MOŽNOSTI ZAŠČITE PROGRAMSKE OPREME NA MIKRORAČUNALNIKIH

INFORMATICA 1/88

Nikolaj Zimic
Jernej Virant
Marjan Bradeško
Ivan Pepelnjak
Fakulteta za elektrotehniko, Ljubljana

UDK 681.3.06:003.6

Izvleček :

V članku so opisane možnosti zaščite programske opreme na obstoječih mikroračunalnikih. Predstavljene so najpogosteje uporabljene zaščite, stopnja zaščite, ki jo nudijo, napadi nanje in obramba programske opreme pred napadom.

Abstract :

The paper presents possible protection of software on existing microcomputers. The most often used protection schemes are explained, together with level of protection they enable, the attacks on these schemes and the possibilities of software defense against such attacks.

1. NAČINI ZAŠČITE PROTI KOPIRANJU

1.1 Uvod

Nedovoljeno kopiranje programov na osebnih in hišnih računalnikih se je začelo s kopiranjem igric. Te zaščite so prebijali mladi nadebudneži, ki so to počeli bolj iz zadovoljstva pred prepovedanim kot pa iz ekonomskih razlogov.

S povečanjem moči osebih računalnikov se je pojavila tudi zahtevna programska oprema, ki lahko stane tudi nekajkrat več kot strojna oprema. Zato je zaščita programov postala resen problem.

Z množičnim prodorom osebnih računalnikov se je razmahnilo tudi nedovoljeno kopiranje ali bolje rečeno kraja programov. S krajo programov je povzročena ogromna škoda avtorjem in proizvajalcem teh programov. Avtorji so se s problemom nedovoljenega kopiranja spopadli na različne načine:

- pri programih z velikim številom prodanih kosov so znižali ceno, tako da kraja programa postane skoraj nesmisel.
- programom so vzdignili ceno, tako da so z nakupom enega kosa plačane tudi morebitne kopije.
- programe so zaščitili na razne načine in tako otežili, če že ne preprečili kopiranje.

Pri ukradenih programih se pojavljajo še dodatni problemi, kot so garancija in vzdrževanje programov. Pri ukradenih programih ni vzdrževanja, zato so neuporabni za resno delo.

1.2 Problemi zaščite programov proti kopiranju na osebnih računalnikih

Pri osebnih računalnikih ni več običajnih zaščit programske opreme, ki so prisotne na velikih računalnikih. Pri osebnih računalnikih ima uporabnik vse privilegije in ima dostop do celotnega pomnilnika in do vseh perifernih enot. Procesor tudi nima svoje serijske številke, po kateri bi programska oprema ločila računalnike med seboj.

Samo kopiranje mora biti s strani sistema dovoljeno, da si lahko uporabnik naredi 'back up' programske opreme. Te zmožnosti omogočajo uporabniku tudi nedovoljeno kopiranje pogramske opreme.

Rezultat je enostavno prenašanje programske opreme z enega računalnika na drug računalnik. Programska oprema pač nima možnosti ugotoviti na katerem računalniku se izvaja.

1.3 Načini zaščite programske opreme proti kopiranju

Proizvajalci programske opreme so začeli s programskim paketom prodajati še dodatek, ki ga je zelo težko prekopirati. Pri cenejših verzijah programske opreme je to kar disketa, ki se razlikuje od običajne diskete in jo programska oprema lahko preverja. To so na primer poškodbe magnetnega materiala na točno določenem mestu ali pa nestandarden zapis na disketi.

Pri dražjih paketih se k programski opremi doda modul. Modul se običajno priključi na zunanje vhode. Programska oprema je pisana tako, da brez modula ne deluje. Modul je običajno zalit v epoksi smolo, kar otežuje kopiranje modula.

1.3.1 Zaščita na disketi

Disketa je običajno medij, na katerem se prodaja program. Ista disketa lahko služi še kakor ključ, brez katerega program ne deluje pravilno. Tak način zaščite je dokaj poceni, saj zahteva samo spremembo na disketi.

Običajna disketa je organizirana po sledeh, to je 40 ali 80 koncentričnih krogov. Vsaka sled je logično razdeljena na sektorje. Pred vsakim sektorjem je poseben zapis (id field), v katerem je kodirana številka sledi, številka strani, številka sektorja in dolžina sektorja. S strani procesorja se v kontroler gibkega diska zapiše samo številka sektorja, sledi in strani, kontroler pa sam poskrbi, da se bodo podatki pisali ali brali iz pravilnega mesta na disketi. Dodatni zapisi, ki skrbijo za pravilen dostop do sektorja, so običajno procesorju nevidni, lahko pa se jih z direktnim dostopom do kontrolerja prebere.

Na disketo se lahko poleg običajnih informacij zapiše dodaten id field, ki pa mu ne sledi sektor. To pomeni, da je na disketi podatek o dodatnem sektorju, ki pa ne obstaja. Pri normalni uporabi diskete z dodanim id field-om se le tega ne opazi. Služi samo za kontrolo pri kopiranju. Pri običajnem formatiranju se dodaten id field ne zapiše, kar programski paket kasneje preveri in 'ugotovi', da ni pognan iz originalne diskete.

Naslednja možnost je zapis ene sledi več kot je običajno. Disketne enote običajno omogočajo formatiranje diskete na večje število sledi, kot je standardno. Ker operacijski sistem dodatnih sledi ne uporablja, so le te iz uporabniškega stališča povsem nevidne. Enostavno pa se lahko uporabijo za shranjevanje dodatnih informacij, ki služijo za zaščito proti kopiranju.

Zaščita diskete je možna tudi z odstranitvijo magnetnga materiala z diskete. To se običajno naredi z laserjem. To so tako imenovane laserske luknje. Glavna prednost laserskih lukenj pred raznimi zapisi na disketi je v težjem kopiranju, saj je potreben fizičen poseg na disketo. Programska oprema preverja napako na disketi. Če napaka ni na pričakovanem mestu, programska oprema ne deluje pravilno. Preverjanje napake se izvede s pisanjem in branjem sekvence podatkov na poškodovan sektor. Ker je magnetni medij poškodovan, so prebrani podatki spremenjeni. Seveda so podatki spremenjeni točno na določenem mestu.

Boljša metoda zaščite je z dvema laserskima luknjama na istem sektorju. To pomeni, da imamo na enem sektorju dve točno določeni napaki. S pisanjem in branjem poškodovanega sektorja lahko izmerimo razdaljo med luknjami na približno 5% natančno. Natančnost meritve nam določa hitrost vrtenja diskete, ki pa ni popolnoma enaka pri vseh disketnih enotah. Dolžina se izmeri s preštevanjem pravilno prebranih zlogov med dvema napakama. Na mestu napake disk kontroler izgubi sinhronizacijo, zato so lahko podatki med dvema napakama pomaknjeni v levo ali desno. Programska oprema tako preverja, če je napaka na sektorju in če je razdalja med dvema napakama pravilna.

Disketo, ki je zaščitena z lasersko luknjo, lahko poljubno formatiramo, ne da bi izgubili ključ zaščite. To pomeni, da si lahko naredimo poljubno število kopij, ki služijo kot back up. Te kopije pa so neuporabne brez originalne diskete, ki je zaščitena z lasersko luknjo. Programsko opremo lahko prekopiramo na trdi disk, seveda pa ne deluje brez originalne diskete.

1.3.2 Zaščita z dodatnim modulom

Pri dražjih paketih se poleg programske opreme doda modul. Modul se običajno priključi na serijski, paralelni vmesnik ali med tipkovnico in računalnik.

Nekateri novejši moduli se priključujejo med računalnik in tiskalnik. Ta modul se sproži ob točno določeni sekvenci in ne moti običajnega dela s tiskalnikom.

V mudulu je običajno nekaj integriranih vezij, ki so zalita v smolo. Ta modul je za uporabnika črna škatla, brez katere programska oprema ne deluje.

Modul je običajno avtomat, ki na sprejeto sekvenco podatkov odgovori s točno določeno sekvenco, ki jo preverja programska oprema. Sekvenca se običajno generira s psevdo naključnim generatorjem.

Zaščita je s stališča kopiranja modula dokaj zanesljiva, saj je pri daljši sekvenci praktično nemogoče ugotoviti način prekodiranja vhodne v izhodno sekvenco.

1.3.3 Zaščita programskih paketov na trdem disku

Določeno programsko opremo je možno instalirati na trdi disk, tako da le ta ne potrebuje diskete s ključem za normalno delo.

Običajno se programsko opremo instalira na trdi disk z zaščiteno disketo. Na disketi se pri inštalaciji zmanjša števec tako, da je z eno zaščiteno disketo možno instalirati samo omejeno število kopij. Števec inštalacij se lahko poveča samo v primeru, da se po posebnem postopku onemogoči prej instalirano kopijo na trdem disku.

Tak način dela omogoča, da se inštalacijsko disketo, ki je zaščitena, uporablja samo pri inštalaciji. Drugače pa ta disketa služi kot back up.

Pri inštalaciji na trdi disk se običajno v programsko opremo zapišejo informacije o fizičnem položaju le te na trdem disku. Pri kopiranju je praktično skoraj nemogoče prekopirati programsko opremo na fizično enake lokacije. Prazen prostor na trdem disku, ki se lahko uporabi za kopiranje, se spreminja pri vsakem brisanju ali pisanju na disk. Tako je samo pri praznem disku enostavno določiti, kam se bodo podatki pri kopiranju pisali.

1.4 Problemi uporabnika pri zaščitenih programskih paketih

Zaščita programskih paketov lahko pri nepreviđni zaščiti postane dvorezen meč za legalnega uporabnika. To pomeni, da se lahko zaščita obrne proti uporabniku, ki je programski paket kupil.

Problemi nastanejo, če se poškoduje disketa, ki služi kot ključ za pravilno delovanje programa. To pomeni avtomatsko nekajdnevno neuporabnost programskega paketa. Proizvajalci namreč zamenjajo poškodovano disketo, na kateri je zakodiran ključ, z novo.

Problemi se pojavljajo tudi pri dodatnih modulih. Če je modul slabo priključen ali če pride do napake pri komunikaciji z modulom, se lahko sproži zaščita, ki je vgrajena v programski paket. To ima za posledico prekinitev programa ali pa njegovo nepravilno delovanje, kar pa zmanjšuje zanesljivost programske opreme.

Zaradi zmanjševanja zanesljivosti delovanja programskega paketa se običajno zaščita postavi samo na začetek programa. To seveda olajšuje napad na zaščito in zmanjšuje zanesljivost zaščite.

2. NAPADI NA ZAŠČITENE PROGRRAME

2.1 Možnosti 'popravljanja' zaščitenih programov

Pri zaščitenih programih vedno obstaja rutina ali skupina rutin, ki preverjajo prisotnost zaščite. Če ključ, ki je na disketi ali pa v posebnem modulu, ne obstaja, se program običajno prekine ali pa ne deluje pravilno. Eden izmed možnih napadov na zaščito je onemogočitev prej omenjenih rutin.

Seveda pa je dokaj zapleteno najti rutine, ki tvorijo zaščito programa. Avtorji rutin seveda poskušajo te rutine čimbolj zakriti. Rutine imajo seveda nekaj značilnosti, ki olajšujejo iskanje.

Glavne značilnosti rutin, ki skrbijo za zaščito so:

- prve razlike med delovanjem programa pri originalnem ključu in brez, se pojavijo prav v njih.
- te rutine kličejo periferne enote, na katere je priključena zaščita (krmilnik gibkega diska, serijski in paralelni izhod, ...).
- včasih vektorji prekinitev kažejo na te rutine, posebno v primeru, če periferne enote delujejo pod prekinitvami.

Zanesljivost programske opreme, proti napadu se lahko poveča s testiranjem okolja. To pomeni, da programska oprema odkrije programe, ki nadzirajo delovanje programske opreme. Če program odkrije, da je 'opazovan' ima možnost, da zavede opazovalca.

2.2 Možnosti kopiranja zaščitenih disket

Diskete so običajno zaščitene na tri načine:

- z nestandardnim zpisom.
- s povečanjem števila sledi.
- z odstranitvijo magnetnega materiala (tako imenovane laserske luknje)

Običajni programi za kopiranje disket ne prepišejo skritih zapisov na disketi. Te lahko prekopirajo programi, ki 'pričakujejo' skrite zapise in naredijo popolno kopijo, ne glede na to, kaj je zapisano na disketi.

Dokaj enostavno je tudi prekopirati sledi, ki jih običajno ni. Seveda pa moramo prej odkriti, da je programska oprema zaščitena na ta način.

Če je disketa zaščitena z laserskimi luknjami, je le to dokaj težko prekopirati brez drage opreme. Seveda pa ni nujno, da poškodbo povzročimo z laserjem. To poškodbo se lahko povzroči z ostrim predmetom. Pomemben je le fizičen položaj poškodbe in njena širina.

Prva dva primera zaščite je možno prekopirati popolnoma programsko, brez fizičnega posega v računalnik ali na disketo. Prekopirajo jih tudi aparature, ki so narejene za hitro kopiranje programske opreme. Take aparature namreč ne preverjajo zapisa na disketi, ampak delajo popolno kopijo.

2.3 Možnosti programskega simuliranja zaščitene diskete

Napako na disketi, ki je povzročena z laserjem, je možno programsko simulirati. To pomeni, da programska oprema ne dobi odgovora z diska ampak iz posebnega programa, ki simulira napako.

Disketo z zaščito lahko programsko analiziramo in izdelamo algoritem po katerem se spremeni zapisani sektor. Če je na sektorju enojna napaka, je začetek sektorja pravilen, nato pa sledi mesto napake. Na mestu napake je prebrana vrednost nesmiselna. Po fizični napaki na disketi spet sledijo podatki, ki so bili zapisani. Ti podatki so zaradi izgube sinhronizacije lahko pomaknjeni. Premik je mišljen na nivoju bita.

Če je na enem sektorju več laserskih lukenj, pomeni, da imamo več področij, kjer so prebrani podatki popolnoma nedefinirani. Točno določena pa je razdalja med laserskimi luknjami. Razdalja se lahko izmeri s številom zlogov med dvema napakama.

Z laserskimi luknjami zaščiten program lahko prelisičimo s spremembo programa, ki bere sektor. To pomeni, da moramo napisati program, ki simulira zaščiten sektor. Ta program inštaliramo kot del sistema tako, da zaščitena programska oprema ne opazi spremenjenega branja diskete. Seveda pa se ta program aktivira samo v primeru branja in pisanja na sektor z laserskimi luknjami. V vseh ostalih primerih je branje in pisanje običajno.

2.4 Možnosti kopiranja dodatnih modulov

Dodatni moduli so običajno vezja, ki so zalita v epoksi smolo. To pomeni da nimamo vpogleda v notranjo zgradbo vezja. Sam algoritem je lahko zakodiran tudi v integrirana vezja, kot so naprimer PAL, EPLD, GAL vezja. Ta vezja lahko zaščitimo proti kopiranju s programiranjem posebnega bita v integriranem vezju. To pomeni, da ne moremo prebrati podatkov za programiranje. Podatke za programiranje lahko dobimo le z analizo delovanja integriranega vezja.

Analiza delovanja integiranih vezij oziroma modulov je dokaj zamudna zaradi velikega števila možnih kombinacij. To število lahko povečamo s številom notranjih stanj flip-flopov. Če imamo v modulu ali v integriranem vezju 10 flip-flopov, je število notanjh stanj 2 na 10. potenco.

Same module oziroma integrirana vezja je dokaj težko prekopirati. Lažje je odkriti rutine, ki uporabljajo modul in jih spremeniti.

2.5 Možnosti uporabe programskih debugerjev

Debuger je program, ki je namenjen ugotavljanju napak v programski opremi. Omogoča nam spremljanje izvajanja programa.

Tako lahko spremljamo izvajanje programa z zaščito in brez nje. Iz spremljanja lahko ugotovimo, kje je jedro zaščite in ga zaobidemo. Z debugerjem lahko iščemo inštrukcije, ki so vezane na delovanje zaščite. To so naprimer instrukcije za branje in pisanje kanala, na katerega je priključen modul za zaščito.

V opazovani program lahko postavimo prekinitvene točke, v katerih opazujemo stanje pomnilnika in registrov. Iz podatkov na skladu lahko ugotovimo, od kje je bil določen podprogram klican.

Opazujemo lahko tudi komunikacijo z dodatnimi moduli. Z dobljenimi podatki lahko napišemo program za analizo modula. Po drugi strani pa lahko odkrijemo algoritem, ki skrbi za preverjanje modula. Ta algoritem mora biti ekvivalenten algoritmu v modulu. Iz algoritma se lahko brez večjh problemov izdela modul.

Nekateri programski paketi imajo vgrajeno preverjanje okolice v kateri delujejo. To pomeni, da lahko odkrijejo, če delujejo pod kakšnim dodatnim programom, ki opazuje njihovo delovanje. Če odkrijejo, da so opazovani, se na ustrezen način 'branijo'. Spremenijo način delovanja tako, da kar najbolj otežijo spremljanje dogajanj v računalniku.

2.6 Možnosti uporabe emulatorjev

Emulatorji so najnevarnejši za zaščitene programe. Njihovo prisotnost je programsko nemogoče odkriti, ker sonda emulatorja zamenjuje procesor računalnika. Emulator omogoča v realnem času spremljanje programa oziroma dogajanj v pomnilniku ali perifernih enotah.

Z emulatorjem dokaj enostavno odkrijemo razliko v delovanju programa, ki ima ključ (modul ali zaščitemo disketo) ali je brez.

3. MOŽNOST UPORABE KRIPTOGRAFIJE

Pri običajnih programskih paketih je problem, da mora zaščita delovati tudi proti nedovoljeni uporabi legalnega uporabnika. To pomeni, da uporabnik programski paket normalno uporablja, ne sme pa ga prekopirati v smislu nedovoljene nadaljnje kopije.

Pri uporabi kriptografije mora biti ključ za uporabo programa skrit tudi pred legalnim uporabnikom. To pomeni, da mora biti skrit na disketi ali dodatnem modulu. Rezultat je, da je tak način zaščite popolnoma enakovreden prej opisanim načinom. Nekoliko se spremeni način napada, ki mu cilj postane iskanje ključa za dešifriranje programske opreme.

Druga možnost bi bila uporaba modula, v katerem bi bil vgrajen postopek in ključ za dešifriranje programske opreme. Tudi ta možnost je neuporabna, ker modul lahko služi za dešifriranje programske opreme, ki se dešifrirana lahko poljubno kopira.

4. ZAKLJUČEK

Zaščita programske opreme ne prenese resnejših napadov. Popolnoma pa zadostuje za preprečevanje kopiranja uporabnikom, ki niso vešči v sistemskem programiranju.

Postavlja se tudi vprašanje smiselnosti kopiranja programov. Pri nedovoljeni kopiji je pravilno delovanje programske opreme vprašljivo. Hkrati nam odpade tudi vzdrževanje s strani prodajalca. To sta dva dejavnika, ki govorita v prid nakupu programske opreme, saj se resnejše delo ne more izvajati na programski opremi za katero ne odgovarja avtor.

Na Zahodu so strogi zakoni, ki ščitijo avtorje pred krajo programske opreme, vendar je kraj le preveč, da bi lahko zaupali samo zakonu.

Na nivoju osebnih računalnikov ni zanesljive zaščite programske opreme. Zaščita je odvisna le od iznajdljivosti avtorjev na eni strani in 'lopovov' na drugi strani. Seveda pa to ne pomeni, da lahko vsakdo v kratkem času prebije zaščito v programski opremi.