Hashing and Salt

06 February 2024 12:22

Possible Questions

*1) OSI Layer — Definite question

2) Ports and Services — Well known ports

3) Firewall — IDS — IPS

4) TLS — Might

5) Hashing, sy. Encryption & Syme

6) CIA — Give some example.

7) Difference b/w Router and a Switch?

— MAC — Switch (L2) — Connect hosts internally

— IP — Router (L3) — Connect two networks

Encryption — De

Hashing — Hash function

mapping data into a fixed set of values

(Mapping is irreversable)

M — Message

h(M) = {hash value} chara

(Data)

{H → M original} guessing

Hash value

derive

Message

One way hash function

$f(x) = x^2$

$f(5) = 25$ 25, x^2 Two way function

$x = 5$

(h)

* You can't derive the original value from the hash functions value.

Hash function – one way

* What's the use of a hash value?
  – Integrity – Verify modified (or) not

M – h(M) –
M' – h(M') – } change the hash.

Even a slight change in the message will change the hash value.

– Software Integration
  – (Hash)
  Re-compute – Hash – Authentic
  Don't match – Virus (or) Modified file.

– For storing passwords (or) Sensitive data – DB (or) a password file.
  got leaked
  Plain passwords – Can be read
  Hashed passwords – Even after a leak
  ↓ Can't guess the original pass.

DNS Records

1) Type A –
2) AAAA –
3) MX
4) CNAME –
5) SOA –
6) NS Records
7) TXT Records
8) ANAME

Salting in a hash functions

Operating – Linux OS – /etc/password file

Uses a Rand Random Hash (RH) – password

Operating — —

Veena : ~~bit~~ Random Hash (RH)

Sritha : {RH} — Test } ≠ common password

Real password ↓ Hash (Dictionary attack)

cracked the password

Avoid - Salt

— Veena ← Random value RV|M = h(M|RV)
= Hash value.

— Random value & h(M|RV)

(Hash value will be different) New Hash value

* Salt Value — Random value appended to the password before hashing - To provide security against dictionary attacks -

(Q/A)

1) What is hashing?

2) What is the use case of a hashing?

Software | download

File — Authentic — Hash value File — Data — bits

File : .exe : Authentic file — windows.exe (Real)

windows.exe → Hash <✗> SHA : Hash value

— ✓ — File is authentic