

INFI Academy

Strong password authentication

Aannames

- Het netwerk (tussen de client en server) kan worden afgeluisterd door derden.
- Informatiestromen op het netwerk kunnen (volledig) worden gemanipuleerd door derden.
- We beschikken niet over een Trusted Third Party, zoals een PKI, certificate authority of arbitrator.

Praktische problemen wachtwoord

- Wachtwoord is in de praktijk beperkt qua lengte en complexiteit
- Gebruikers herbruiken het wachtwoord voor meerdere systemen.

Voorbeelden

Weak password authentication

- Cleartext (bijv. http post, telnet)
- Encoded (bijv. http basic authentication)
- Hashed (bijv. client-side md5)
- Challenge-response (bijv. http digest, ntlm)

Challenge-response is weak??

- Wachtwoord gaat niet plain text over het netwerk -> beschermd
- Server challenge verandert, dus een captured response is onbruikbaar in nieuwe sessies -> geen replay attack mogelijk.
- maar: Man-in-the-middle (MITM) attack mogelijk.
- maar: Offline dictionary attack mogelijk als challenge en response bekend zijn.
- maar: server moet wachtwoord plain text (equivalent) opslaan.

Pseudo-Strong Authentication

- Public key authentication (voorbeeld: SSH pubkey, SSL/TLS certificates)
- maar: private key kan lekken, bijvoorbeeld via backups of gebruikersfout. Wachtwoord encryptie key is dan de enige resterende beveiliging. Ideaal target voor offline dictionary attack.
- Encrypted password authentication (voorbeeld https inlog, SSH password authenticatie)
- maar: wachtwoord komt plaintext (equivalent) aan op server. Rogue ssh daemon -> potentieel verstekkende gevolgen.

Feature list

Strong password authentication

- An attacker with neither the user's password nor the host's password file cannot mount a dictionary attack on the password. Mutual authentication is achieved in this scenario.
- An attacker who captures the host's password file cannot directly compromise user-to-host authentication and gain access to the host without an expensive dictionary search.
- An attacker who compromises the host does not obtain the the password from a legitimate authentication attempt.
- An attacker who captures the session key cannot use it to mount a dictionary attack on the password.
- An attacker who captures the user's password cannot use it to compromise the session keys of past sessions.

SRP

- Strong Remote Password Protocol
- We gaan 'ruiken' aan SRP:
 - geen wiskundige bewijzen,
 - we volgen het protocol om toch een magisch begrip te krijgen hoe SRP de voorgaande doelstellingen realiseert
 - Onbevredigend, maar beter dan niets ;)
- In het volgende heeft Steve de rol van een server, Carol de rol van een gebruiker.

Wachtwoord instellen

- To establish a password P with Steve, Carol picks a random salt s , and computes:
 $x = H(s, P)$
 $v = g^x \pmod{n}$
[N.B. het is zeer lastig om x te herleiden uit v , Discrete logarithms zijn computationally difficult voor grote n]
- We noemen v de password verifier.
- Carol geeft (s, v) en Steve slaat deze op als wachtwoordfile entry ('Carol', s, v)

Authenticatie I

- Carol stuurt gebruikersnaam 'Carol'
- Server stuurt salt s naar Carol
- Carol genereert een (private,public) pair $(a, A=g^a)$ en communiceert A naar Steve.
- Steve genereert een (private,public) pair $(b, B=v + g^b)$. [v is de password verifier] en stuurt B naar Carol
- Steve genereert een willekeurig getal u en deelt deze met Carol

Authenticatie II

- Neem aan dat $(B - g^x)^{(a + ux)} = S = (A \cdot v^u)^b$ (sorry)
Carol en Steve kunnen dan $K = H(S)$ uitrekenen, met H een hash functie
- Carol bewijst dat zij S weet: ze stuurt
 $M1 = H(A | B | K)$ naar Steve, procedure stopt als M1 fout is
- Steve bewijst dat hij S weet: hij stuurt $M2 = H(A, M1, K)$ naar Carol
- Carol en Steve kunnen K gebruiken als key voor een versleutelde verbinding voor hun sessie.

Bewijsbare beweringen I

- No useful information about the password P or its associated private key x is revealed during a successful run. Specifically, we wish to prevent an attacker from being able to guess and verify passwords based on exchanged messages.
- No useful information about the session key K is revealed to an eavesdropper during a successful run. Since K is a cryptographically strong key instead of a limited-entropy password, we are not concerned about guessing attacks on K , as long as K cannot be computed directly by an intruder.
- Even if an intruder has the ability to alter or create his own messages and make them appear to originate from Carol or Steve, the protocol should prevent the intruder from gaining access to the host or learning any information about passwords or session keys. At worst, an intruder should only be able to cause authentication to fail between the two parties (often termed a denial-of-service attack).

Bewijsbare beweringen II

- If the host's password file is captured and the intruder learns the value of v , it should still not allow the intruder to impersonate the user without an expensive dictionary search.
- If the session key of any past session is compromised, it should not help the intruder guess at or otherwise deduce the user's password.
- If the user's password itself is compromised, it should not allow the intruder to determine the session key K for past sessions and decrypt them. Even present sessions should at least be protected from passive eavesdropping.

Meer informatie

- <http://srp.stanford.edu>
- http://en.wikipedia.org/wiki/Euler%27s_theorem