Eemil Aspholm

CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi

Viikkotehtävä 5.

Tehtävä 1.

A bug hunter’s diary kirjassa kirjan kirjoittaja testaa erilaisia bugeja erilaisista ohjelmista ja kyseisessä kappaleessa hän käsittelee VLC media player ohjelmistoa. Kirjoittaja tutki ohjelmiston lähdekoodia ja lähti tarkastelemaan mahdollisia haavoittuvuuksia ja bugeja, joita lähdekoodi saattaisi sisältää. Noin puolen päivän tutkimisen jälkeen hän löysi haavoittuvuuden, pinopuskurin ylivuodon. Tämä aiheutui, kun ohjelman lähdekoodissa tietyntyyppisellä media tiedostomuodolla (.TIVO) ohjelma kirjoittaa kutsupinon muistiosoitteen aiotun tietorakenteen ulkopuolelle, joka on suuruudeltaan kiinteän kokoinen. (Pinon puskuriin kirjoitettiin enemmän tietoa, kun puskurille oli varattu).

Haavoittuvuuden kirjoittaja löysi luomalla ensin listan VLC mediaplayerin demuxereista (prosesseista, jotka erottavat äänen ja videon ja muun datan streamista, jotta video voidaan toistaa), identifioimalla syötteen, jota demuxerit käsittelevät, ja seuraamalla syötteen viitteitä (missä demuksereita käytetään). Lopputuloksena haavoittuvuuden löytäjä ilmoitti VLC mediaplayerille löydetystä haavoittuvuudesta, joka saatiin kahden yrityksen jälkeen korjattua. Lopulta VLC mediaplayer julkaisi uuden hotfixin ja käyttäjä julkaisi myös postauksen internettiin tietoturva haavoittuvuudesta.

Kuten aiemmassa kappaleessa mainittiin kyseessä, on pinopuskurin ylivuoto hyökkäys. Kyseisessä hyökkäyksessä pinolla on rajallinen koko, ja jos käyttäjän syöte ylittää tuon varatun muistimäärän, eikä ohjelma tarkista, että syöte mahtuu pinoon, tapahtuu pinon ylivuoto. Pinon ylivuoto ei itsessään ole suuri ja vaarallinen ongelma, mutta siitä tulee iso tietoturvariski, jos käyttäjän syöte on tarkoitusperältänsä haitallinen. Mikäli syöte on haitallinen voi se korruptoida pinon ”ruiskuttamalla” suoritettavaa koodia käynnissä olevaan ohjelmaan, ja ottaa tällä tavalla kontrollin prosessista. Tällä tavoin syötteen antajalla on mahdollisuus saada luvattomasti pääsy tietokoneeseen.

Tehtävä 2.

Ohjelmakoodi:

Text

Description automatically generated

Kun kääntäjällä suoritetaan ohjelma, jossa varataan muistia malloc käskyllä, mutta joka ei vapauta varattua muistia ennen ohjelman lopetusta ei tapahdu virheilmoitusta, tai muuta ohjelman suorittamista haittaavaa ajon aikana. Gdb käskyä ei voi käyttää ongelmien löytämiseen, sillä se ei pysty ilmoittamaan muistivuotoja ohjelmassa. Valgrind puolestaan ilmoittaa muistivuodoista ohjelmassa seuraavan kuvan avulla:

Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Ohjelmaa nro 2 suoritettaessa ei myöskään tule ongelmia (suoritus / kääntäminen). Kuitenkin valgrind ilmoittaa virheellisestä kirjoituksesta ja muistivuodosta ohjelmassa.

Text

Description automatically generated

Tehtävä 3.

Ohjelmakoodi:

Text

Description automatically generated

Ohjelmaa suorittuu ja kääntyy normaalisti ilman ongelmia, lisäksi ohjelma tulostaa terminaaliin numeron 0, vaikka muistipaikat ovat vapautettu ennen tulostamista. Valgrind ilmoittaa puolestaan ettei muistivuotoja ole havaittu, mutta syntaksista löytyy silti lukuvirhe. Alla kuva valgrindin tulosteesta:

Text

Description automatically generated

Muokattaessa ohjelmaa vapauttamaan aluksi osoitin listan kuudenteen jäseneen, ohjelma kääntyy, mutta suorittaessa tulee vastaan ongelma. Suorittaessa tapahtuu suoritusaikainen virhe. Tällaisen löytämiseen ohjelmistokoodista tarvita gdb:tä erikoisempia työkaluja, sillä se ilmoittaa ohjelmaa käännettäessä terminaaliin, mikäli kääntämiskäskyn oheen lisättäisiin erilaisia virhetestauksia.

A screenshot of a computer

Description automatically generated with medium confidence

Vektori on tietorakenne, jonka alkioiden määrää voi muuttaa ohjelman suorituksen, vahvuutena vektorilla voidaan katsoa olevan etenkin alkioiden poistaminen tietorakenteen lopusta, satunnainen alkioiden lukeminen tietorakenteesta, sekä alkioiden lisääminen rakenteen loppuun. Linkitetyt lista puolestaan ovat hyviä alkioiden lisäämiseen tietorakenteen alkuun tai loppuun vakioajassa, mutta esimerkiksi linkitetyn listan keskeltä alkion poistaminen ei ole kovinkaan tehokasta (ensin täytyy löytää haluttu alkio ja tämän jälkeen suorittaa halutut operandit). Yhteenvetona vektorit ovat parempi vaihtoehto dynaamiselle tietorakenteelle, mikäli säilöttävänä ei ole miljoonia olioita.

Alla vector.c ohjelman suorittaminen Valgrindin avulla:

Text

Description automatically generated

Ohjelmakoodi, jossa lähteenä käytetty:

https://github.com/martalist/ostep/blob/master/chapter\_13/hw13/vector.c

Text

Description automatically generated with medium confidenceText

Description automatically generated