Bezpieczeństwo systemów komputerowych

Zbiór zadań, część pierwsza

Katarzyna Mazur

19 września 2024

Art. 267 KK

§ *Kto bez uprawnienia uzyskuje dostęp do całości lub części systemu informatycznego podlega karze pozbawienia wolności do lat 2.*

Art. 269a KK

§ *Kto, nie będąc do tego uprawnionym w istotnym stopniu zakłóca pracę systemu systemu teleinformatycznego lub sieci teleinformatycznej, podlega karze pozbawienia wolności od 3 miesięcy do lat 5.*

Art.269c KK

§ *Nie podlega karze za przestępstwo określone w art. 267 § 2 lub art. 269a, kto działa wyłącznie w celu zabezpieczenia systemu teleinformatycznego albo opracowania metody takiego zabezpieczenia i niezwłocznie powiadomił dysponenta tego systemu lub sieci o ujawnionych zagrożeniach, a jego działanie nie naruszyło interesu publicznego lub prywatnego i nie wyrządziło szkody.*

1 Informacje

GitHub

Na potrzeby zajęć swoje własne repozytorium na Githubie. Rezpozytorium będziesz wykorzystywać, aby wrzucać do niego rozwiązania zadań z przedmiotu. Możesz nazwać repozytorium dowolnie, ale najlepiej wykorzystaj nazwę: `imie-nazwisko-bsk-umcs`.

Wirtualne środowisko pracy

Wirtualne środowisko pracy w Python, znane również jako `virtualenv` lub `venv`, jest sposobem za zapewnienie działania naszego programu niezależnie od maszyny na której jest uruchamiany ORAZ sposobem na uruchomienie innych programów na NASZEJ maszynie, tak aby instalowane z nią zależności nie zakłóciły pracy innych programów. Wirtualne środowisko pracy, jest swojego rodzaju odizolowanym katalogiem, zawierającym instalację języka programowania Python oraz zainstalowane biblioteki na potrzeby programu, który będziemy w nim uruchamiać.

Instalacja

```
sudo apt update
sudo apt install python3-pip
python3 -m pip install --upgrade pip

sudo apt install python3-venv
python3 -m pip install --user virtualenv
```

Tworzenie środowiska

```
python3 -m venv venv
source ./venv/bin/activate
```

Instalowanie pakietów

```
pip list
pip install black
black test.py
```

Dezaktywacja (koniec pracy) wirtualnego środowiska

```
deactivate
```

Dobre praktyki

W zadaniach najczęściej będziemy wykorzystywać język programowania Python. Aby zachować dobre praktyki programistyczne, będziemy używać black'a oraz pylamy. Jeśli to możliwe, możesz wykorzystać inny język programowania (jest to zależne od zadania). Pamiętaj jednak również o zachowaniu dobrych praktyk.

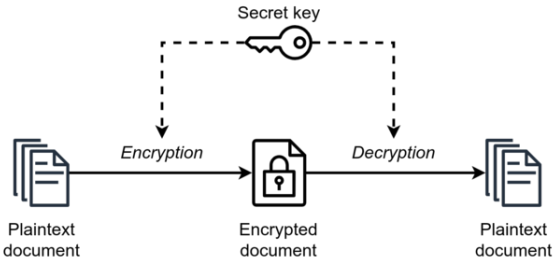
Materiały

- `https://securecodingdojo.owasp.org/public/index.html`
- `https://overthewire.org/wargames/`
- `https://pentesterlab.com/exercises`

3 Szyfrowanie / deszyfrowanie symetryczne

3.1 Teoria

Szyfrowanie to proces przekształcania danych w taki sposób, aby stały się one nieczytelne dla nieautoryzowanych osób. Głównym celem szyfrowania jest zapewnienie poufności danych, co oznacza, że tylko osoby posiadające odpowiedni klucz mogą odczytać oryginalną treść. Proces szyfrowania wykorzystuje algorytmy kryptograficzne, które operują na danych wejściowych i przekształcają je za pomocą klucza kryptograficznego.



Szyfrowanie symetryczne to metoda ochrony danych, w której ten sam klucz kryptograficzny jest używany zarówno do szyfrowania, jak i deszyfrowania informacji. Charakteryzuje się:

- Jednym kluczem: Ten sam klucz jest używany do szyfrowania i deszyfrowania danych. Zarówno szyfrowanie, jak i deszyfrowanie wykorzystują ten sam tajny klucz. Klucz musi być bezpiecznie przechowywany i wymieniany między stronami komunikacji.
- Wysoką wydajnością: Szyfrowanie symetryczne jest zazwyczaj szybsze i bardziej efektywne obliczeniowo niż szyfrowanie asymetryczne. Algorytmy symetryczne są zazwyczaj szybsze niż asymetryczne, co sprawia, że są idealne do szyfrowania dużych ilości danych w czasie rzeczywistym.
- Wymagana bezpieczna dystrybucja klucza: Bezpieczeństwo zależy od utrzymania klucza w tajemnicy; obie strony muszą bezpiecznie wymienić klucz przed rozpoczęciem komunikacji.
- Algorytmy symetryczne są zaprojektowane tak, aby były odporne na różne typy ataków kryptograficznych, takie jak kryptoanaliza liniowa i różnicowa.

Typy algorytmów symetrycznych:

- Algorytmy blokowe (np. AES, DES) operują na danych w blokach o stałej długości, zwykle 64 lub 128 bitów. Algorytmy blokowe mogą działać w różnych trybach operacyjnych, takich jak ECB (Electronic Codebook), CBC (Cipher Block Chaining), CFB (Cipher Feedback), OFB (Output Feedback) i CTR (Counter), które wpływają na bezpieczeństwo i sposób przetwarzania danych.
- Algorytmy strumieniowe (np. RC4) szyfrują dane jako ciąg bajtów lub bitów, jeden po drugim.

3.2 Narzędzia

John the Ripper (JtR)

John the Ripper (JtR) to zaawansowane narzędzie do łamania haseł, które jest szeroko stosowane przez specjalistów ds. bezpieczeństwa i pentesterów do testowania odporności systemów na ataki brute force i inne techniki łamania haseł. Oto kilka kluczowych aspektów tego narzędzia:

- Łamanie haseł: John the Ripper może łamać hasła zapisane w różnego rodzaju formatach haseł (hashach), takich jak DES, MD5, SHA-1, bcrypt i wiele innych.
- Wszechstronność: Obsługuje wiele systemów operacyjnych, w tym Unix, Linux, Windows, DOS, BeOS i OpenVMS. Może łamać hasła z różnych źródeł, takich jak pliki shadow z systemów Unix, pliki SAM z systemów Windows, czy nawet hasła Wi-Fi.
- Techniki łamania:
  - Brute force: Przeszukiwanie wszystkich możliwych kombinacji znaków do momentu znalezienia prawidłowego hasła.
  - Dictionary attack: Wykorzystanie listy słów (słownik) do prób odgadnięcia hasła.
  - Hybrid attack: Połączenie ataku słownikowego z modyfikacjami słów, takimi jak dodawanie cyfr, zmiana wielkości liter itp.

fcrackzip

fcrackzip to narzędzie do łamania haseł zabezpieczających archiwa ZIP. Jego głównym celem jest odzyskiwanie haseł do plików ZIP poprzez różne metody ataków, co jest przydatne w przypadku zapomnienia hasła do własnego archiwum lub testowania odporności zabezpieczeń ZIP.

OpenSSL

To wieloplatformowa, otwarta implementacja protokołów SSL (wersji 2 i 3) i TLS (wersji 1) oraz algorytmów kryptograficznych ogólnego przeznaczenia. Dostępna jest dla systemów uniksopodobnych (m.in. Linux, BSD, Solaris), OpenVMS i Microsoft Windows. OpenSSL zawiera biblioteki implementujące wspomniane standardy oraz mechanizmy kryptograficzne, a także zestaw narzędzi konsolowych (przede wszystkim do tworzenia kluczy oraz certyfikatów, zarządzania urzędem certyfikacji, szyfrowania, dekryptażu i obliczania podpisów cyfrowych). OpenSSL pozwala na używanie wszystkich zastosowań kryptografii. Poniżej kilka kluczowych cech OpenSSL:

- **Wszechstronność**: OpenSSL oferuje bogaty zestaw narzędzi i bibliotek do obsługi różnych protokołów kryptograficznych, w tym SSL/TLS, kryptografię klucza publicznego, szyfrowanie danych i wiele innych.
- **Bezpieczeństwo sieciowe**: OpenSSL dostarcza narzędzia do zarządzania certyfikatami, generowania kluczy, podpisywania cyfrowego i szyfrowania danych, co umożliwia tworzenie bezpiecznych aplikacji internetowych i usług.
- **Wsparcie dla różnych platform**: OpenSSL jest dostępny na wielu platformach, w tym na systemach Unix, Linux, macOS, Windows oraz innych, co czyni go popularnym narzędziem wśród programistów i administratorów systemów.
- **Otwarty kod źródłowy**: OpenSSL jest projektem open-source, co oznacza, że jego kod jest dostępny publicznie i może być modyfikowany oraz rozwijany przez społeczność, co sprzyja ciągłemu ulepszaniu i dostosowywaniu narzędzia do różnych potrzeb i zastosowań.
- **Wsparcie dla wielu protokołów**: OpenSSL obsługuje wiele standardowych protokołów kryptograficznych, takich jak SSL/TLS, SSH, S/MIME, PKCS, co czyni go wszechstronnym narzędziem do implementacji bezpiecznych komunikacji w różnych aplikacjach i systemach.

Hashcat

To potężne narzędzie do łamania haseł, które jest często używane przez specjalistów ds. bezpieczeństwa informatycznego do testowania i oceny siły haseł oraz algorytmów haszujących. Poniżej przedstawiam kilka kluczowych cech Hashcata:

- **Wsparcie dla wielu algorytmów haszujących**: Hashcat obsługuje szeroki zakres algorytmów haszujących, w tym popularne funkcje takie jak MD5, SHA-1, SHA-256, bcrypt, scrypt i wiele innych.
- **Zaawansowane techniki ataków**: Narzędzie Hashcat oferuje różnorodne techniki ataków, takie jak ataki brute-force, ataki słownikowe, ataki kombinatoryczne, ataki hybrydowe i wiele innych, co pozwala na skuteczne łamanie haseł przy różnych scenariuszach i konfiguracjach.
- **Wsparcie dla platform sprzętowych**: Hashcat korzysta z potęgi obliczeniowej kart graficznych (GPU), co znacznie przyspiesza proces łamania haseł. Ponadto, jest również kompatybilny z CPU, co daje większą elastyczność w wyborze platformy sprzętowej.
- **Możliwość pracy na wielu platformach**: Hashcat jest dostępny na różnych systemach operacyjnych, w tym na Windowsie, Linuxie i macOS, co czyni go wszechstronnym narzędziem dostępnym dla szerokiego grona użytkowników.
- **Otwarte oprogramowanie**: Hashcat jest projektem open source, co oznacza, że jego kod jest dostępny publicznie i może być modyfikowany i rozwijany przez społeczność, co sprzyja ciągłemu ulepszaniu i dostosowywaniu narzędzia do różnych potrzeb i zastosowań.

cURL

`curl` to potężne narzędzie wiersza poleceń, które umożliwia wykonywanie zapytań HTTP (i nie tylko) do serwerów internetowych. Jego nazwa pochodzi od Ćlient URL", co odzwierciedla jego główną funkcję - interakcję z URL-ami. `curl` jest szeroko używane w skryptach i automatyzacji, a także jako narzędzie do testowania i debugowania API.

Główne cechy `curl`:

- **Prosta interakcja z URL-ami**:
  - `curl` pozwala na pobieranie danych z serwera za pomocą różnych metod HTTP (GET, POST, PUT, DELETE, itp.).
  - Może także wysyłać dane do serwera, co jest użyteczne do interakcji z API i przesyłania formularzy.
- **Obsługuje wiele protokołów**:
  - `curl` wspiera wiele protokołów, w tym HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP, i wiele innych.
- **Obsługuje nagłówki i ciasteczka**:
  - Możesz dodawać, edytować i usuwać nagłówki HTTP w zapytaniach, co jest użyteczne przy pracy z API, które wymagają specjalnych nagłówków.

  - `curl` może również zarządzać ciasteczkami.
- **Autoryzacja i uwierzytelnianie**:
  - `curl` obsługuje różne mechanizmy uwierzytelniania, takie jak Basic Auth, Digest Auth, Bearer Tokens, OAuth, itp.
- **Zarządzanie danymi wejściowymi i wyjściowymi**:
  - Możesz przekazywać dane do zapytań za pomocą opcji `-d` lub `-data`.
  - Odpowiedzi można zapisywać do plików lub przekierowywać do innych poleceń za pomocą standardowych mechanizmów Unixowych.
- **Debugowanie i testowanie**:
  - `curl` ma wiele opcji do debugowania, takich jak `-v` (verbose) do wyświetlania szczegółowych informacji o zapytaniu i odpowiedzi.
  - Możesz używać opcji `-i` do wyświetlania nagłówków odpowiedzi wraz z treścią.

Linki

- `https://www.openssl.org/docs/man1.0.2/man1/openssl-enc.html`
- `https://wiki.openssl.org/index.php/Enc`
- `https://www.kali.org/tools/fcrackzip/`
- `https://github.com/hyc/fcrackzip`
- `https://manpages.ubuntu.com/manpages/xenial/man1/fcrackzip.1.html`
- `https://www.openwall.com/john/`
- `https://github.com/openwall/john`
- `https://www.kali.org/tools/john/`
- `https://hashcat.net/hashcat/`
- `https://hashcat.net/wiki/doku.php?id=example_hashes`
- `https://github.com/hashcat/hashcat`
- `https://pypi.org/project/pycryptodome/`
- `https://www.pycryptodome.org/`
- `https://pypi.org/project/cryptography/#description`
- `https://cryptography.io/en/stable/`
- `https://github.com/Legrandin/pycryptodome/`
- `https://www.pycryptodome.org/`
- `https://cisspmadeeasy.com/2024/02/28/encryption-algorithm-types-and-modes/`

Zadania

2.1 Wykonaj szyfrowanie ciągu znaków z pliku `ex2.1.txt` za pomocą algorytmu AES-256-ECB z użyciem podanego klucza. Klucz znajduje się w pliku `ex2.1.key`. Odpowiedź (zaszyfrowany tekst) zakoduj kodowaniem Base64. Klucz użyty podczas szyfrowania powinien być podawany z linii komend. Prawidłowa odpowiedź do zadania znajduje się w pliku `ex2.1.enc`.

2.2 Wykonaj deszyfrowanie pliku `ex2.2.enc` za pomocą algorytmu AES-256-ECB z użyciem podanego klucza z pliku `ex2.2.key`. Klucz powinien być podawany w linii komend. Wynikiem powinien być zrozumiały tekst.

2.3 Wykonaj deszyfrowanie pliku `ex2.3.enc` za pomocą algorytmu CAMELLIA-128-ECB z użyciem podanego hasła z pliku `ex2.3.key`. Hasło powinno być podawane z pliku.

2.4 Wykonaj deszyfrowanie pliku `ex2.4.enc` za pomocą algorytmu AES-256-CBC z użyciem podanego hasła z pliku `ex2.4.pass`, wiedząc, że funkcja generowania klucza to PBKDF2.

2.5 Wykonaj deszyfrowanie pliku `ex2.5.enc` za pomocą algorytmu 3DES z użyciem podanego klucza z pliku `ex2.5.key`, wiedząc, że funkcja generowania klucza to PBKDF2.

2.6 Wykonaj szyfrowanie pliku `ex2.6.txt` za pomocą algorytmu ARIA-ECB z użyciem klucza, który wygenerujesz za pomocą `OpenSSL rand` (wybierz odpowiednią długość klucza dla algorytmu). Następnie wykonaj deszyfrowanie pliku, zapisując wynik deszyfrowania do pliku `ex2.6.dec`. Za pomocą polecenia `diff` lub `md5sum` sprawdź, czy pliki `ex.2.6.txt` oraz `ex2.6.dec` są identyczne.

2.7 Wykonaj deszyfrowanie pliku `ex2.7.enc` za pomocą algorytmu AES-256-ECB z użyciem podanego klucza z pliku `ex2.7.key`, algorytmu generowania klucza PBKDF1 oraz wskazanej ilości iteracji algorytmu równej 356.

2.8 Wykonaj deszyfrowanie pliku `ex2.8.txt` za pomocą algorytmu AES-256-CBC z użyciem podanego hasła z pliku `ex2.8.pass`, algorytmu generowania klucza PBKDF2 oraz wskazanej ilości iteracji algorytmu równej 41331.

2.9 Ze strony kursu pobierz plik `ex2.9.zip`. Plik ten jest zabezpieczony hasłem. Jest to jedno z najczęściej używanych przez użytkowników haseł. Za pomocą programu JohnTheRipper spróbuj złamać hasło, którym zaszyfrowany jest plik. Możesz skorzystać z listy najpopularniejszych haseł dostępnej na githubie: `https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10k-most-common.txt`.

2.10 Ze strony kursu pobierz plik `ex2.10.zip`. Plik ten jest zabezpieczony hasłem. Wiedząc, że plik ten jest zabezpieczony hasłem o długości pomiędzy 5-6 znaków, i zawiera jedynie cyfry, za pomocą programu JohnTheRipper spróbuj złamać hasło, którym zaszyfrowany jest plik. Wygeneruj listę możliwych haseł za pomocą programu `crunch`.

2.11 Wykonaj zadanie **2.9** za pomocą narzędzia `fcrackzip`.

2.12 Zidentyfikuj, jaki algorytm szyfrujący został wykorzystany do zaszyfrowania tekstu: `Z8CerTOLe1JlDKWfvDeifw==` przy pomocy klucza: `a35febba42490abe`.

2.13 W pliku `ex2.13.enc` znajduje się zaszyforwany za pomocą klucza z pliku `ex2.13.key` obrazek w formacie `*.png`. Odszyfruj obrazek. Rozwiązaniem zadania powinien być plik `*.png`. Do szyfrowania obrazka użyto algorytmu SEED-ECB.

2.14 Ze strony kursu pobierz plik `ex2.14.zip`. Plik ten jest zabezpieczony hasłem. Spróbuj złamać hasło za pomocą narzędzia `hashcat` wykorzystując atak słownikowy.

2.15 Korzystając z przygotowanego obrazu Dockerowego (`mazurkatarzyna/bsk-book-p1-ch2-ex215:latest`), uruchom serwer HTTP. Po uruchomieniu obrazu, pod adresem IPv4 `127.0.0.1` na porcie TCP o numerze `20015` działa serwer obsługujący protokół HTTP w wersji 1.1. Serwer udostępnia 2 endpointy:

- `/random` - generuje losowe słowo,
- `check_aes256ecb/[ciphertext]` - sprawdza, czy zaszyfrowany tekst podany jako argument jest wylosowanym słowem, prawidłowo zaszyfrowanym algorytmem `AES-256-ECB`

Aby rozwiązać zadanie:

- Wylosuj słowo za pomocą endpointa `/random`
- Następnie, pomocą narzędzia `OpenSSL` zaszyfruj za pomocą algorytmu `AES-256-ECB` wylosowane słowo. Wykorzystując endpoint `/check_aes256ecb/[ciphertext]` sprawdź, czy prawidłowo zaszyfrowałaś/eś wylosowane słowo.

- Do rozwiązania zadania możesz użyć przeglądarki lub narzędzia `cURL`. Klucz szyfrujący znajduje się w pliku `ex2.15.key`.

2.16 Wykorzystując bibliotekę PyCryptodome, napisz 2 funkcje: szyfrującą i deszyfrującą dowolny tekst za pomocą algorytmu AES CBC z kluczem 128 bitów. Program możesz napisać i przetestować korzystając z kontenera Dockerowego, który ma już zainstalowaną najnowszą wersję biblioteki PyCryptodome:
`docker run -it mazurkatarzyna/bsk-book-p1-ch2-ex216:latest`.

2.17 Wykorzystując bibliotekę PyCryptodome, napisz 2 funkcje: szyfrującą i deszyfrującą dowolny tekst za pomocą algorytmu AES GCM z kluczem 256 bitów. Program możesz napisać i przetestować korzystając z kontenera Dockerowego, który ma już zainstalowaną najnowszą wersję biblioteki PyCryptodome:
`docker run -it mazurkatarzyna/bsk-book-p1-ch2-ex217:latest`.