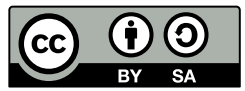
Tinklai; SSH

Saulius Gražulis

2009 ruduo

Vilnius University, Faculty of Mathematic and Informatics
Institute of Informatics



Trumpa istorija

- The telnet protocol, enabling logging on to a remote computer, was published as a Request for Comments (RFC) in 1972.
- The ftp protocol, enabling file transfers between Internet sites, was published as an RFC in 1973.

http://www.walthowe.com/navnet/history.html
2009-12-22

- Pagrindinės savybės:
  - nešifruotas duomenų perdavimas atviru kanalu;
  - autentikacija paremta slaptažodžiais, perduodamais atviru tekstu.

SSH ir OpenSSH

- SSH (Secure SHell) buvo sukurtas kaip alternatyva ankstesniems telnet ir ftp protokolams, siekiant apsisaugoti slaptažodžius, prisijungimo vardus bei perduodamus duomenis nuo perėmimo (eavesdropping, connection hijacking) ar „šniukštinėjimo“ (“sniffing”)

  http://en.wikipedia.org/wiki/Secure_Shell
  2010-01-06

- Duomenų saugumas ir vientisumas užtikrinamas naudojant patikimus šifravimo algoritmus.

- OpenSSH yra laisva SSH protokolo realizacija

  http://www.openssh.com/
  2010-01-06

SSH galimybės

`ssh` – „šveicariškas peiliukas“

- SSH protokolą galima naudoti, norint prisijungti prie nutolusio kompiuterio ir valdyti jame veikiantį komandų interpretatorių (shell); komanda: `ssh`;
- OpenSSH ir kitos realizacijos turi programas duomenims iš nutolusių kompiuterių saugiai nukelti (komanda `sftp`) bei duomenims tarp kompiuterių kopijuoti (komanda `scp`);
- `ssh` turi galimybę perduoti saugiu (šifruotu) kanalu X11 langų sistemos sujungimus ir tuo būdu leisti saugiai naudoti grafines (GUI) programas nutolusiame kompiuteryje;
- daug programų pačios naudoja ssh duomenims saugiai perduoti šifruotu „tuneliu“ – pvz. `rsync`, `svn` (Subversion klientas)
- `ssh` leidžia organizuoti šifruotus bet kokio protokolo tunelius tarp dviejų kompiuterių (`ssh -R`, `ssh -L` komandos).

Saugaus duomenų perdavimo principai – šifravimas

Nuo visiškai saugių iki visiškai nesaugių sistemų – vienas žingsnis...

- Tegul M yra pranešimas, K yra slaptas raktas, C yra šifrograma. Reikalinga funkcija $E()$, tokia, kad:

$$C = E(M, K)$$

$$M = E(C, K)$$

yra lengvai suskaičiuojamos, žinant slaptą raktą ir vieną iš pranešimo variantų (M arba C), bet nežinant K, $E(C, ?)$ nustatymui reikia labai daug skaičiavimų, praktiškai tiek, kiek reikia visų galimų raktų perrinkimui.

Istorinis slapto rakto šifro pavyzdys

Nuo seno šifrai buvo naudojami karyboje ir diplomatijoje...

- Cezario šifras: pakeiskim kiekvieną abėcėlės raidę kita, nutolusia nuo jos duotu atstumu (šis nuotolis yra slaptas raktas):

$$A \to C, B \to D, \ldots, Z \to B$$

- pavyzdys:
  M = PULSIMRYTOJ, K = 2
  C = RWNUKOTAVQL

```
perl -le '$key = 2; print map \
  {chr((ord($_)-ord("A")+$key)%26+ord("A"))} \
  split("",$ARGV[0])' PULSIMRYTOJ
```

- Cezario šifro variantai: ROT13, XOR
  šie primityvūs „šifrai“ šiandien tik iliustruoja principą, bet mūsų dienomis nesuteikia jokio saugumo!

Absoliučiai saugus šifras

Taip, tai įmanoma!

- „Bloknoto šifras“ (one-time pad cipher):
  http://en.wikipedia.org/wiki/One-time_pad
  2010-01-06

- Šifravimas panašus į Cezario šifrą, bet kiekvienai teksto raidei imamas savas, visiškai atsitiktinis postūmis (paprastai užrašomas kaip rakto raidė, kur *A* reiškia jokio postūmio, *B* reiškia postūmį per vieną raidę ir t.t). Pvz.:

```
M = ATVYKSTAMEPORYTPRIESAUSRA
K = YPNZZDPIXDZEEQEZBNJYYSFMS
C = YIIXJVIIJHOSVOXOSVNQYMXDS
```

```
perl -e '@m=split("",$ARGV[0]);@k=split("",$ARGV[1]);
    for (@m){
        print chr( (ord($_)+ord(shift(@k))-2*ord("A")) % 26 + ord("A") )
    }
    print "\n"'
    YIIXJVIIJHOSVOXOSVNQYMXDS YPNZZDPIXDZEEQEZBNJYYSFMS
```

Bet su sąlyga...

Nuo visiško saugumo iki visiško nesaugumo – vienas žingsnis...

- bloknoto šifras yra visiškai saugus ta prasme, kad gauta šifrograma neduoda jokios papildomos informacijos apie pranešimą, jei neturime rakto.
- metodas saugus su sąlyga, kad raktas yra visiškai atsitiktinis, rakto ilgis lygus pranešimo ilgiui ir raktas naudojamas vieną vienintelį kartą;
- panaudojus, tyčia ar netyčia, raktą kelis kartus šifras tampa **labai lengvai įveikiamu!**
- dėl didelės rakto apimties šis metodas yra nepraktiškas ir dėl to retai kada naudojamas. Be to, jis neatsparus kitoms atakoms (pvz. teksto pakeitimo) ir neduoda autentikacijos.

Šiuolaikiniai simetriniai šifrai

Saugūs skaičiavimų prasme (computationally secure)

- Said of a cipher that cannot be broken with the current computer technology within a period short enough to be practicable.

  http://www.businessdictionary.com/definition/computationally-secure.html
  2010-01-06

  A coding technique based on CRYPTOGRAPHY which cannot be broken using available technology in such a time that some gain, financial or otherwise, can be made.

  http://www.encyclopedia.com/doc/1O12-computationallysecure.html
  2010-01-06

- šia savybe, manoma, pasižymi šiuolaikiniai šifrai: 3DES, TWOFISH, BLOWFISH, AES, jei naudojami pakankamai ilgi raktai...

Raktų paskirstymo problema

- Reikia saugiai pristatyti bendrą slaptą raktą abiem bendraujančioms šalims, pvz. serveriui ir darbo stočiai (terminalui)
  - šią problemą galima išspręsti Difi-Helmano (Diffie–Hellman) protokolu;
- Reikia įsitikinti, kad bendraujanti šalis (serveris ar terminalo naudotojas) yra būtent tie, kurių tikimės (autentikacijos problema).
  - Ši problema išsprendžiama, panaudojant asimetrinę, viešo rakto kriptosistema (public key cryptographic system).

Asimetriniai šifrai (kriptosistemos su viešais raktais)

- Sistema su dviem raktais: K_p – viešas (public) raktas, K_r – slaptas (private) raktas.
  Funkcijos:

$$C = E(K_p, M), \quad S = E(K_r, M)$$
$$M = E(K_r, C), \quad M = E(K_p, S)$$

  turi būti lengvai suskaičiuojamos, bet nežinant K_r, turi būti labai sunku atstatyti M arba K_r, turint vien tik C arba S ir K_p. Taip pat turi būti sunku sukurti pakeistą M', kad nepasikeistų „skaitmeninis parašas“ S[1]:

$$S = E(K_r, M')$$

[1] Realiose kriptosistemose parašui naudojama pranešimo santrauka, gaunama vienakrypte funkcija $H()$: $S = E(K_r, H(M))$, bet mes dabar į tai nesigilinsim.

Autentikacija ir skaitmeninis parašas

- Užšifruota slaptu raktu K_r atsitiktinė (nenuspėjama) žinutė, perduota saugiu kanalu, įrodo, kad kanalu naudojasi slaptojo rakto savininkas (nes niekas negali nei perimti užšifruotos žinutės, nei atlikti šifravimo, nežinodami slaptojo rakto).
  Tuo būdu, naudodami asimetrinę kriptosistemą, galime įsitikinti, kad „šnekamės“ su slaptojo rakto valdytoju, taigi galime autentikuoti korespondentą.
- SSH sistemos naudoja šią galimybę serveriui autentikuoti, ir gali naudoti ją sistemos vartotojo tapatybei nustatyti – vietoj slaptažodžio.

Kriptografiniai elementai OpenSSH

Šifruokimės...

- OpenSSH yra laisva atviro kodo SSH protokolo realizacija.
  http://www.openssh.com/
  2010-01-06
- SSH naudoja RSA (Rivest, Shamir and Adleman) ir DSA (Digital Signature Algorithm) autentikacijai (tiek serverio, tiek, jei reikia naudotojo), ir 3DES, BLOWFISH, CAST128, Arcfour arba AES simetrinius algoritmus duomenų šifravimui.
  http://www.openssh.com/faq.html
  2010-01-06

Serverio autentikavimas OpenSSH – pirmas prisijungimas

Su kuo mes kalbamės?

- Serveris autentikuojamas pagal jo viešą raktą `/etc/ssh/ssh_host_rsa_key.pub` arba `/etc/ssh/ssh_host_dsa_key.pub`; atitinkami slapti raktai yra `/etc/ssh/ssh_host_{rsa,dsa}_key`. Šie raktai paprastai sugeneruojami, pirmąkart paleidžiant OpenSSH serverio demoną `sshd`.
- Pirmą kartą jungiantis prie serverio, `ssh` programa (klientas) pateikia naudotojui **serverio rakto skaitmeninę santrauką**[2] ir paklausia, ar ji tikrai priklauso nutolusiam serveriui. Idealiu atveju reikėtų nepriklausomu kanalu (telefonu, SMS, asmeniškai susitikus su administratoriumi) patikrinti, ar tikrai raktas priklauso serveriui, o ne „žmogui viduryje“.

[2] Santrauka – tai eilutė, gauta vienakrypte (hash) funkcija iš serverio viešojo rakto. Vienakryptę funkciją lengva suskaičiuoti, žinant raktą, bet

Serverio autentikavimas OpenSSH – vėlesni prisijungimai

- Visiems vėlesniems prisijungimams autentikuoti serverio rakto santrauka įrašoma į failą `~/.ssh/known_hosts`
- Jei serverio raktai pasikeičia, `ssh` klientas nesijungia prie tokio serverio, o praneša apie pasikeitusius raktus. Jei raktų pakeitimas teisėtas, būtina ištrinti eilutę su sena rakto santrauka iš failo `~/.ssh/known_hosts` ir pakartoti pirmojo prisijungimo procedūrą.
- *Pasikeitęs raktas gali byloti apie tai, kad vykdoma kriptografinė ataka – įrengtas „priešiškas“ serveris, pasisavinęs tikrojo serverio IP numerį arba vardą (ataka „žmogus viduryje“, vardų serverio padirbimas arba „ARP kešo apnuodijimas“).*

Naudotojo autentikavimas SSH viešų raktų pagalba

- Analogiškai serverio autentikavimui, kiekvienas naudotojas gali susikurti savo slapto/viešo rakto porą (vadinamą „ssh sertifikatą“).
- Raktų pora sugeneruojama komanda `ssh-keygen`
- Raktai paprastai patalpinami failuose `~/.ssh/id_rsa` (slaptas raktas) ir `~/.ssh/id_rsa.pub` (viešas) raktas, jei naudojamas RSA algoritmas, ir failuose `~/.ssh/id_dsa` bei `~/.ssh/id_dsa.pub`, jei naudojamas DSA algoritmas.
- Norint prisijungti prie nutolusio serverio, naudojant šiuos raktus, reikia **viešą** raktą pridėti prie failo `~/.ssh/authorized_keys` **serveryje**.
- **Slaptą raktą reikia saugoti – niekur nekopijuoti bei laikyti piktadariams neprieinamoje, tamsioje ir vėsioje vietoje!**

SSH raktų agentas

- Saugumo sumetimais slaptas SSH raktas yra papildomai šifruojamas tik savininkui žinoma slapta fraze.
- Kad šios frazės nereikėtų įvedinėti kiekvieną kartą jungiantis prie nutolusio serverio, galima iššifruotą raktą įkelti į SSH agentą komanda `ssh-add`. Ši komanda slaptos frazės paprašys tik vieną kartą.
- Šiuolaikinės Linux sistemos SSH agentą (`ssh-agent`) paleidžia automatiškai, įsijungus į sistemą, ir gali panaudoti slaptažodį raktui iššifruoti.
- Jei reikia, agentą galima paleisti komanda `eval` `` `ssh-agent` ``.
- Tuo būdu, naudojant šifruotus SSH raktus ir agentą, gaunamas labai patrauklus saugumo ir patogumo derinys.
- Visgi, sistemos administratorius gali išjungti SSH raktų naudojimą (tiek serverio, tiek kliento pusėje), jei tai neatitinka sistemai keliamų reikalavimų.