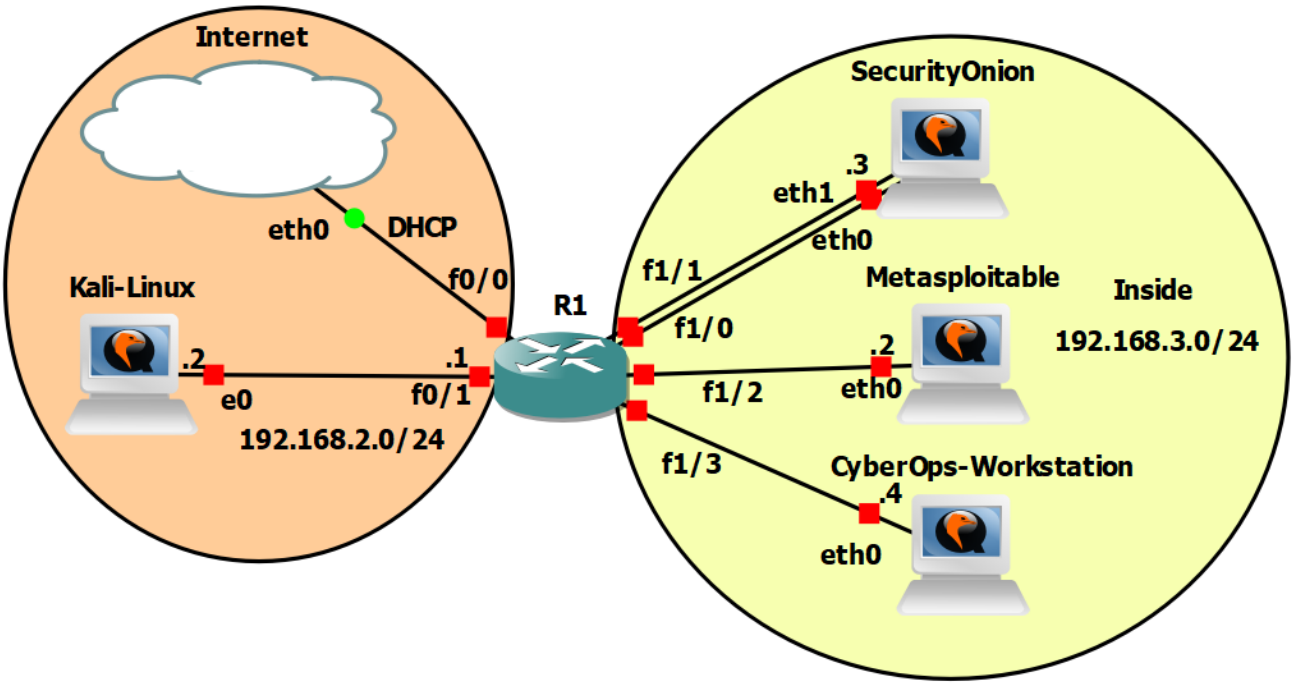
RBI / Cvičenie 08/ **Šifrovanie a dešifrovanie dát**

Topológia



Inštrukcie a scenár

V tomto cvičení sa naučíte používať rôzne typy hašovacích algoritmov na vytváranie hašov zo súborov. Taktiež sa naučíte používať haš na overenie integrity súborov. Vyskúšate si aj šifrovanie textového súboru a jeho následné dešifrovanie pomocou nástroja OpenSSL. Vašou úlohou bude ďalej aj dešifrovať heslá, ktoré ste získali z aplikácie DVWA na cvičení **06_Webové útoky SQL Injection a XSS.** Na dešifrovanie budete používať nástroj **john.** V ďalšej časti vytvoríte súbory, ktoré zaheslujeme do ZIP súboru a následne použijete nástroj **fcrackzip** na prelomenie hesla. Ukážeme si aj metódy ako šifrovať a vytvárať heslo, aby bolo takmer neprelomiteľné. Vyskúšame rôzne šifrovacie metódy.

Toto laboratórne cvičenie vzniklo aj na základe týchto oficiálnych Netacad labov a ich doplnením:

- 21.1.6 - Hashing Things Out
- 21.2.10 - Encrypting and Decrypting Data Using OpenSSL
- 21.2.11 - Encrypting and Decrypting Data Using a Hacker Tool

Požiadavky

- Topológia v GNS3 alebo virtuálne zariadenia vo VirtualBox-e

Používatelia

Názov	Meno	Heslo
KALI LINUX	kali	kali
METASPLOITABLE	msfadmin	msfadmin
SECURITY ONION	analyst	cyberops
CYBEROPS WORKSTATION	analyst	cyberops

Časť 1: Hašovanie s OpenSSL

Transformačné (hašovacie) funkcie sú matematické algoritmy navrhnuté tak, aby brali údaje ako vstup a generovali jedinečný reťazec znakov s pevnou veľkosťou, známy aj ako hash. Hashovacie funkcie sú navrhnuté tak, aby boli rýchle a aby bolo veľmi ťažké obnoviť údaje z daného hašu. Ďalšou dôležitou vlastnosťou hašovacej funkcie je, že aj najmenšia zmena vykonaná vo vstupných údajoch prinesie úplne iný haš (hash). Zatiaľ čo OpenSSL možno použiť na generovanie a porovnávanie hašov, k dispozícii sú aj iné nástroje. Niektoré z týchto nástrojov sú zahrnuté aj v tomto cvičení.

OpenSSL možno použiť ako samostatný nástroj na hašovanie. Ak chcete vytvoriť „hash“ textového súboru, postupujte podľa nasledujúcich krokov:

1. Otvorte terminál na zariadení CyberOps Workstation
2. Vytvorte si súbor **password.txt**, do ktorého zadáte 5-10 miestne heslo – zadajte tam vaše priezvisko
3. Z okna terminálu zadajte príkaz uvedený nižšie na zahašovanie vášho súboru s heslom Príkaz použije `SHA-2-256` ako hašovací algoritmus na generovanie hašu textového súboru. Haš sa zobrazí na obrazovke, keď ho OpenSSL vypočíta
   a. `$ openssl sha256 password.txt`
   b. Všimnite si formát výstupu. Najskôr je vypísaný použitý hašovací algoritmus, ďalej v zátvorkách je uvedený vstupný súbor a za znakom „=“ je vypísaný samotný hash
4. Upravte si textový súbor **password.txt** tak, že zmeníte jeden znak z vášho hesla
5. Teraz znova použite ten istý hashovací algoritmus
   a. Je nový hash iný ako hash generovaný v bode 3.a)? Ak áno ako veľmi iný?
6. Možno použiť aj hashovací algoritmus s väčšou bitovou dĺžkou, ako je SHA-2-512. Ak chcete vygenerovať hash SHA-2-512 súboru **password.txt**, použite príkaz nižšie:
   a. `$ openssl sha512 password.txt`
7. Teraz použijeme iný nástroj na hashovanie. Pomocou **sha256sum** a **sha512sum** vygenerujte hash SHA-2-256 a SHA-2-512 súboru **password.txt:**
   a. `$ sha256sum password.txt`
   b. `$ sha512sum password.txt`
   c. Zhodujú sa hashe vygenerované pomocou sha256sum a sha512sum s hashmi vygenerovanými v položkách (3.a) a (6.a)? Vysvetlite prečo je tomu tak.
8. Poznámka: SHA-2 je odporúčaný štandard pre hashovanie. Zatiaľ čo SHA-2 ešte nebol efektívne kompromitovaný, počítače sú čoraz výkonnejšie. Očakáva sa, že tento prirodzený vývoj čoskoro umožní útočníkom prelomiť SHA-2
9. SHA-3 je najnovší hashovací algoritmus a je náhradou za SHA-2
10. Poznámka: CyberOPS Workstation VM obsahuje iba podporu pre SHA-2-224, SHA-2-256 a SHA-2-512 (sha224sum, sha256sum a sha512sum)

Časť 2: Verifikácia Hashov

Ako už bolo spomenuté vyššie, hash sa bežne používa na overenie integrity súboru. Ak chcete použiť hodnoty hashu SHA-2-256 na overenie integrity súboru sample.img, súboru stiahnutého z internetu, postupujte podľa nasledujúcich krokov

1. Spolu so sample.img bol stiahnutý aj sample.img_SHA256.sig. sample.img_SHA256.sig je súbor obsahujúci hash SHA-2-256, ktorý vypočítala webová stránka. Najprv použite príkaz cat na zobrazenie obsahu súboru sample.img_SHA256.sig:
    a. `$ cat sample.img_SHA256.sig`
2. Na výpočet hashu SHA-2-256 súboru sample.img použite SHA256sum
    a. `$ sha256sum sample.img`
3. Bol súbor sample.img stiahnutý bez chýb? Vysvetlite.
4. Poznámka: Aj keď je porovnávanie hashov pomerne robustná metóda na zistenie chýb prenosu, existujú lepšie spôsoby, ako zistiť, že sa so súborom nemanipulovalo. Nástroje, ako napríklad gpg, poskytujú oveľa lepšiu metódu na zistenie toho, že stiahnutý súbor nebol upravený tretími stranami a v skutočnosti ide o súbor, ktorý vydavateľ zverejnil.

Časť 3: Šifrovanie a dešifrovanie s OpenSSL

OpenSSL je open source projekt, ktorý poskytuje robustnú, komerčnú a plne vybavenú sadu nástrojov pre protokoly Transport Layer Security (TLS) a Secure Sockets Layer (SSL). Je to tiež univerzálna kryptografická knižnica. V tejto časti budete používať OpenSSL na šifrovanie a dešifrovanie textových správ.

Poznámka: Zatiaľ čo OpenSSL je dnes de facto kryptografickou knižnicou, použitie prezentované v tomto laboratórnom cvičení sa NEODPORÚČA na robustnú ochranu. Nižšie sú uvedené dva bezpečnostné problémy:

1) Metóda opísaná v tejto časti používa slabú kľúčovú odvodenú funkciu. JEDINÉ zabezpečenie predstavuje veľmi silné heslo.

2) Metóda opísaná v tejto časti nezaručuje integritu textového súboru.

Príklady z tejto časti by sa mali používať iba na inštruktážne účely. Uvedené metódy by sa NEMALI používať na zabezpečenie skutočne citlivých údajov.

1. Zašifrujeme textový súbor
    a. OpenSSL možno použiť ako samostatný nástroj na šifrovanie. Aj keď je možné použiť mnoho šifrovacích algoritmov, v tomto cvičení sa zameriava na AES
    b. Prihlásime sa do CyberOps Workstation
    c. Otvoríme terminálové okno
        i. Otvoríme adresár `./lab.support.files/`
            - Vypíšeme obsah súboru v termináli `letter_to_grandma.txt`
            - Premenuje si názov tohto súboru na `letter_to_priezvisko.txt`, kde namiesto `priezvisko` zadáte vaše priezvisko, a vždy keď v zadaní uvedieme súbor `letter_to_grandma.txt`, vy použijete tento svoj premenovaný súbor.
    d. Zašifrujeme tento súbor priamo v termináli
        i. Použijeme OpenSSL a algoritmus AES-256

ii. Výstup uložíme do súboru `message.enc` – v zadaní sa budeme odkazovať na súbor `message.txt`, vy použite názov súboru `message_priezvisko.enc`, kde namiesto priezvisko zadajte svoje priezvisko
- $ `openssl aes-256-cbc -in letter_to_grandma.txt -out message.enc`

iii. OpenSSL si vyžiada heslo a jeho potvrdenie
- Zapamätajte si ho

e. Použite príkaz `cat` na vypísanie súboru `message.enc`
i. Zobrazil sa súbor správne? (hint: binary file)
ii. Ako vyzerá?

f. Použite OpenSSL znova pre zašifrovanie súboru, ale zvoľte `-a` možnosť. Voľba `-a` hovorí OpenSSL, aby pred uložením výsledkov do súboru zakódoval zašifrovanú správu pomocou inej metódy kódovania, a to Base64.
i. $ `openssl aes-256-cbc -a -in letter_to_grandma.txt -out message.enc`

g. Vypíšte súbor `message.enc` znova
i. Zobrazil sa súbor správne? (hint: binary to text)

2. Rozšifrovanie textového súboru
a. Použite príkaz na rozšifrovanie súboru – namiesto `decrypted_letter` použite `decrypted_letter_priezvisko`, a ďalej pracujte s týmto zmeneným názvom súboru
i. $ `openssl aes-256-cbc -a -d -in message.enc -out decrypted_letter.txt`
b. Vložte heslo, ktoré ste si zvolili
c. Vypíšte súbor `decrypted_letter.txt`
i. Bol tento súbor rozšifrovaný správne?
d. Príkaz používaný na dešifrovanie obsahuje aj voľbu `-a`. Viete vysvetliť prečo?

Časť 4: Dešifrovanie MD5 hesiel z aplikácie DVWA a útok hrubou silou na ZIP

1. **Pripravenie hesiel**
a. Prihláste sa do Kali Linux
b. Nájdite súbor, kde ste si uložili zašifrované hesla z cvičenia *RBI_LAB06_Webové útoky SQL Injection a XSS a pohľad SOC analytika*
c. Otvorte súbor a upravte ho tak aby bol formát v podobe `meno:zašifrované_heslo`
d. Výsledný súbor by mal vyzerať nasledovne:
```
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99
```
e. Uložte si tento súbor

2. **Dešifrovanie hesiel**
a. Na dešifrovanie týchto hesiel použijeme nástroj **john the ripper**
i. **John the Ripper** je bezplatný open-source nástroj slúžiaci na prelomenie hesiel
b. Otvorte si terminál a zadajte príkaz: `john -h`
i. Na terminál sa vám vypíšu možné prepínače, ktoré podporuje tento program
ii. Ako ďalší príkaz použite: `john --list=formats`
- Zobrazia sa vám podporované formáty

c. Teraz bude vašou úlohou rozšifrovať heslá z daného súboru, ktorý ste si uložili. Použite formát `raw-md5`

i. Príkaz: `john --format=raw-md5 súbor`
- Čo vidíme vo výsledku?

iv. Pomocou príkazu `john --show --format=raw-md5 hesla` si môžete zobraziť heslá, ktoré sa rozšifrovali

3. **Vytvorenie hesla pre ZIP súbor a útok hrubou silou na heslo**
   a. Prihláste sa do CyberOps Workstation
   b. Vytvorenie 5 textových súborov
      i. Otvorte terminál
      iii. Vytvorte nový priečinok: ZipFiles (mkdir ZipFiles)
      iv. Choďte do priečinku: ZipFiles (cd ZipFiles)
      v. Pre vytvorenie súboru použite príkaz: `echo Text > priezvisko#.txt` (# číslo súboru, a namiesto `priezvisko` zadajte vaše priezvisko, a v nasledujúcej časti zadania používajte tento svoj názov súborov)
         - Napr.: echo Tajná sprava1 pre kapitána > priezvisko1.txt
           echo Tajná sprava2 pre kapitána > priezvisko2.txt atď..
      vi. Použite príkaz `ls` na overenie či sa tieto správy podarilo vytvoriť
   c. Zazipujte a zaheslujete tieto textové súbory
      i. Na tento účel použijeme nástroj **zip**
         - Použite príkaz: `zip -h`
         - Zobrazí sa vám nápoveda pre nástroj **zip**

      ii. Pre zaheslovanie použite príkaz:
      `zip -e zip_priezvisko.zip priezvisko*`
         - Zadajte maximálne 5 miestne heslo (kvôli tomu aby vám rozšifrovanie netrvalo dlho)
         - `-e`: znamená, že sa výsledný súbor zašifruje
         - `zip_priezvisko.zip` : výsledný zaheslovaný súbor
         - `priezvisko*` : zazipuje všetky súbory, ktoré obsahujú na začiatku mena súboru „priezvisko“

         Aký je výsledok?
         - Vyskúšajte **unzip** pre tento súbor a použite zle heslo: `unzip zip_spravy.zip`

         Aký je výsledok?
   c. Na odšifrovanie tohto zip súboru použijeme nástroj **fcrackzip**, ktorý skúša heslá hrubou silou
      i. Použite príkaz na zobrazenie prepínačov: `fcrackzip -h`
         - Následne použite príkaz pre zistenie hesla:
           `fcrackzip -v -u -l 1-5 zip_priezvisko.zip`
           Aký je výsledok?
   d. Čím zložitejšie heslo použijete, tým dlhšie trvá útok hrubou silou na heslo.

Časť 5: Porovnajte šifrovacie algoritmy

1. Zašifrujte si heslo
   a. Choďte na stránku https://passwords-generator.org/
      i. Zvoľte si ľubovoľné heslo (najlepšie krátke aby rozšifrovanie netrvalo dlho – použite vaše priezvisko, alebo jeho časť)
      ii. Heslo zašifrujte všetkými hašovacími funkciami na stránke (md5, sha1, sha256, sha512)
2. Porovnajte zložitosť šifier

a. Použite nástroj **john the ripper** na rozšifrovanie vašich hašov a porovnajte, ktorý haš, prislúchajúci danej hašovacej funkcií, bol rozšifrovaný najskôr a naopak, ktorý bol rozšifrovaný ako posledný
b. Môžete použiť heslo pre zip súbor a následne vykonať to isté s nástrojom fcrackzip
c. Skúste tieto šifry porovnať s nástrojom OpenSSL