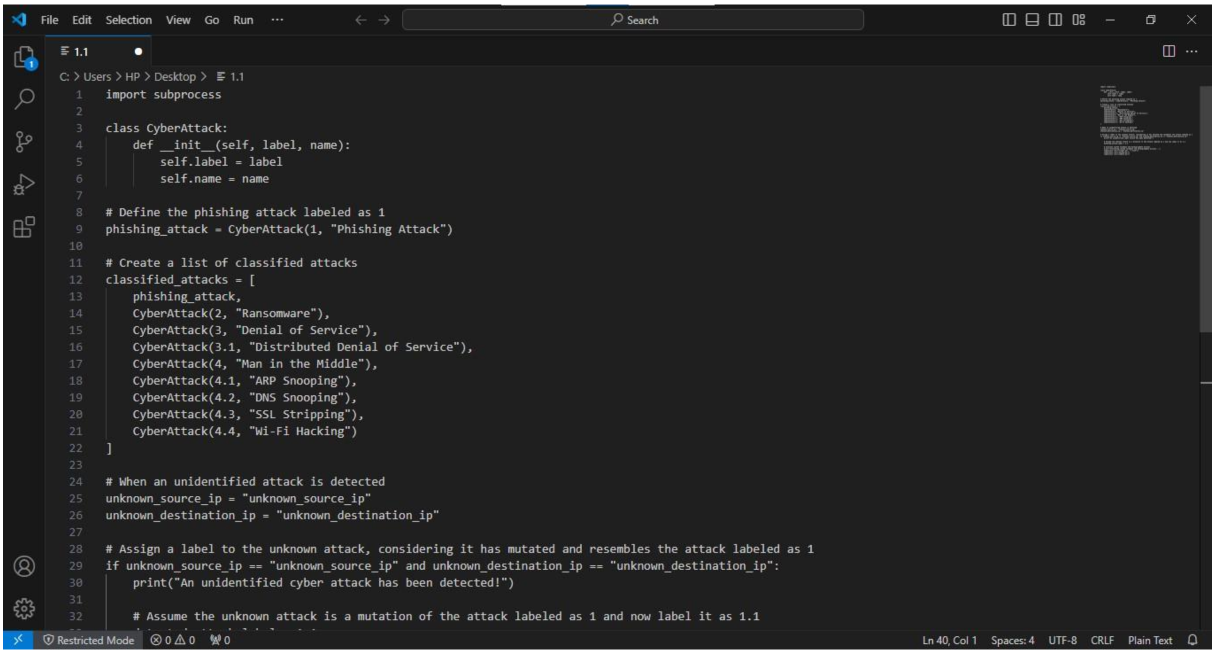
```python
# Main loop
while True:
    # Detecting different types of cyber attacks
    attack_number = detect_cyber_attack()
    if attack_number:
        handle_attack(attack_number)  # When an attack is detected
        break  # End the loop
```

"FLIGHT" Defense Mechanism

When faced with an unidentified cyber attack, we activated the "FLIGHT" command in the system to lock it down. After generating a report, we proceeded with the backup and update process to ensure the system's security and data integrity.



```python
import subprocess

def detect_cyber_attack():
    # Mechanism for detecting cyber attacks will be implemented here
    # For example, monitoring network traffic, inspecting firewall logs, checking alerts from security systems, etc.

    # For this example, we're simply returning a fixed value
    return "unidentified"  # Assume an unidentified cyber attack is detected

def handle_attack(attack_type):
    print(f"{attack_type}: Cyber Attack Detected! Initiating system lock and backup process.")

    # Command to lock the system
    lock_command = "flight.py --lock"
    subprocess.run(lock_command, shell=True)

    # Command for data backup and update process
    backup_command = "backup_and_update.py --data important_data --destination backup_server"
    subprocess.run(backup_command, shell=True)

    print("System locked and data backup/update process completed.")

# Main loop
while True:
    # Detecting different types of cyber attacks
    attack_type = detect_cyber_attack()
    if attack_type:
        handle_attack(attack_type)  # When an attack is detected
        break  # End the loop

```

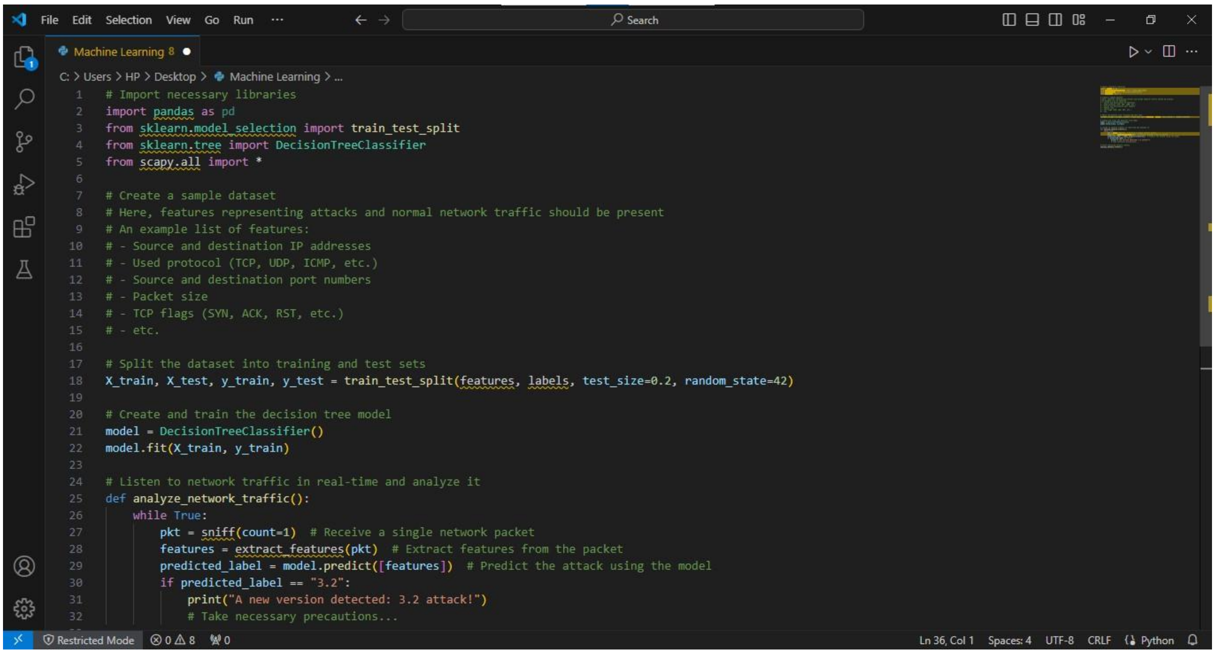


File Edit Selection View Go Run ···
Search
1.1
C: > Users > HP > Desktop > 1.1
Restricted Mode ⊗ 0 ⚠ 0 0
Ln 40, Col 1 Spaces: 4 UTF-8 CRLF Plain Text

```
import subprocess

class CyberAttack:
    def __init__(self, label, name):
        self.label = label
        self.name = name

# Define the phishing attack labeled as 1
phishing_attack = CyberAttack(1, "Phishing Attack")

# Create a list of classified attacks
classified_attacks = [
    phishing_attack,
    CyberAttack(2, "Ransomware"),
    CyberAttack(3, "Denial of Service"),
    CyberAttack(3.1, "Distributed Denial of Service"),
    CyberAttack(4, "Man in the Middle"),
    CyberAttack(4.1, "ARP Snooping"),
    CyberAttack(4.2, "DNS Snooping"),
    CyberAttack(4.3, "SSL Stripping"),
    CyberAttack(4.4, "Wi-Fi Hacking")
]

# When an unidentified attack is detected
unknown_source_ip = "unknown_source_ip"
unknown_destination_ip = "unknown_destination_ip"

# Assign a label to the unknown attack, considering it has mutated and resembles the attack labeled as 1
if unknown_source_ip == "unknown_source_ip" and unknown_destination_ip == "unknown_destination_ip":
    print("An unidentified cyber attack has been detected!")

    # Assume the unknown attack is a mutation of the attack labeled as 1 and now label it as 1.1
```

```
# Assume the unknown attack is a mutation of the attack labeled as 1 and now label it as 1.1
    detected_attack_label = 1.1


    # Initiate system lockdown and backup/update process
    print("Initiating system lockdown and backup/update process...")
    subprocess.run(["flight.py", "--lock"])
    subprocess.run(["backup.py"])
    subprocess.run(["update.py"])
```

This code indicates that when an unidentified attack is detected, it resembles a phishing attack previously labeled as 1 and has mutated, thus it is now labeled as 1.1. Subsequently, the system is locked, and backup/update processes are initiated.



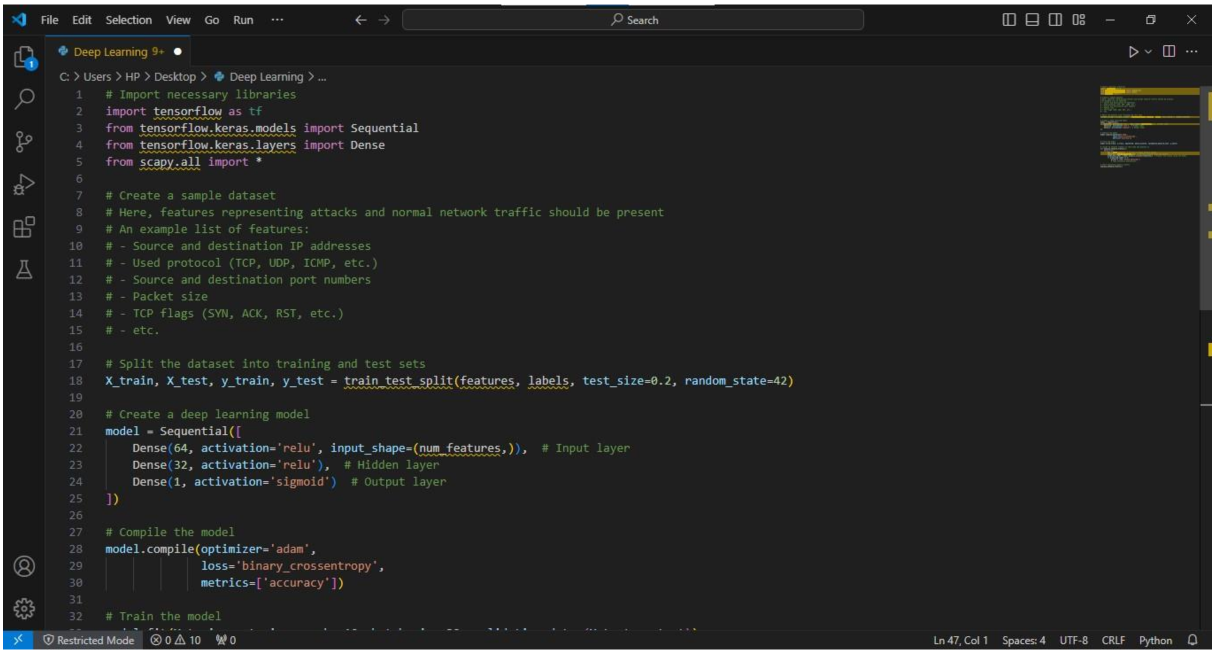
```python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from scapy.all import *

# Create a sample dataset
# Here, features representing attacks and normal network traffic should be present
# An example list of features:
# - Source and destination IP addresses
# - Used protocol (TCP, UDP, ICMP, etc.)
# - Source and destination port numbers
# - Packet size
# - TCP flags (SYN, ACK, RST, etc.)
# - etc.

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# Create and train the decision tree model
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Listen to network traffic in real-time and analyze it
def analyze_network_traffic():
    while True:
        pkt = sniff(count=1)  # Receive a single network packet
        features = extract_features(pkt)  # Extract features from the packet
        predicted_label = model.predict([features])  # Predict the attack using the model
        if predicted_label == "3.2":
            print("A new version detected: 3.2 attack!")
            # Take necessary precautions...
```

```python
# Start analyzing network traffic
analyze_network_traffic()
```

In this code example, the goal is to detect cyber attacks using a decision tree model. The processes are carried out through the following steps:

- **Dataset Creation**: A sample dataset is created. This dataset consists of features representing attacks and normal network traffic. Features include information such as IP addresses, protocols used, port numbers, packet size, and TCP flags.
- **Splitting the Dataset into Training and Testing Sets**: The created dataset is divided into training and testing sets. This step is important for training the model and evaluating its performance.
- **Creation and Training of the Decision Tree Model**: A decision tree model is created and trained using the DecisionTreeClassifier class from the scikit-learn library. The model will be used to recognize attacks based on the training data.
- **Real-Time Network Traffic Analysis**: Within a while loop, real-time network traffic is monitored with the help of the scapy library. Features are extracted for each packet, and an attack prediction is made using the decision tree model.
- **Attack Detection and Processing**: If the predicted attack type is labeled “3.2”, an output stating “A new version detected: 3.2 attack!” is given. Subsequently, necessary measures can be taken.

This code example focuses on detecting a specific type of cyber attack using a decision tree model.



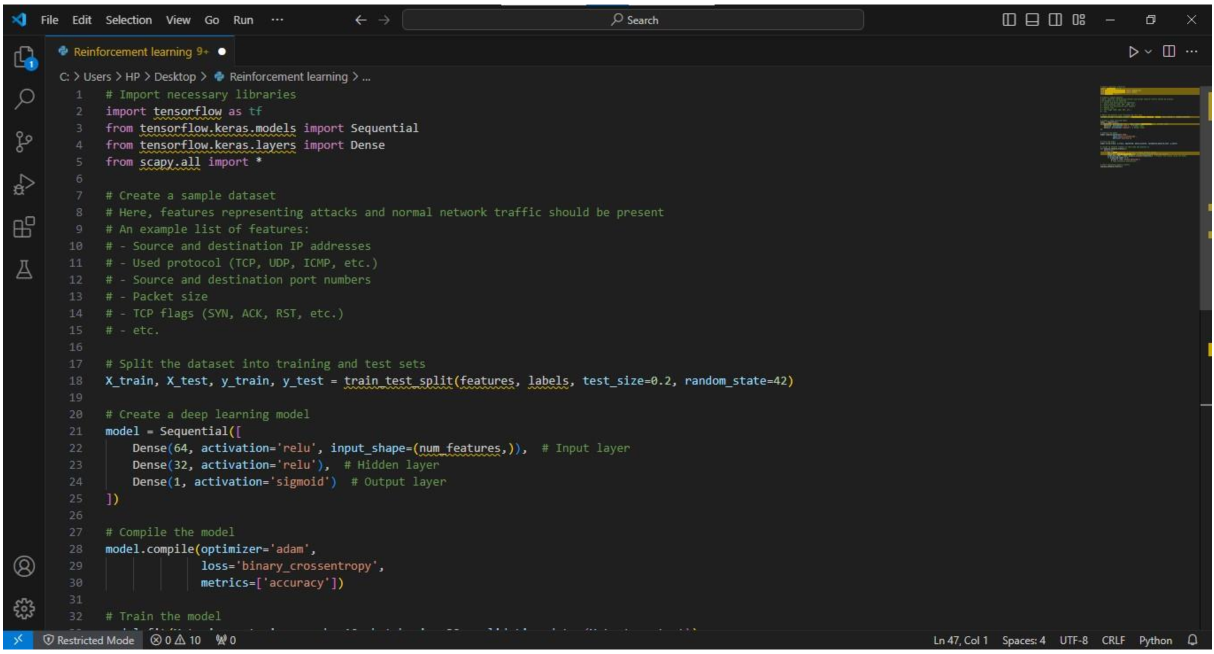
File Edit Selection View Go Run ···
Search
Deep Learning 9+
C: > Users > HP > Desktop > Deep Learning > ...
Restricted Mode ⊗ 0 ⚠ 10 0
Ln 47, Col 1 Spaces: 4 UTF-8 CRLF Python

```python
# Import necessary libraries
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from scapy.all import *

# Create a sample dataset
# Here, features representing attacks and normal network traffic should be present
# An example list of features:
# - Source and destination IP addresses
# - Used protocol (TCP, UDP, ICMP, etc.)
# - Source and destination port numbers
# - Packet size
# - TCP flags (SYN, ACK, RST, etc.)
# - etc.

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# Create a deep learning model
model = Sequential([
    Dense(64, activation='relu', input_shape=(num_features,)),  # Input layer
    Dense(32, activation='relu'),  # Hidden layer
    Dense(1, activation='sigmoid')  # Output layer
])

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Train the model
```

```python
# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

# Listen to network traffic in real-time and analyze it
def analyze_network_traffic():
    while True:
        pkt = sniff(count=1)  # Receive a single network packet
        features = extract_features(pkt)  # Extract features from the packet
        predicted_label = model.predict_classes([features])  # Predict the attack using the model
        if predicted_label == 1:
            print("A cyber attack detected!")
            # Take necessary precautions...

# Start analyzing network traffic
analyze_network_traffic()
```

In this code example, we aim to detect cyber attacks by monitoring network traffic using deep learning. The model has been pre-trained with training data and is then used to analyze network traffic in real-time to detect attacks. This is an example of how machine learning and deep learning can be utilized in the field of cyber security. We train the model using a pre-training dataset and then detect attacks by monitoring network traffic in real-time.

https://doi.org/10.33774/coe-2024-xszz9 **ORCID:** https://orcid.org/0009-0008-4985-7191 Content not peer-reviewed by Cambridge University Press. **License:** All Rights Reserved



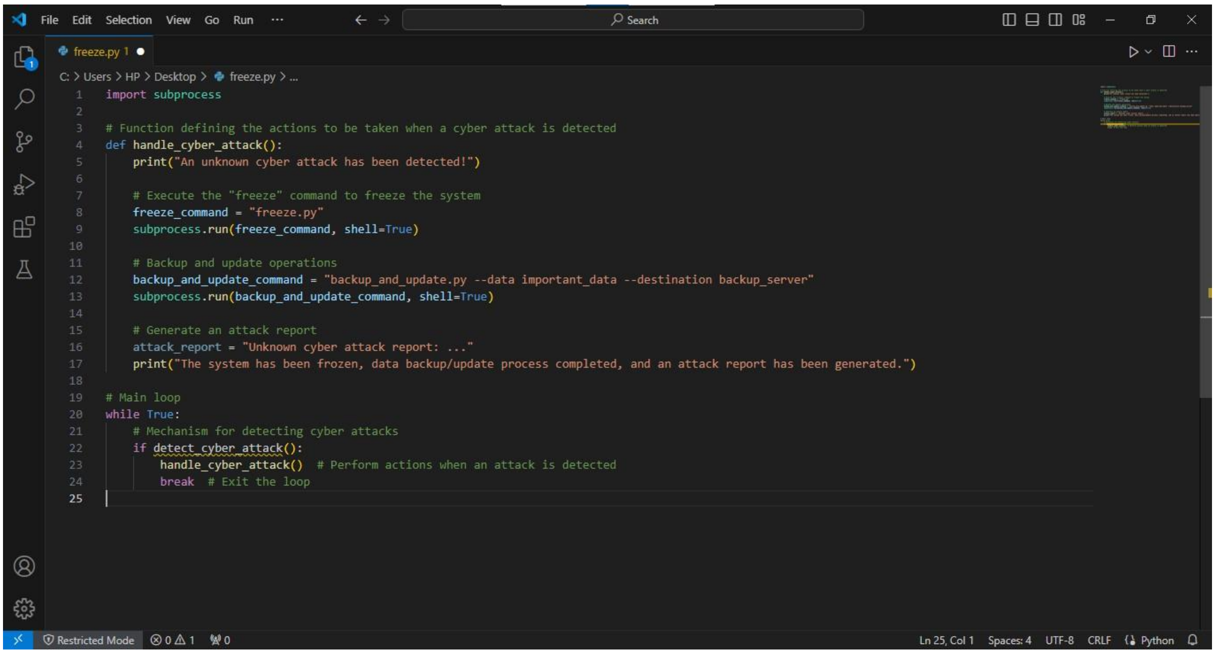
```python
# Import necessary libraries
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from scapy.all import *

# Create a sample dataset
# Here, features representing attacks and normal network traffic should be present
# An example list of features:
# - Source and destination IP addresses
# - Used protocol (TCP, UDP, ICMP, etc.)
# - Source and destination port numbers
# - Packet size
# - TCP flags (SYN, ACK, RST, etc.)
# - etc.

# Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# Create a deep learning model
model = Sequential([
    Dense(64, activation='relu', input_shape=(num_features,)),  # Input layer
    Dense(32, activation='relu'),  # Hidden layer
    Dense(1, activation='sigmoid')  # Output layer
])

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Train the model
```

```python
# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

# Listen to network traffic in real-time and analyze it
def analyze_network_traffic():
    while True:
        pkt = sniff(count=1)  # Receive a single network packet
        features = extract_features(pkt)  # Extract features from the packet
        predicted_label = model.predict_classes([features])  # Predict the attack using the model
        if predicted_label == 1:
            print("A cyber attack detected!")
            # Take necessary precautions...

# Start analyzing network traffic
analyze_network_traffic()
```

Reinforcement learning models can indeed be used to train an artificial intelligence component that monitors network traffic and detects attacks. Over time, as the model is trained with more data, it improves its ability to accurately detect attacks. This method allows the model to learn from its interactions with the environment, adapting its strategies for better detection and response to cyber threats. It’s a dynamic approach that evolves with the changing patterns of network traffic and potential security breaches.

“FREEZE” Defense Mechanism

In this code, the steps to be taken in the system in the event of a detected cyber attack are specified. First, we executed the “freeze” command to freeze the system, then we carried out data backup and update processes, and finally, we created an attack report. These actions are performed to ensure the security of the data in the system against the detected cyber attack and to report information related to the attack.



```python
import subprocess

# Function defining the actions to be taken when a cyber attack is detected
def handle_cyber_attack():
    print("An unknown cyber attack has been detected!")

    # Execute the "freeze" command to freeze the system
    freeze_command = "freeze.py"
    subprocess.run(freeze_command, shell=True)

    # Backup and update operations
    backup_and_update_command = "backup_and_update.py --data important_data --destination backup_server"
    subprocess.run(backup_and_update_command, shell=True)

    # Generate an attack report
    attack_report = "Unknown cyber attack report: ..."
    print("The system has been frozen, data backup/update process completed, and an attack report has been generated.")

# Main loop
while True:
    # Mechanism for detecting cyber attacks
    if detect_cyber_attack():
        handle_cyber_attack()  # Perform actions when an attack is detected
        break  # Exit the loop

```

Attack Report

Date: 2024-05-05
Type of Attack: Unidentified Cyber Attack
Description: The system experienced an unidentified cyber attack, which caused freezing and disrupted normal operations. The precise nature of the attack is unknown, but urgent measures were taken to secure the system.

Measures Taken:

- **System Lockdown**: A lockdown procedure was initiated to freeze the system upon detection of the attack, mitigating its impact.
- **Data Backup and Update**: Data was backed up and updated during the attack to prevent loss of important information.
- **Security Checks**: Detailed security checks were conducted on the system to better understand the cause and effects of the attack.
- **Attack Analysis**: Further information about the source and method of the attack was obtained through detailed examinations of the system.

Conclusion and Recommendations: This attack posed a significant risk to the system and disrupted its functioning. To prevent similar attacks in the future, it is recommended to strengthen network security measures and review cyber security policies.

This report provides detailed information about the detection of the attack, the measures taken, and the outcomes.

5. Conclusion and Discussion:

Artificial Neural Networks (ANNs) have the potential to revolutionize the field of cyber security. The complexity and constant evolution of today's cyber threats often render traditional security solutions inadequate. In this context, ANNs and deep learning techniques offer new and effective methods for detecting, analyzing, and preventing cyber attacks. Machine learning and ANN-based systems can detect malware and phishing attacks more quickly and accurately, thereby saving valuable time for cyber security teams.

In this study, the amygdala's fight, flight, and freeze responses have been modeled as defense mechanisms for cyber security systems. Each type of cyber attack is labeled with numbers attributed to these fundamental defense responses. This approach aims to make the responses to cyber threats more intuitive and rapid. The fight response represents systems actively defending against attacks; the flight response represents moving away from attack vectors to minimize potential damage; and the freeze response represents systems pausing in the face of uncertain and unidentified threats until more information is gathered.

Although the practical effectiveness of this model has not been tested on real-world data, it is theoretically considered to add a new dimension to cyber security strategies. It particularly has the potential to improve the adaptation of cyber security systems to the dynamic and constantly changing threat environment and make defense mechanisms more flexible.

While deep learning is recognized as an advanced tool in cyber security, the integration of natural defense mechanisms such as fight, flight, and freeze into this field has added a new dimension. However, the advantages and disadvantages of this approach should be considered and compared with other methods.

Advantages:

- **Rapid Learning**: Deep learning algorithms can quickly analyze large datasets and identify patterns, facilitating the effective learning of natural defense mechanisms like fight, flight, and freeze.
- **Automatic Response**: Mechanisms taught with deep learning can automatically respond when they detect specific threat patterns, providing fast and effective cyber defense without the need for human intervention.
- **Scalability**: Deep learning algorithms can be used in large-scale systems and are continuously improvable, allowing for the constant updating and enhancement of defense mechanisms.

Disadvantages:

- **False Positives**: Deep learning algorithms can sometimes misinterpret normal activities as threats, producing false positives, which can lead to unnecessary alarms and resource wastage.
- **Lack of Emotional Context**: Deep learning is limited in understanding emotional context, and thus may struggle to fully reflect emotional responses like fight, flight, and freeze.
- **Bias in Learning Data**: Deep learning algorithms can reflect biases and misunderstandings present in training data, which can reduce the effectiveness of defense mechanisms.

Future studies should evaluate how this model performs in real-time cyber security scenarios and how it could affect the decision-making processes of cyber security experts. Additionally, integrating this approach with different cyber security architectures and protocols could enhance the model's applicability and effectiveness. This work can be seen as an innovative step in the application of artificial intelligence and machine learning techniques in the field of cyber security and could lay the groundwork for further research in this area. Collaboration between humans and machines can provide a more effective defense in terms of cyber security.

References:

1. Öztemel, E. (2003). Artificial Neural Networks. Papatya Publishing, Istanbul.
2. ÖZTÜRK, E. N., & KARŞIDAĞ, Ç. The Place and Importance of Emotional Intelligence in Human Life. ISTANBUL GEDIK UNIVERSITY VOLUME 1-NUMBER 2-YEAR 2023, 332.
3. KayaY, A., & CoşkuntanE, İ. Interoception: Emotions and Neurophysiology.
4. Hekim, H., & BAŞIBÜYÜK, O. (2013). Cyber Crimes and Turkey's Cyber Security Policies. International Security and Terrorism Journal, 4(2), 135-158.
5. Seyyarer, E., & Ayata, F. (2023). The Era of Machine Learning in Cyber Security.
6. ARSLAN, M. E. CYBER SECURITY AND TYPES OF CYBER ATTACKS.
7. Akram, J. and Ping, L. (2020). How to create a vulnerability comparison to overcome cyber security attacks? IET Information Security, 14 (1), 60-71.
8. www.nvd.nist.gov
9. Mobayen, S., Volos, C., Çavuşoğlu, Ü., & S. Kaçar, S. (2020). A simple chaotic flow with hyperbolic sinusoidal function and its application to voice encryption. Symmetry, 12(12), 2047.
10. Yazğılı, E., & Baykara, M. (2021). Cyberbullying detection methods potential application areas and challenges. Dicle University Engineering Faculty Engineering Journal, 12(1), 23-35.
11. Özgür, SB (2021). Algorithms, Artificial Intelligence, Machine Learning, Deep Learning and Applications: A Comparison of Humanitarian Benefit Production by Software. Journal of Economics and Management Research, 10 (1), 1-29.
12. Vural, Y. (2007). Corporate Information Security and Penetration Testing (Master's thesis, Institute of Science).
13. www.yapayzekatr.com