Шеста национална конференција со меѓународно учество ЕТАИ'2003
Sixth National Conference With International Participation ETAI'2003
Охрид, Република МАКЕДОНИЈА – Ohrid, Republic of MACEDONIA
17–20. IX 2003

I2–5

ЗАШТИТА НА СОФТВЕР СО ПРОЦЕДУРА ЗА АКТИВАЦИЈА

Игор Крстев, Елена Ановска, Ирена Спасовска, Дејан Ѓорѓевиќ

Електротехнички факултет – Скопје, Карпош II, б. б. П. Фах 574, 1000 Скопје,
krstev@eudoramail. com

***Извадок* – Со развојот на компјутерската техника и развојот на хардверот кој е сé помоќен, помал и поевтин, се развива софтверот кој е сé помоќен, поголем, и поскап. Вредноста на софтверот како и можноста за неговата едноставна репликација го прави мета за негова нелегална употреба. Нелегалната употреба и дистрибуција на софтверот условува развој на системи за негова заштита. Во трудот е изложен механизам за заштита на софтвер базиран на активација со клуч карактеристичен за хардверот на кој е инсталирана заштитената апликација. Истиот е реализиран како библиотека на рутини кои може да се вградат во произволна апликација која треба да се заштити.**

***Клучни зборови* – регистрација на софтвер, пиратство, хардвер.**

1. ВОВЕД

Софтверот по дефиниција претставува множество на програми кои го управуваат хардверот во извршувањето на бараните функции. Софтверот е длабоко инкорпориран во денешното живеење и е една од највредните технологии кој доаѓа заедно со новата информатичка ера, притоа наметнувајќи се сам по себе првенствено поради својата неопходност. Тој претставува главен двигател на сé она кое во себе имплементира хардвер од кој и да е вид, од телекомуникациските и воените сателити до обичен кујнски тостер. Меѓутоа еден од најголемите проблеми кои се наметнуваат при употребата на софтверот, а кои доаѓаат пред сé поради неговата вредност и неговата едноставност за репликација, е секако неговата нелегална дистрибуција.

Нелегална дистрибуција на софтвер или софтверско пиратство, по дефиниција, претставува неавторизирано дуплицирање на софтвер со кое се прекршуваат сопственичките права, со цел лично или комерцијално користење на софтверот. Софтверското пиратство може да се изврши во повеќе категории:

Неовластено користење на софтвер од страна на краен корисник

- Користење на копија со една лиценца, меѓутоа нејзино инсталирање на повеќе компјутери;
- Копирање на медиумот кој го содржи софтверот и негова дистрибуција;
- Комерцијално користење на софтвер чија употреба е дозволена само во некомерцијални цели;

Клиент - сервер користење на софтвер

- Пиратство кое настанува тогаш кога бројот на корисници кој го користат софтверот преку мрежа е поголем од бројот на лиценци (дозволи) кој се купени за истиот тој софтвер

Интернет пиратство

- Пиратски Web сајтови кои овозможуваат слободно превземање на софтвер или негова размена

Дистрибуција на софтвер преку хард диск

- Случај кога дистрибутер на компјутери инсталира софтвер на компјутерите

кои ги продава, без лиценца за инсталираниот софтвер

Зошто е потребно да се заштити софтверот? Развивањето на софтверски апликации со себе повлекува големо вложување на време, материјални средства и интелектуален напор. Софтверското пиратство овде се наметнува како фактор кој ги оштетува не само оние кои развиваат софтвер, иако нив најмногу ги засегнува, туку и крајните корисници. Пиратството придонесува да поединците и фирмите кои развиваат софтвер се приморани да во своите апликативни решенија имплементираат механизми кои во голема мерка ќе спречат да нивната интелектуална сопственост биде користена на начин кој не им оди во прилог. Овие механизми придонесуваат да софтверот биде покомплициран и бараат вложување на дополнителни средства и напор што од своја страна повлекува зголемување на цената на софтверското решение, зголемување кое го го плаќа, а со тоа и оштетува крајниот корисник. Се ова повлекува дека заштитата на софтверот се наметнува сама по себе и претставува неминовност која мора да биде имплементирана во крајните софтверски решенија.

2. ЗАШТИТА НА СОФТВЕР

Терминот софтверска заштита се користи како синоним кој ги претставува сите методи и механизми кои му стојат на располагање на оној кој развива софтвер, а кои ги користи со цел да во голема мерка се осигура дека корисниците ќе користат копии од програми кои се легално дистрибуирани и купени. Како показател на квалитетот на софтверската заштита не е само тежината на вграденото решение кое ја реализира заштитата, иако е најпресудно, туку како фактор кој придонесува за нејзиниот квалитет е и способноста на вградената заштита да одговори на идните софтверски комерцијални решенија. Доброто решение за софтверска заштита не само што ќе овозможи оние кои развиваат и пласираат софтвер да ги зголемат своите приходи, туку и ќе им даде можност да се насочат во правец на понатамошен развој и подобрување на својот производ.

Генерално, механизмите кои се изградени со цел да го заштитуваат софтверот може да се најдат во една од следниве категории:

- Легално – правни механизми
- Технолошки базирани механизми

Софтверската лиценца и авторските права се дел од заштитните механизми насочени против софтверското пиратство и кои припаѓаат на првата категорија на механизми за заштита. Меѓутоа моќта на овие средства е само во рамките на легалните институции, и тие со ништо нема да го спречат оној кој се работи со нелегална дистрибуција и користење на софтвер да го користи софтверот надвор од легалните рамки.

Втората категорија на механизми на заштита на софтвер доаѓа во две форми:

- Хардверска
- Софтверска

Овие механизми за заштита, за разлика од легално-правните, веќе имаат моќ да го спречат или во голема мера да го оневозможат нелегалното користење и дистрибуција на софтверот. Нивната ефикасност во својата намена зависи пред од квалитетот на имплементираните методи за заштита и од спремноста и знаењето на оној кој сака да ги пренебрегне.

Хардверските базирани изведби на заштита, се темелат главно на следниве решенија:

- EPROM – програмабилен бришлив ROM
- ASIC (Application Specific Integated Controller) - базирано решение

Овој пристап во заштитувањето на софтвер е еден од најмоќните механизми, но ја наметнува потребата со софтверот да се дистрибуира и хардверот за заштита. Затоа ваквата заштита обично се применува кај софтверот со многу голема вредност а ограничена дистрибуција.

Софтверската заштита од своја страна може да се подели на неколку групи. Во една група спаѓаат програмите чија заштита е изведена со заштита на медиумот на кој се испорачуваат од копирање и барање на задолжително присуство на медиумот додека се извршува програмата. На овој начин се обично заштитени компјутерските игри. Друга група на програми пак е заштитена на тој начин што по инсталацијата бара активација (со соодветен клуч) за да може да работи. Кај примитивната изведба на ваквата заштита клучот за активација не е врзан за хардверот на кој работи програмата, со што е можно со истиот клуч да се активираат повеќе инсталации,

или веќе активираната инсталација да се пренесе на друг компјутер. Кај одредена класа на програми кои се наменети за исклучива работа во мрежно или интернет опкружување, се појавува и нов вид на заштита со употреба на сервер за лиценци. Кај ваквиот пристап, во централниот сервер постои база на сите легално дистрибуирани копии на софтверот (со соодветна единствена лиценца/клуч за секоја од нив). Секоја инсталација при стартувањето на програмата или во некои предодредени временски интервали комуницира со серверот испраќајќи ја својата лиценца. Серверот ја проверува лиценцата и доколку е валидна враќа одговор дека може да продолжи со работа или во спротивно да престане со работа. Предноста на ваквото решение е што централизирано многу лесно може да се следи употребата на сите инсталации и по потреба да се изврши деактивирање на точно одредена инсталација. Ограничувањата што ги наметнува ваквиот пристап е потребата од одржување на специјален постојано активен сервер за лиценци како и постојана конекција на сите компјутери на кои работи програмата со него.

Во овој труд е претставен механизам за заштита на софтвер кој обединува некои аспекти од хардверската и софтверската заштита со клуч за активација. Во реализираниот пристап за идентификација на валидноста на активираната апликација се користат идентификатори на некои од хардверските компоненти стандардно присутни во денешните персонални компјутери.

3. СОФТВЕРСКА ЗАШТИТА СО КЛУЧ ЗА АКТИВАЦИЈА

Основната методологија врз која почива пристапот според кој се заснова нашиот труд е софтверската заштита, каде крајниот корисник ја добива соодветната апликација во ограничен мод на работа, зависно од употребената стратегија на испорачување на апликацијата:

- неактивна дистрибуција
- временски активна
- ограничено активна

Во сите случаи, апликацијата ќе работи во ограничени услови, согласно применетата стратегија сé до внесувањето на активациониот код. Заштитата е така реализирана да може да биде имплементирана во различни апликации и користена од независни програмери за заштита на апликациите кои ги развиваат. Предвидена е нејзина имплементација во различни околини за развој на софтвер и тоа преку дистрибуција на кодот кој ја реализира заштитата во стандарден и униформен облик на делење на код помеѓу апликации во форма на dll, ActiveX компонента или COM објект. Апликацијата ќе треба при своето извршување, т.е. во различни фази од извршување на нејзиниот код да ги тестира вредностите кои ги добива од имплементираните рутини за заштита и да продолжи да работи во соодветниот мод на работа.

Еден од главните параметри кај софтверската заштита, врз која се заснова овој труд, е генерирањето на единствен идентификатор кој единствено ќе го идентификува компјутерот на корисникот и подоцна врз основа на тој идентификатор ќе побарува соодветен активационен код кој ќе обезбеди целосно функционирање на апликацијата. Ова обезбедува извршувањето на апликацијата да зависи од два клучни елемента: идентификациониот број и активациониот број.

Хардверските компоненти сами по себе содржат идентификатори кои на единствен начин ги карактеризираат. Во насока на ова размислување, единствениот идентификатор се генерира со соодветна комбинација на карактеристичните броеви од хардверските компоненти: BIOS, HDD, CPU, Network Card кои се идентификувани на единствен начин.

Комплетното решение на досега изложениот проблем, е претставено во следниве фази

- Исчитување на сериските броеви на хардверските компоненти (BIOS, HDD, CPU)
- Генерирање на карактеристичен број (единствен идентификатор) врз основа на прочитаните сериски броеви
- Споредување на карактеристичниот број со активациониот број (доколку е внесен)
- Мод на работа на апликацијата, во согласност со резултатот од тестирањето на карактеристичниот со активациониот број

3.1. Исчитување на сериските броеви на хардверските компоненти

Реализацијата на овој дел од кодот, код кој пристапува до хардверските ресурси на ОС (оперативниот систем) се реализира со користење на рутини и структури на податоци кои ги обезбедува јадрото на оперативниот систем и се имплементирани во неговите системски датотеки. Овој код, неговото извршување и неговата реализација, зависат од платформата за која се реализира заштитата, бидејќи Windows оперативните системи во зависност дали се работи за ОС базиран на NT јадро (Windows NT, Windows 2000, Windows XP) или не (Windows 95, Windows98), на различен начин го третираат корисничкиот пристап до ресурсите со кои работи ОС. NT системите со цел да обезбедат поголема доверливост и голем степен на паралелизам (мултитаскинг) се многу поригорозни во однос на корисничкиот пристап до ресурсите, што резултира во покомплексен код за реализација.

Ова повлекува, да во зависност од ОС се извршува различен код за една иста намена, меѓутоа ова се разбира е невидливо од аспект на оној кој ја интегрира заштитата во своите апликации.

Хардверските компоненти од кои се исчитуваат карактеристични и единствени параметри треба да го задоволат условот да бидат релативно неменливи за време на просечниот век на употреба на стандардна компјутерска конфигурација. Во реализацијата на нашето решение се користат: BIOS-от, хард дискот и процесорот.

Параметар кој се исчитува од BIOS-от и кој како вредност го прави единствен е неговиот сериски број одреден од производителот. Оваа вредност се исчитува од меморискиот простор во кој е сместен BIOS-от, во процесот на стартување на компјутерот, на однапред предефинирана, од производителот, мемориска локација. Оваа вредност се исчитува на два начина, во зависност од оперативниот систем на кој се извршува кодот на заштитата: директно (кај оперативни системи кои не се од класата на NT оперативните системи) или индиректно (при што прво се врши пресликување на меморискиот простор на процесот од оперативниот систем во чиј адресен простор е сместена содржината на БИОС-от, и потоа индиректно се исчитува саканата вредност).

Карактеристиките на процесорот кои се од интерес и кои се исчитуваат се неговата работна фреквенција, моделот и производителот.

Параметар кој е единствен за хард дискот е, исто како и за БИОС-от, неговиот единствен сериски број, доделен од производителот. Во овој случај, до серискиот број на овој уред се доаѓа со директно исчитување на оваа вредност од самиот уред, користејќи ги рутините кои ги обезбедува јадрото на оперативниот систем.

3.2. Генерирање на единствен идентификатор

Откако ќе се прочитаат броевите од хардверските компоненти врз основа на нив и со нивна комбинација, се генерира единствен идентификатор (карактеристичен број) за соодветниот кориснички компјутер. Оваа вредност сама по себе е единствена бидејќи е генерирана од единствени вредности, и на единствен начин го идентификува компјутерот на корисникот што е една од темелните поставки врз која стои идејата за софтверска заштита.

3.3. Споредување на карактеристичниот број со активациониот број

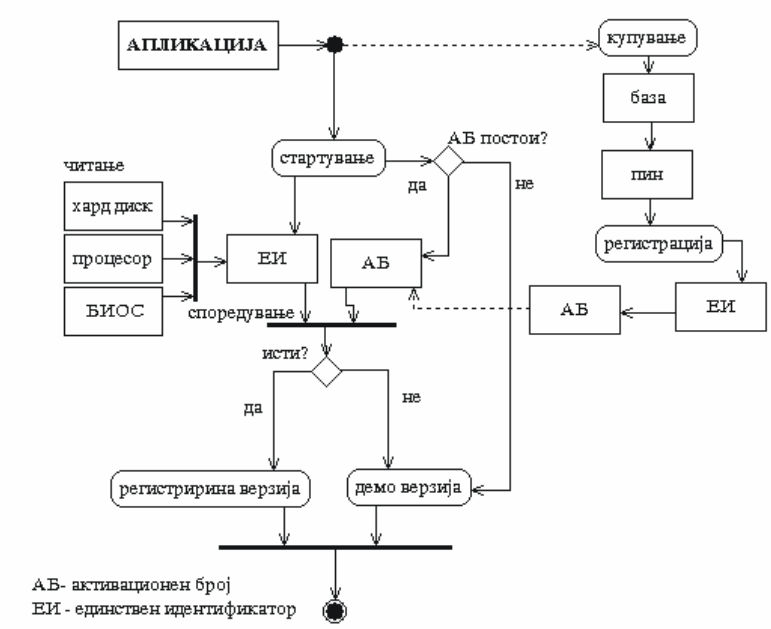
При стартувањето на апликацијата, најпрвин се иницира постапката за генерирање на единствениот идентификатор и споредување на оваа вредност со друга вредност (активационен број) кој е показател дека соодветната апликација е легално купена, бидејќи оваа вредност, овој број, крајниот корисник го добива по регистрирањето на својата верзија на апликација, од дистрибутерот. Активациониот број е вредност која на единствен начин кореспондира со единствениот идентификатор, а со тоа и со компјутерот на корисникот, односно овој број се генерира со пресликување на единствениот идентификатор во друга форма. Активациониот број го добиваме од единствениот идентификатор со негово криптирање, со што секогаш е возможно да од едниот број на единствен начин се добие другиот. Со ова, проверката дали еден број е активационен ја вршиме со декриптирање на вредноста која се проверува и нејзино споредување со единствениот идентификатор.

3.4. Различен мод на работа на апликацијата

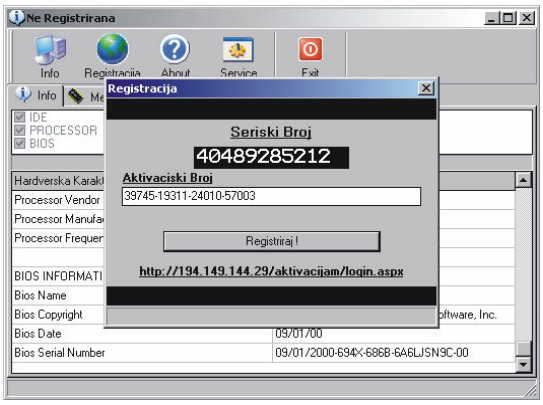
Врз база на погоре изложениот механизам, апликацијата која во себе имплементира заштита, секогаш има информација дали се извршува кај корисник кој има нејзина легална копија. Активациониот број е "клучот" со кој се "отклучува" апликацијата. Како ќе се извршува апликацијата понатаму, врз основа на она кое го добива како повратна информација од кодот за заштита, е одлука на оној кој ја развива самата апликација. Доколку тестирањето на единствениот идентификатор и активациониот број е во ред, апликацијата ќе може да се извршува во својата целосна форма, во спротивно таа може да продолжи да работи во модот на работа во кој била испорачана, или пак воопшто да не работи. Сето тоа е одлука на програмерот кој ја развива својата апликација, и одлука која се потпира врз кодот за заштита.

4. РЕГИСТРАЦИЈА

Активациониот број, крајниот корисник го добива од дистрибутерот од кого ја набавува апликацијата. Постапката за активација на софтверот, заедно со приказот на генерирање на единствен идентификатор, шематски е прикажана на слика 1. По испорачувањето на апликацијата, за секој краен корисник се генерира единствен број (ПИН) кој се сместува во базата на серверот за активација. По инсталацијата на вака дистрибуираниот софтвер истиот е нефункционален или е само делумно функционален до неговата активација.
При процесот на активација, слика 2, програмата ги чита карактеристичните параметри на хардверот на кој е инсталирана и со ПИН-от кој го внесува корисникот се генерира регистрационен код кој е единствен за дадениот систем.



Сл. 1. Процес на функционирање на софтверска заштита



Сл. 2. Процес на активација на заштитен софтвер

Регистациониот код се праќа на серверот за активација каде од него се екстрахира ПИН-от и во базата се проверува дали истиот е валиден. Во потврден случај се генерира единствен активационен код кој се испраќа назад до корисникот и се зачувува на корисниковиот компјутер, со што програмата е активирана. При секое извршување на програмата истата прво проверува дали е активирана и дали се извршува на соодветниот хардвер, на тој начин што проверува дали е присутен активациски код и дали тој соодветствува на регистрациониот код генериран од карактеристич-

ните параметри на хардверот и ПИН-от. На слика 2 е даден приказ на апликација која во себе имплементира заштита, односно дијалогот со кој корисникот го внесува активациониот број, добиен од серверот за активација, откако претходно го внел карактеристичниот (на сликата "сериски број")

5. ЗАКЛУЧОК

Не постои совршена и апсолутно сигурна заштита на софтвер, барем не досега. Сепак ова не го руши концептот на заштита на софтвер и негова имплементација во апликациите.

Презентираниот систем за заштита на софтвер, овозможува апликациите кои ќе го имплементираат да се дистрибуираат на незаштитени медиуми или дури и преку интернет. При регистрацијата на апликацијата валидниот ПИН претставува доказ за нејзино легално поседување. Регистрацијата може да се изведе по интернет (и единствено тогаш е потребна конекција кон серверот за активација) или преку телефон (со оператор или говорен автомат). По успешно изведената активација апликацијата може неограничено да работи сѐ додека се извршува на истиот компјутер на која била активирана.

Од степенот на интегрираната заштита, колку заштитата која е имплементирана во апликацијата е покомплексна и како е имплементирана во кодот на апликацијата, зависи колку напор и време ќе вложи некој за да најде решение како да ја пренебрегне. Што значи, колку повеќе интелектуален и интелигентен напор е вложен во код кој има за задача да контролира друг код, толку потешко ќе биде тој труд и напор да бидат совладани. Ова повлекува дека сепак не е сеедно дали апликацијата ќе интегрира во себе заштита или не, далеку е од сеедно. Примерно: во случај кога се работи за апликации кои не се за масовна употреба, што значи дека се конкретизирани за тесен и специфичен пазар, што наметнува и нивна висока цена, интегрирањето на соодветна заштита, заштита со доволна комплексност, во секој случај ќе го наметне прашањето дали е воопшто исплатливо да се вложува време и средства за нејзино пробивање. Заштитата е и индикатор на легално или нелегално користење на софтверот. Сите оние кои нелегално користат даден софтвер, а посебно кога се работи за комплексен софтвер, нема да можат да ги користат поволностите кои евентуално ги нуди софтверската куќа: техничка поддршка или пак надградба на софтверот, што е мошне корисно во голем број на случаи, бидејќи во спротивно корисникот и нема да има голема полза од својата копија на апликацијата.

Сепак едно вакво резонирање ја оправдува исплатливоста и нужноста за имплементација на код во апликациите кој ќе ги заштитува од нивна нелегална редистрибуција.

6. ЛИТЕРАТУРА

[1] Marco Cantù: Mastering™ Delphi™ 6.

[2] John Ayres, David Bowden, Larry Diehl, Phil Dorcas, Kenneth Harrison, Rod Mathes, Ovais Reza, Mike Tobin: The Tomes of Delphi 3: Win32 Core API.

[3] Richard Simon: Windows NT Win32 API SuperBible.

Summary

SOFTWARE PROTECTION USING ACTIVATION PROCEDURE

Igor Krstev, Elena Anovska, Irena Spasovska, Dejan Gorgevik

Faculty of Electrical Engineering, Skopje, Karpos II, b. b. P. O. Box 574, 1000 Skopje, krstev@eudoramail. com

Considering the rapid progress in computer technology we are witnessing the fact that the hardware is becoming better, smaller and cheaper, as the software is becoming better, bigger and more expensive. Unfortunately, because software is so valuable, and because it is very easy to create an exact copy of a program in seconds, software piracy is widespread. The illegal usage and distribution of software has created the need for software protection. This paper presents a mechanism for software protection based on hardware dependent activation key. A library of applicable procedures was developed that can be easily incorporated in any application that needs protection.

Key words – software, protect, piracy