Tietoturvan perusteet, harjoitus 3, 12.-16.4. 2010

1. Esittele haittaohjelma *rootkit*. Mihin haittaohjelmaluokkaan se kuuluu vai muodostaako se oman luokkansa? Miten se havaitaan ja miten se hävitetään?

2. Käy läpi CERT-FI:n haavoittuvuuslistaa ja etsi viisi viimeisintä haavoittuvuutta. Esittele jokainen haavoittuvuus ja pohdi, mitä implementointisääntöä tai hallintosääntöä haavoittuvuuden yhteydessä on rikottu (jos tapauksesta on saatavissa tarpeeksi tietoa sellaisiin johtopäätöksiin).

3. Lue SANS-instituutin sivuilta kohdasta 20 Most Critical Security Controls erityisesti kohta Critical Control 3: Secure Configurations for Hardware and Software on Laptops, Workstations, and Servers (http://www.sans.org/critical-security-controls/). Tee yhteenveto ja vertaa ohjeita luentomateriaalin lukuun Haittaohjelmien torjunta. Mitä yhteneväisyyksiä ja eroja löydät?

4. Oletetaan, että ohjelma käyttää tietyissä operaatioissa juurioikeuksia, jotka on asetettu setuid-komennolla. Näyttäisi siltä, että nämä operaatiot voitaisiin ohjelmoida myös palveluina. Silloin ohjelma autentikoisi käyttäjän, ottaisi yhteyden palvelimeen, lähettäisi käskyn ja roolin ja antaisi palvelimen toteuttaa käskyn.

   a) Mitkä olisivat tämän vaihtoehtoisen ratkaisun edut verrattuna alkuperäiseen ratkaisuun?

   b) Jos palvelin vastaa vain paikallisen koneen asiakkaille käyttäen prosessien välistä kommunikointia, kumpaa lähestymistapaa käyttäisit? Miksi?

   c) Jos palvelin toimisi verkon yli, muuttaisiko tämä b-kohdan vastaustasi? Miten ja miksi?

   d) Jos asiakas lähettää salasanan palvelimelle ja palvelin autentikoi, muuttuisivatko edellisten kohtien vastauksesi? Miksi tai miksi ei?

5. Implementointisäännön 6 jälkeen luentomateriaalissa tarkasteltiin mahdollisuutta torjua puskurin ylivuotohyökkäyksiä satunnaisluvun (canary) avulla. Tässä tapauksessa satunnaisluku vietiin aliohjelmapinoon. Tarkastellaan nyt tilannetta, jossa satunnaisluku viedään taulukkoon.

   a) Oletetaan, että satunnaisluku sijoitetaan heti taulukkoa seuraavaan muistipaikkaan ja että jokaisen taulukko-operaation jälkeen tarkistetaan, onko satunnaisluku muuttunut. Löydettäisiinkö tällä tavalla puskurin ylivuoto? Jos löydettäisiin, miksi menetelmää ei käytetä käytännössä? Jos taas ei löydettäisi, niin kuvaa hyökkäys, joka muuttaisi muistipaikkaa taulukon jälkeen muuttamatta satunnaislukua.

   b) Oletetaan, että satunnaisluku viedään heti taulukon jälkeen, mutta tarkistus tapahtuisi vain juuri ennen funktion paluuta. Olisiko tämä tapa tehokas?