**Zadania dla początkujących (20)**

1. **Hello World**:  
   Prosty program, który wypisuje na ekranie tekst „Hello, World!”. Jest to klasyczny pierwszy krok w nauce programowania.
2. **Podstawowe obliczenia**:  
   Program, który przyjmuje dwie liczby od użytkownika i wykonuje na nich podstawowe operacje matematyczne: dodawanie, odejmowanie, mnożenie i dzielenie.
3. **Sprawdzanie siły hasła**:  
   Skrypt sprawdzający, czy hasło jest wystarczająco silne, analizując jego długość i obecność różnych znaków (litery, cyfry, znaki specjalne).
4. **Szyfrator Cezara**:  
   Program implementujący szyfr Cezara, który przesuwa litery w alfabecie o określoną liczbę pozycji, umożliwiając szyfrowanie i deszyfrowanie wiadomości.
5. **BMI Kalkulator**:  
   Program obliczający BMI (wskaźnik masy ciała) na podstawie wagi i wzrostu użytkownika, wskazując, czy wynik mieści się w normie.
6. **Tabliczka mnożenia**:  
   Skrypt generujący tabliczkę mnożenia dla liczby podanej przez użytkownika, umożliwiający naukę matematyki.
7. **Konwerter jednostek**:  
   Program do konwertowania jednostek (np. z kilometrów na metry, stopni Celsjusza na Fahrenheita) na podstawie danych wejściowych.
8. **Obliczanie liczby pierwszych**:  
   Skrypt, który sprawdza, czy liczba wprowadzona przez użytkownika jest liczbą pierwszą.
9. **Operacje na plikach**:  
   Program, który tworzy nowy plik tekstowy, zapisuje do niego dane i umożliwia odczytanie jego zawartości.
10. **Prosty kalkulator**:  
    Program obsługujący podstawowe operacje matematyczne: dodawanie, odejmowanie, mnożenie, dzielenie, przyjmując dane od użytkownika.
11. **Zliczanie znaków**:  
    Skrypt, który liczy, ile razy w tekście występują litery, cyfry i inne znaki.
12. **Sortowanie liczb**:  
    Program, który sortuje listę liczb podaną przez użytkownika w porządku rosnącym lub malejącym.
13. **Zamiana temperatury**:  
    Program konwertujący temperaturę między różnymi skalami, np. Celsjusz, Fahrenheit, Kelvin.
14. **Znajdowanie średniej**:  
    Skrypt, który oblicza średnią arytmetyczną liczb wprowadzonych przez użytkownika.
15. **Konwersja czasu**:  
    Program, który konwertuje godziny, minuty i sekundy na jedną jednostkę (np. minuty lub sekundy).
16. **Prosty ping test**:  
    Skrypt, który sprawdza dostępność serwera lub urządzenia w sieci za pomocą polecenia ping.
17. **Liczby doskonałe**:  
    Program, który sprawdza, czy liczba wprowadzona przez użytkownika jest liczbą doskonałą (suma jej dzielników równa liczbie).
18. **Sortowanie słów**:  
    Skrypt, który sortuje słowa w ciągu tekstowym w porządku alfabetycznym.
19. **Obliczanie pola figury**:  
    Program, który oblicza pole prostokąta, trójkąta lub koła na podstawie danych podanych przez użytkownika.
20. **Licznik cyfr**:  
    Skrypt, który liczy liczbę cyfr w liczbie wprowadzonej przez użytkownika.

**Zadania dla średnio zaawansowanych (20)**

1. **Zaawansowany generator haseł**:  
   Skrypt, który generuje losowe hasła o określonej długości, zawierające małe i wielkie litery, cyfry oraz znaki specjalne, aby zapewnić ich bezpieczeństwo.
2. **Szyfrowanie AES**:  
   Program wykorzystujący algorytm AES do szyfrowania i deszyfrowania danych. Użytkownik podaje dane, które są szyfrowane kluczem symetrycznym, a następnie deszyfrowane przy użyciu tego samego klucza.
3. **Skaner portów**:  
   Skrypt skanujący wybrany adres IP w celu wykrycia otwartych portów, umożliwiający ocenę dostępności usług sieciowych na zdalnym hoście.
4. **Analiza logów serwera**:  
   Skrypt przetwarzający logi serwera (np. Apache, Nginx) i wykrywający potencjalnie niebezpieczne aktywności, takie jak wielokrotne próby logowania.
5. **Monitor katalogów**:  
   Skrypt, który monitoruje zmiany w folderze i wysyła powiadomienie, gdy jakikolwiek plik zostanie zmodyfikowany lub dodany.
6. **Tworzenie wykresów**:  
   Program używający biblioteki matplotlib do tworzenia wykresów na podstawie danych wprowadzonych przez użytkownika lub zapisanych w plikach CSV.
7. **Keylogger edukacyjny**:  
   Skrypt, który rejestruje naciśnięcia klawiszy w systemie (wyłącznie do celów edukacyjnych), pokazując, jak niebezpieczne mogą być tego typu narzędzia.
8. **Podmiana DNS**:  
   Program, który przechwytuje zapytania DNS na lokalnym serwerze i zmienia je w celu przeprowadzenia ataku typu man-in-the-middle lub analizy bezpieczeństwa.
9. **Zarządzanie hasłami**:  
   Skrypt przechowujący zaszyfrowane hasła użytkownika w bezpieczny sposób i umożliwiający ich odczyt w przypadku potrzeby.
10. **Symulacja kostki do gry**:  
    Program, który losowo generuje liczby w zakresie 1-6, symulując rzut kostką. Skrypt pozwala na wielokrotne rzuty i zlicza sumę wyników.
11. **Automatyzacja pobierania plików**:  
    Skrypt, który umożliwia masowe pobieranie plików z internetu na podstawie listy URL-i, co jest przydatne w zadaniach związanych z analizą danych.
12. **Analiza danych JSON**:  
    Program wczytujący dane zapisane w formacie JSON, analizujący je i generujący raport lub wykres przedstawiający istotne informacje.
13. **Tworzenie kalendarza**:  
    Skrypt generujący kalendarz na dany miesiąc lub rok, umożliwiający użytkownikowi przeglądanie dni tygodnia, dni roboczych i świąt.
14. **Prosty chatbot**:  
    Program, który odpowiada na zadane pytania użytkownika na podstawie predefiniowanych odpowiedzi. Skrypt może uczyć się nowych odpowiedzi z każdej rozmowy.
15. **Tworzenie notatnika**:  
    Aplikacja, która umożliwia zapis i edycję notatek w formie tekstowej. Skrypt zapisuje tekst w pliku i pozwala na jego późniejsze odczytywanie.
16. **Monitor sieci lokalnej**:  
    Skrypt wykrywający urządzenia podłączone do tej samej sieci lokalnej, analizujący ich adresy IP i MAC.
17. **Prosty system CAPTCHA**:  
    Program generujący proste testy CAPTCHA, które sprawdzają, czy użytkownik jest człowiekiem, czy botem. Skrypt generuje obrazki z losowymi znakami i cyframi.
18. **Analiza logów SSH**:  
    Skrypt analizujący logi SSH pod kątem podejrzanych działań, takich jak nieudane próby logowania, zmiany w konfiguracji lub przejścia do nowych urządzeń.
19. **Obliczanie czasu pracy**:  
    Program obliczający różnicę między godzinami rozpoczęcia i zakończenia pracy, umożliwiający monitorowanie czasu pracy pracowników.
20. **Tworzenie interfejsu GUI**:  
    Skrypt wykorzystujący bibliotekę tkinter do tworzenia prostych aplikacji z interfejsem graficznym, umożliwiający użytkownikowi interakcję z programem.

**Zaawansowane Zadania (20)**

1. **Firewall oparty na Pythonie**  
   Tworzy prosty filtr ruchu sieciowego na podstawie reguł definiowanych przez użytkownika, takich jak adresy IP, porty czy protokoły. Firewall wykorzystuje bibliotekę socket do monitorowania ruchu przychodzącego i wychodzącego. Pozwala na blokowanie podejrzanych połączeń i logowanie prób naruszeń. Użytkownik może konfigurować listę dozwolonych i zabronionych połączeń. Projekt wprowadza podstawy zarządzania ruchem sieciowym. Można go rozszerzyć o funkcje takie jak dynamiczne wykrywanie wzorców ruchu.
2. **System IDS (Intrusion Detection System)**  
   System do wykrywania podejrzanej aktywności w ruchu sieciowym. Wykorzystuje narzędzia takie jak scapy, by przechwytywać i analizować pakiety sieciowe. IDS porównuje ruch z bazą wzorców ataków, takich jak skanowanie portów czy ataki DoS. Program może być skonfigurowany do alarmowania administratora w czasie rzeczywistym o wykrytych problemach. Możliwość integracji z SIEM zwiększa jego efektywność. Projekt pozwala zrozumieć kluczowe zasady cyberbezpieczeństwa w sieci.
3. **VPN Kill-Switch**  
   Skrypt monitoruje połączenie z VPN i automatycznie blokuje dostęp do internetu, jeśli VPN zostanie przerwane. Wykorzystuje narzędzia takie jak psutil do monitorowania procesów i iptables do kontroli ruchu. Zapobiega przypadkowemu ujawnieniu danych użytkownika w przypadku utraty szyfrowanego tunelu. Może działać w tle jako usługa systemowa. Idealne narzędzie dla użytkowników VPN wymagających dodatkowej warstwy ochrony. Projekt jest praktycznym ćwiczeniem z zarządzania systemem i bezpieczeństwa sieciowego.
4. **Skaner podatności**  
   Wykrywa popularne luki bezpieczeństwa w systemach i aplikacjach, takie jak słabe hasła SSH czy niezabezpieczone serwery WWW. Korzysta z bibliotek nmap lub paramiko, aby przeprowadzać audyty bezpieczeństwa. Analizuje konfigurację systemu, otwarte porty i podatne usługi. Tworzy raport z wynikami skanowania oraz zaleceniami naprawczymi. Może być rozszerzony o integrację z platformami bezpieczeństwa, takimi jak Metasploit. Uczy zasad testowania bezpieczeństwa.
5. **Analizator phishingu**  
   Program do analizy podejrzanych e-maili i linków w celu wykrycia phishingu. Sprawdza domeny pod kątem wiarygodności, weryfikuje certyfikaty SSL i analizuje treść wiadomości. Używa API, takich jak VirusTotal, do identyfikacji zagrożeń. Może być wykorzystywany przez użytkowników indywidualnych oraz w ramach większych systemów bezpieczeństwa. Analiza wyników jest wyświetlana w przejrzystym raporcie. Projekt jest praktycznym wprowadzeniem do analizy zagrożeń cybernetycznych.
6. **Eksporter logów do SIEM**  
   Skrypt agreguje logi systemowe, aplikacyjne i sieciowe, a następnie przesyła je do systemu zarządzania informacją i zdarzeniami bezpieczeństwa (SIEM). Wykorzystuje biblioteki takie jak syslog i json do formatowania danych. Zapewnia centralizację danych z różnych źródeł, co pozwala na lepszą analizę bezpieczeństwa. Może integrować się z popularnymi narzędziami, takimi jak Splunk czy Elastic. Automatyzacja tego procesu jest kluczowa w zarządzaniu dużymi systemami IT. Projekt pozwala zrozumieć zarządzanie logami w środowiskach korporacyjnych.
7. **Zaawansowany skaner portów**  
   Identyfikuje otwarte porty oraz usługi na hostach w sieci, używając technik takich jak SYN scan czy UDP scan. Generuje szczegółowe raporty na temat uruchomionych usług, wersji oprogramowania i potencjalnych podatności. Może być używany jako narzędzie testowe podczas audytów bezpieczeństwa. Wprowadza użytkownika w bardziej zaawansowane techniki analizy ruchu sieciowego. Projekt jest idealny do nauki działania protokołów sieciowych.
8. **Monitorowanie ARP Spoofingu**  
   Skrypt wykrywa nieautoryzowane zmiany w tablicy ARP, które mogą wskazywać na ataki typu Man-In-The-Middle (MITM). Wykorzystuje scapy do monitorowania pakietów ARP i identyfikowania fałszywych odpowiedzi. W przypadku wykrycia ataku może automatycznie informować administratora. Projekt jest doskonałym przykładem zastosowania analizy ruchu sieciowego w celu zwiększenia bezpieczeństwa. Wprowadza do tematyki zabezpieczeń na poziomie warstwy łącza danych.
9. **Tworzenie honeypotów**  
   Implementacja pułapki na potencjalnych atakujących, która symuluje działanie rzeczywistego systemu. Może monitorować próby włamań, zbierać informacje o technikach używanych przez napastników oraz odciągać ich uwagę od prawdziwych systemów. Wykorzystuje narzędzia takie jak Honeyd lub Cowrie. Projekt dostarcza wiedzy na temat strategii obronnych i psychologii napastników. Wprowadza także podstawy analizy zachowań atakujących.
10. **Analizator logów sieciowych**  
    Program analizuje ruch zapisany w plikach PCAP przy użyciu narzędzi takich jak scapy czy pyshark. Wyciąga informacje o podejrzanych pakietach, takich jak powtórzone żądania, duży ruch do jednego hosta czy nieoczekiwane protokoły. Może generować raporty wizualizujące ruch w sieci. Projekt jest praktycznym przykładem pracy z danymi sieciowymi. Uczy zarządzania logami i analizy ruchu w kontekście cyberbezpieczeństwa.
11. **Zaawansowany system szyfrowania**  
    Skrypt szyfruje pliki przy użyciu asymetrycznych kluczy publicznych i prywatnych. Wykorzystuje bibliotekę cryptography do implementacji funkcji szyfrowania i deszyfrowania. Umożliwia użytkownikowi generowanie własnych par kluczy oraz zarządzanie kluczami publicznymi. System oferuje także opcję uwierzytelniania wiadomości za pomocą podpisów cyfrowych. Wprowadza użytkownika do kryptografii na poziomie produkcyjnym. Projekt może być rozszerzony o szyfrowanie strumieni danych w czasie rzeczywistym.
12. **Automatyczny backup danych**  
    Skrypt tworzy kopie zapasowe ważnych plików lub folderów w regularnych odstępach czasu. Wykorzystuje biblioteki takie jak os i shutil do kopiowania plików oraz schedule do automatyzacji zadań. Obsługuje opcje takich jak kompresja danych przed zapisaniem oraz przesyłanie kopii zapasowych na zdalny serwer. Projekt może być zintegrowany z systemami monitorowania, aby sprawdzać status backupów. Wprowadza użytkownika w zarządzanie danymi i automatyzację procesów IT.
13. **Symulacja botnetu**  
    Tworzy kontrolowane środowisko do symulacji zachowań botnetu, w którym kilka instancji programu komunikuje się ze sobą i wykonuje zadane polecenia. Skrypt może obsługiwać takie funkcje jak rozsyłanie wiadomości, wykonywanie określonych działań na zdalnych komputerach lub monitorowanie zasobów. Symulacja pozwala zrozumieć działanie botnetów oraz sposoby ich wykrywania. Projekt dostarcza praktycznego doświadczenia w analizie złożonych systemów rozproszonych i ich zabezpieczaniu.
14. **Detekcja botów w sieci**  
    Narzędzie analizuje ruch sieciowy, identyfikując nietypowe wzorce, które mogą wskazywać na obecność botów. Wykorzystuje techniki analizy statystycznej i uczenia maszynowego do klasyfikacji ruchu. Może wykrywać zjawiska takie jak masowe wysyłanie żądań do jednego hosta lub nieoczekiwane wzorce aktywności. Projekt wprowadza użytkownika do analizy danych w kontekście bezpieczeństwa sieciowego. Może być rozszerzony o integrację z systemami IDS lub SIEM.
15. **Analiza plików pod kątem malware**  
    Skrypt porównuje pliki z bazą znanych sygnatur wirusów, analizując zawartość pod kątem potencjalnych zagrożeń. Może korzystać z API takich jak VirusTotal lub implementować własne mechanizmy porównywania sygnatur. Projekt umożliwia także statyczną analizę kodu w plikach wykonywalnych w poszukiwaniu podejrzanych wzorców. Analiza jest raportowana w postaci szczegółowych wyników, które mogą być wykorzystane w dalszym procesie zabezpieczania systemu. Doskonałe wprowadzenie do analizy zagrożeń i inżynierii odwrotnej.
16. **Prosty system IDS**  
    Narzędzie do monitorowania ruchu sieciowego i identyfikacji typowych ataków, takich jak skanowanie portów czy próby logowania brute-force. Skrypt wykorzystuje bibliotekę scapy do przechwytywania pakietów oraz analizuje wzorce w ruchu sieciowym. W przypadku wykrycia podejrzanej aktywności może generować alarmy lub blokować źródło ataku. Projekt wprowadza użytkownika do podstawowych zasad działania systemów IDS. Może być rozbudowany o bardziej zaawansowane funkcje, takie jak wykrywanie ataków DDoS.
17. **System CAPTCHA oparty na obrazie**  
    Tworzy system CAPTCHA, który generuje obrazy z tekstem lub symbolami w celu weryfikacji użytkownika. Wykorzystuje bibliotekę Pillow do generowania obrazów oraz random do losowego umieszczania znaków. Może być używany w aplikacjach internetowych w celu ochrony przed botami. Projekt wprowadza użytkownika do pracy z grafiką w Pythonie oraz podstawami zabezpieczeń w interfejsach użytkownika. Może być rozbudowany o integrację z uczeniem maszynowym do rozpoznawania obrazów.
18. **Tworzenie reguł IPTables**  
    Skrypt automatyzuje konfigurację firewalla IPTables w systemie Linux, generując reguły blokowania lub dozwolenia ruchu na podstawie określonych kryteriów. Obsługuje konfigurację takich funkcji jak blokowanie określonych adresów IP, ograniczanie ruchu na wybranych portach lub tworzenie reguł NAT. Projekt wprowadza użytkownika w podstawy zarządzania systemami operacyjnymi i konfiguracji bezpieczeństwa na poziomie sieciowym. Może być wykorzystywany w scenariuszach produkcyjnych.
19. **Analiza danych Big Data**  
    Skrypt analizuje duże zbiory danych przy użyciu biblioteki pandas oraz narzędzi takich jak Dask do obliczeń równoległych. Może być używany do analizy logów systemowych, danych użytkowników lub ruchu sieciowego. Wprowadza użytkownika do pracy z dużymi zestawami danych, ich transformacją, agregacją oraz wizualizacją wyników. Projekt uczy optymalizacji kodu i zarządzania danymi w środowiskach wymagających dużych zasobów.
20. **Implementacja RSA**  
    Skrypt implementuje algorytm RSA, umożliwiając szyfrowanie i deszyfrowanie danych za pomocą kluczy publicznych i prywatnych. Generuje własne pary kluczy, zarządza ich przechowywaniem i używa ich do szyfrowania komunikacji między użytkownikami. Projekt wprowadza użytkownika do kryptografii asymetrycznej oraz zapewnia praktyczne doświadczenie z podstawami matematyki stosowanej. Można go rozszerzyć o funkcje takie jak podpisy cyfrowe i zabezpieczenia transmisji danych.

**Bardziej Zaawansowane Skrypty i Projekty**

1. **Analizator PCAP**  
   Narzędzie analizujące dane zapisane w plikach PCAP, które przechowują informacje o ruchu sieciowym. Skrypt wykorzystuje bibliotekę scapy lub pyshark do przechwytywania i interpretacji pakietów. Może identyfikować typy ruchu (HTTP, FTP, DNS), analizować źródła i cele komunikacji, a także wykrywać nietypowe aktywności. To narzędzie jest kluczowe dla analityków bezpieczeństwa zajmujących się analizą incydentów w czasie rzeczywistym. Może być rozszerzone o funkcje raportowania lub integrację z narzędziami SIEM.
2. **Zaawansowany IDS z SIEM**  
   Tworzy system detekcji intruzów (IDS) zintegrowany z SIEM, który analizuje ruch sieciowy i logi systemowe. Projekt pozwala na korelację zdarzeń między siecią a systemem, np. wykrycie anomalii w logowaniu powiązanej z podejrzanym ruchem. Wykorzystuje bibliotekę elastic-stack lub splunk-sdk. System automatycznie generuje alerty o podejrzanych działaniach i tworzy szczegółowe raporty. Jest przeznaczony dla zaawansowanych środowisk wymagających wysokiego poziomu bezpieczeństwa.
3. **Skrypt do wykrywania ransomware**  
   Monitoruje zmiany w plikach, takie jak masowe szyfrowanie, i wykrywa podejrzane działania. Korzysta z bibliotek takich jak watchdog do śledzenia systemu plików oraz hashlib do porównywania hashów plików. Może generować alerty w czasie rzeczywistym i wstrzymywać podejrzane procesy. Idealny do ochrony krytycznych systemów przed atakami ransomware. Projekt można rozbudować o integrację z systemami kopii zapasowych.
4. **Monitorowanie bezpieczeństwa serwerów**  
   Skrypt analizujący konfigurację serwerów (np. Apache, Nginx) w poszukiwaniu luk bezpieczeństwa. Wykorzystuje narzędzia takie jak sslscan czy nmap do analizy certyfikatów SSL, otwartych portów i błędnych ustawień. Generuje raporty zgodne z normami bezpieczeństwa, takimi jak OWASP czy PCI DSS. Może być używany do ciągłego monitorowania lub okresowych audytów. To narzędzie zapewnia lepszą widoczność podatności w infrastrukturze.
5. **Szyfrowanie wiadomości**  
   System do wysyłania i odbierania zaszyfrowanych wiadomości za pomocą kluczy asymetrycznych (RSA). Umożliwia użytkownikom generowanie własnych kluczy, szyfrowanie wiadomości kluczem publicznym i odszyfrowanie ich kluczem prywatnym. Skrypt zapewnia poufność, integralność i uwierzytelnianie nadawcy za pomocą podpisów cyfrowych. Może być rozbudowany o funkcje takie jak bezpieczna wymiana kluczy. Idealny do zrozumienia zastosowań kryptografii asymetrycznej w praktyce.
6. **System wykrywania prób ataków brute-force**  
   Skrypt monitoruje logi systemowe (np. /var/log/auth.log w Linuxie) w poszukiwaniu powtarzających się nieudanych prób logowania. Może wykorzystywać wyrażenia regularne do filtrowania danych i identyfikacji wzorców ataków. Po wykryciu podejrzanej aktywności automatycznie blokuje adres IP za pomocą firewalla (np. IPTables). Narzędzie to pomaga administratorom w szybkiej reakcji na zagrożenia, zmniejszając ryzyko przejęcia kont użytkowników.
7. **Zaawansowany skaner luk w aplikacjach webowych**  
   Analizuje aplikacje internetowe pod kątem podatności, takich jak SQL Injection, XSS czy CSRF. Wykorzystuje bibliotekę BeautifulSoup lub Selenium do interakcji z aplikacjami i narzędzie sqlmap do automatycznego testowania SQL Injection. Generuje szczegółowe raporty i rekomendacje dotyczące naprawy wykrytych problemów. Może być rozszerzony o moduły uczenia maszynowego do identyfikacji nowych typów podatności. Projekt dostarcza praktyczne doświadczenie w testach penetracyjnych.
8. **Wykrywanie złośliwego ruchu w sieci**  
   Analizuje dane sieciowe w czasie rzeczywistym, identyfikując nietypowe zachowania wskazujące na obecność złośliwego oprogramowania. Wykorzystuje algorytmy uczenia maszynowego (np. k-średnich, drzewa decyzyjne) zaimplementowane w Scikit-learn. Skrypt trenuje modele na danych o ruchu sieciowym i wykorzystuje je do klasyfikacji pakietów jako bezpieczne lub złośliwe. Projekt rozwija umiejętności analizy dużych zbiorów danych i zastosowania uczenia maszynowego w cyberbezpieczeństwie.
9. **Automatyzacja testów penetracyjnych**  
   Integruje narzędzia takie jak Nmap, Metasploit i Nikto w celu automatycznego przeprowadzania testów bezpieczeństwa. Skrypt tworzy scenariusze testów, takie jak skanowanie portów, testowanie podatności w aplikacjach webowych czy symulacje ataków exploitami. Generuje raporty z wynikami testów i proponowanymi poprawkami. Narzędzie może być używane przez zespoły bezpieczeństwa w celu skrócenia czasu audytu bezpieczeństwa.
10. **Skrypt monitorujący aktywność użytkowników**  
    Zapisuje aktywność użytkowników na serwerze, takie jak logowania, uruchamianie procesów czy zmiany w plikach. Wykorzystuje logi systemowe oraz narzędzia takie jak psutil do śledzenia procesów. Identyfikuje podejrzane działania, takie jak wykonywanie poleceń z uprawnieniami root bez autoryzacji. Skrypt generuje szczegółowe raporty i może wysyłać alerty do administratora. Jest przydatny w środowiskach wymagających ścisłego monitorowania zgodności z politykami bezpieczeństwa.
11. **Eksploracja danych z logów SIEM**  
    Analizuje logi z systemów SIEM przy użyciu bibliotek Python takich jak pandas oraz matplotlib. Skrypt agreguje dane, identyfikuje wzorce związane z bezpieczeństwem i generuje wizualizacje, np. wykresy anomalii w czasie. Umożliwia również korelację zdarzeń z różnych źródeł, takich jak logi serwerów, firewalla czy IDS. Dzięki temu narzędzie dostarcza analitykom cennych informacji na temat potencjalnych zagrożeń w systemie.
12. **Wykrywanie DNS Tunnelingu**  
    Analizuje zapytania DNS w celu identyfikacji nieautoryzowanego przesyłania danych za ich pomocą. Skrypt wykorzystuje analizy statystyczne (np. długość zapytań, nietypowe nazwy domen) oraz czarne listy. Może być wdrożony w czasie rzeczywistym lub jako narzędzie do analizy danych historycznych z logów DNS. Projekt jest użyteczny w środowiskach z wysokimi wymaganiami bezpieczeństwa, takich jak instytucje finansowe.
13. **Automatyczne łatanie luk bezpieczeństwa**  
    Wykrywa dostępne aktualizacje systemowe i aplikacyjne oraz automatycznie je instaluje. Skrypt obsługuje różne systemy operacyjne (np. apt na Linuxie) i sprawdza integralność plików po aktualizacji. Może również generować raporty o zainstalowanych poprawkach i statusie bezpieczeństwa systemu. To narzędzie pomaga w zarządzaniu aktualizacjami w dużych środowiskach IT.
14. **Prosty serwer honeypot**  
    Tworzy serwer-pułapkę, który symuluje różne usługi sieciowe (np. SSH, HTTP) w celu logowania prób ataków. Skrypt analizuje aktywność napastników i zapisuje dane, takie jak adresy IP, używane narzędzia czy techniki. Honeypot może być dostosowany do symulowania różnych scenariuszy ataków, co czyni go użytecznym narzędziem w badaniach nad zagrożeniami.
15. **Generowanie raportów zgodności z normami bezpieczeństwa**  
    Automatycznie analizuje konfigurację systemu i generuje raporty zgodne z normami takimi jak PCI DSS, ISO 27001 czy NIST. Skrypt sprawdza ustawienia haseł, otwarte porty, konfigurację firewalli i inne parametry. Wyniki są porównywane z wymaganiami normy, a narzędzie generuje listę działań naprawczych. Projekt rozwija umiejętności w zakresie audytu IT.
16. **Tworzenie kopii zapasowych z szyfrowaniem**  
    Program automatycznie tworzy zaszyfrowane kopie zapasowe wybranych plików i katalogów. Skrypt korzysta z algorytmu AES do szyfrowania i zapisuje dane na zewnętrznych nośnikach lub w chmurze. Dodatkowo implementuje harmonogram backupów oraz funkcję przywracania danych. To narzędzie pomaga w zarządzaniu danymi wrażliwymi w organizacjach.
17. **System analizy sandboxowej**  
    Uruchamia podejrzane pliki w kontrolowanym środowisku i monitoruje ich działania, takie jak zmiany w rejestrze systemowym, tworzenie nowych plików czy nawiązywanie połączeń sieciowych. Projekt wykorzystuje wirtualizację lub kontenery (np. Docker) oraz biblioteki Python do analizy zdarzeń. Jest przydatny w identyfikacji złośliwego oprogramowania.
18. **Skrypt do blokowania botów w aplikacjach webowych**  
    Analizuje ruch przychodzący na serwer i identyfikuje podejrzane zachowania, takie jak nadmierna liczba żądań z jednego adresu IP. Skrypt automatycznie blokuje podejrzane adresy za pomocą firewalla lub reguł serwera HTTP. Może być rozbudowany o integrację z systemami CAPTCHA. Narzędzie pomaga w ochronie aplikacji przed atakami typu DDoS.
19. **Analiza kryptograficzna**  
    Narzędzie do łamania prostych szyfrów, takich jak Vigenère czy transpozycyjne, na podstawie analizy częstotliwości znaków. Projekt wprowadza użytkownika w techniki kryptoanalizy i rozwija umiejętności związane z obróbką tekstu. Może być rozszerzony o obsługę bardziej złożonych szyfrów i ich wariantów. Narzędzie jest idealne do nauki kryptografii ofensywnej.
20. **Integracja z API Threat Intelligence**  
    Skrypt pobiera dane o potencjalnych zagrożeniach z zewnętrznych API, takich jak VirusTotal, AbuseIPDB czy Shodan. Analizuje pliki, adresy IP i domeny, dostarczając informacje o ich statusie bezpieczeństwa. Może działać jako samodzielne narzędzie lub element większego systemu bezpieczeństwa. Projekt rozwija umiejętności integracji z API i pracy z danymi JSON.
21. **Wykrywanie Beaconingu w sieci**  
    Analizuje dane sieciowe pod kątem sygnatur beaconów używanych przez złośliwe oprogramowanie do komunikacji z serwerami Command and Control (C2). Wykorzystuje techniki analizy statystycznej oraz algorytmy ML do identyfikacji regularnych wzorców komunikacji. Projekt wymaga integracji z narzędziami do monitorowania sieci i rozwija wiedzę z zakresu zaawansowanej analizy ruchu.
22. **System wczesnego ostrzegania (EWS)**  
    Skrypt analizuje logi lub ruch sieciowy w czasie rzeczywistym, generując alerty w przypadku wykrycia anomalii. Wykorzystuje podejścia heurystyczne oraz reguły zdefiniowane przez użytkownika. Projekt może być rozbudowany o moduł wizualizacji danych. Przydatny w dużych środowiskach sieciowych wymagających szybkiej reakcji na incydenty.
23. **Tworzenie skryptu do testowania haseł**  
    Automatyczne sprawdzanie złożoności haseł w systemie na podstawie znanych słowników i wzorców. Skrypt identyfikuje słabe hasła, generując raporty z rekomendacjami. Może być rozszerzony o funkcję generowania bezpiecznych haseł. Projekt jest idealny do poprawy bezpieczeństwa systemów IT.
24. **Prosty parser SNORT**  
    Analizuje i wizualizuje logi z systemu IDS Snort, umożliwiając szybkie zrozumienie zagrożeń w sieci. Skrypt wykorzystuje pandas do przetwarzania danych oraz matplotlib do ich wizualizacji. Narzędzie może być zintegrowane z SIEM w celu wzbogacenia analizy danych.
25. **Skrypt do analizy plików PDF pod kątem złośliwego kodu**  
    Sprawdza pliki PDF pod kątem nietypowych elementów, takich jak osadzone skrypty JavaScript lub ukryte dane. Wykorzystuje biblioteki takie jak PyPDF2 do parsowania plików i wyszukiwania podejrzanych sekcji. Narzędzie jest kluczowe w analizie incydentów związanych z phishingiem.
26. **Automatyczna detekcja plików o nietypowej aktywności**  
    Analizuje metadane i hashe plików w systemie w celu wykrycia potencjalnych zagrożeń. Skrypt identyfikuje pliki o nietypowych rozmiarach, częstotliwości zmian lub zgodnych z bazą sygnatur wirusów. Może działać jako narzędzie prewencyjne w systemach operacyjnych.
27. **Analizator pakietów sieciowych w czasie rzeczywistym**  
    Skrypt do analizy ruchu sieciowego w czasie rzeczywistym, identyfikujący podejrzane pakiety lub anomalie. Korzysta z scapy do przechwytywania danych i analizowania ich struktury. Idealny do dynamicznego monitorowania zagrożeń w sieci.
28. **Zaawansowany menedżer kluczy szyfrowania**  
    System do przechowywania i zarządzania kluczami szyfrowania, w tym ich automatycznej rotacji. Wspiera algorytmy takie jak AES i RSA oraz implementuje bezpieczne przechowywanie kluczy w bazach danych lub HSM. Projekt rozwija wiedzę na temat bezpieczeństwa kryptograficznego.
29. **Automatyzacja audytu Active Directory**  
    Analizuje konfigurację AD w poszukiwaniu słabych punktów, takich jak nadmiarowe uprawnienia czy błędne konfiguracje GPO. Skrypt generuje raporty z rekomendacjami dotyczącymi poprawy bezpieczeństwa. Jest przydatny w dużych środowiskach korporacyjnych.
30. **Symulacja ransomware (bez destrukcji)**  
    Tworzy symulację zachowania ransomware w kontrolowanym środowisku, umożliwiając testowanie skuteczności systemów ochrony. Skrypt symuluje proces szyfrowania plików i komunikacji z serwerem C2, bez faktycznego uszkadzania danych. Projekt pomaga w opracowywaniu procedur odpowiedzi na incydenty.