Nenad Crnko
IGEA d.o.o Varaždin
Varaždin

UDK: 681.3
Prethodno priopćenje

Novi pristup zaštiti računala pred napadom parazitskih virusa

U radu je prikazan novi pristup zaštiti računala pred napadom parazitskog oblika virusa. Umjesto korištenja posebnih antivirusnih programa, zaštita se ugrađuje u izvršnu verziju aplikacije kod njezinog razvoja, i automatski se aktivira kod svakog korištenja aplikacije. Na taj način postiže se visok stupanj zaštite koji je za korisnika potpuno transparentan. Konkretnim primjerom prikazana je implementacija zaštite u programskom jeziku C, kao i funkcioniranje zaštite u slučaju napada virusa na MS DOS računalima.

Ključne riječi: računarski virusi, antivirusni programi, zaštita podataka i programa, MS DOS, C programski jezik.

1 Uvod

Jedan od najčešćih uzroka gubitka podataka i programa na računalima jesu računarski virusi. Zbog posljedica, koje oni mogu prouzročiti, postaju sve važnije različite metode otkrivanja i uklanjanja virusa s računala. Pri tome se najčešće koriste posebni antivirusni programi, koji uz brojne dobre osobine, imaju i jedan veliki nedostatak, korisnik računala osim poznavanja aplikacija, koje koristi u svakodnevnom radu, mora upoznati i funkcioniranje dodatnih programa koji štite računalo od napada virusa. U radu su opisane osnovne karakteristike računarskih virusa, kao i najčešće korištene metode za njihovo otkrivanje i uklanjanje. Izložena je i primjena nove metode zaštite, koja se može ugraditi u svaku aplikaciju prilikom njezinog razvoja, čime postaje transparentna za upotrebu. Time se uklanja najvažnija negativna osobina svakog antivirusnog programa, potreba za dodatnim proučavanjem njegove upotrebe. Metoda je demonstrirana na konkretnim primjerima u programskom jeziku C, a za provjeru njezine djelotvornosti korišteno je nekoliko najraširenijih računarskih virusa. Čitav rad je ograničen na računala koja rade pod MS DOS operativnim sustavom, jer se radi o danas najraširenijoj kategoriji računala, koja je istovremeno i najčešće meta napada virusa.

2 Osnovne karakteristike računarskih virusa

Računarski virus je program posebne namjene, koji u većini slučajeva ima za cilj uništenje podataka i programa na zaraženom računalu. Osim ovakvih destruktivnih osobina, postoje i bezopasni virusi čije se funkcioniranje svodi samo na pokazivanje svoje prisutnosti na računalu, bez izazivanja dodatnih negativnih efekata. Slično biološkim virusima, računarski virusi imaju mogućnost širenja s računala na računalo. Pri tome se koriste magnetni mediji za čuvanje podataka (npr. disketa zaražena na jednom računalu može uzrokovati zarazu na drugom računalu na kojem se koristi), ili mreže u koje su spojena zaražena i nezaražena računala. Osnovne kategorije virusa:

1. parazitski virusi
2. virusi boot sektora
3. multi-partite virusi
4. companion virusi
5. link virusi

Parazitski virusi predstavljaju najrašireniji oblik računarskih virusa. Mijenjaju sadržaj izvršne datoteke koju zaraze, tako da se dodaju na njezin početak ili kraj, najčešće ne mijenjajući pri tome funkcionalnost programa. Kada se pokrene ovakav zaraženi program, prvo se izvodi programski kod virusa, a tek nakon toga ostatak programa. Virus pri tome nastoji osigurati svoje daljnje širenje na nezaražene datoteke, a u određenom trenutku (npr. točno određenog dana u mjesecu) dolazi do izražaja njegov destruktivni dio koji uništava podatke. Nova metoda zaštite pred virusima, opisana u ovom radu, ograničena je na ovu kategoriju virusa.

Virusi boot sektora - aktiviraju se kod uključivanja računala. Prilikom aktiviranja operativnog sustava računala dio operativnog sustava učitava se s diska ili diskete. Na zaraženom računalu virus je izmijenio glavni sektor diska u kojem se nalazi rutina za pokretanje operativnog sustava (engl. boot sektor), tako da se aktivira prije operativnog sustava. Oblici djelovanja na računalo, kao i kod ostalih oblika virusa, ograničeni su samo maštom autora.

Multi-partite virusi - istovremeno pokazuju karakteristike i virusa boot sektora, kao i parazitskih virusa. To znači da se aktiviraju kod uključenja računala zbog izmjene boot sektora diska, ali istovremeno napadaju i izvršne datoteke.

Companion virusi - kreiraju alternativnu izvršnu datoteku koja se izvodi prije originalne izvršne datoteke. U MS DOS operativnom sustavu postoje dva oblika izvršnih datoteka: .COM i .EXE nastavci u imenima datoteka. Današnje komercijalne aplikacije najčešće su u .EXE obliku, a virus kreira .COM datoteku jednakog imena koja se uvijek izvodi prije odgovarajuće .EXE datoteke. Na taj način osigurano je aktiviranje virusa, kod svakog pokretanja aplikacije. Ovaj oblik virusa najlakše se otkriva pa sve više gubi na značenju.

Link virusi - preusmjeravaju pokazivače koje vodi operativni sustav o sektorima na kojima je spremljena izvršna datoteka. Nakon ove izmjene prvi dio svakog izvršnog programa pokazuje na dio diska na kojem se nalazi virus, čime se omogućuje njegovo aktiviranje.

3 Metode za otkrivanje i uklanjanje virusa

Za otkrivanje i uklanjanje računarskih virusa postoji više različitih metoda, koje se međusobno razlikuju po brzini izvodjenja, uspješnosti i načinu implementacije. Antivirusni programi koji postoje na tržištu koriste jednu ili istovremeno više različitih metoda za uklanjanje virusa. Vrste metoda:

- usmjerena dijagnostika s pregledom diskova
- usmjerena dijagnostika sa sondiranjem
- diferencijalna dijagnostika
- rezidentna dijagnostika
- heuristička dijagnostika

Usmjerena dijagnostika s pregledom diskova predstavlja najjednostavniji oblik pregleda računala. Prvo je potrebno svaki od postojećih virusa analizirati i u njemu pronaći niz bajtova koji su karakteristični samo za taj virus. Nakon toga se u antivirusni paket koji koristi ovu metodu, ugrađuje ovaj niz bajtova, nakon čega je moguće na računalu pronaći analizirani virus. Pri tome je moguće pregledavati i RAM memoriju računala (za viruse koji nakon aktiviranja ostaju rezidentni), kao i boot sektor, odnosno izvršne datoteke programa (po potrebi i sve ostale). Navedena metoda ima tri osnovna nedostatka:

- sporost - porastom broja virusa potrebno je koristiti sve veći broj uzoraka koji se pokušavaju pronaći na računalu, što automatski smanjuje brzinu pretraživanja,
- lažni alarmi - postoji mogućnost da i nezaražene datoteke sadrže isti niz bajtova kao i neki od virusa. U tom slučaju korisnik je obaviješten da je njegovo računalo zaraženo virusom, iako zaraza u stvarnosti ne postoji,
- nemogućnost otkrivanja novih virusa - navedena metoda je potpuno nemoćna da otkrije novi virus, jer je vrlo mala vjerojatnost da će sadržavati isti niz karakterističnih bajtova, kao i neki od prije analiziranih virusa. Čak kada bi se ovo i dogodilo, korisnik bi dobio nepravilnu obavijest o zarazi. Unatoč navedenim nedostacima, ova metoda se najčešće koristi u antivirusnim paketima na tržištu.

Usmjerena dijagnostika sa sondiranjem zahtijeva točno poznavanje načina djelovanja virusa. U tom slučaju izbjegava se pretraživanje ogromnih uzoraka bajtova za karakterističnim nizom, jer je moguće odrediti točno mjesto u datoteci (boot sektoru, RAM memoriji) koje je potrebno pregledati da bi se detektirala zaraženost.

Diferencijalna dijagnostika od svih navedenih metoda osigurava najveći stupanj otrivanja zaraze. Metoda se sastoji u tome da se za svaku datoteku (ili boot sektor) određenim algoritmom dobije njezina karakteristična osobina (npr. broj bajtova od kojih se sastoji datoteka, zbroj svih bajtova datoteke itd.). Svaki virus koji pokušava zaraziti računalo mora izmijeniti ove karakteristike (npr. povećati dužinu datoteke). Uspoređivanjem ranije spremljene vrijednosti s naknadno proračunatom moguće je otkriti promjene na datoteci, a samim tim i mogućnost zaraze. Najveća prednost metode je mogućnost otkrivanja i potpuno nepoznatih i novih virusa, a osnovni nedostatak u tome što korisnik nije obaviješten o tome kojom vrstom virusa je računalo zaraženo, nego samo da je došlo do zaraze.

Rezidentna dijagnostika omogućuje prevenciju pred zarazom. Posebni antivirusni program se pokreće na početku rada s računalom, nakon čega se u RAM memoriju premješta dio koji tokom čitavog rada kontrolira pristup ključnim dijelovima računala (npr. boot sektoru), korištenjem procedura operativnog sustava. U slučaju da virus pokuša pristupiti nekom od tih dijelova, obavještava se korisnik i sprečava akcija virusa. Osnovna prednost metode je da može spriječiti pojavu i širenje zaraze na računalu. Istovremeno postoje i dva bitna nedostatka. Sprečava se i svaki regularan pristup ključnim resursima (npr. kod namjernog formatiranja diska), što se najčešće rješava tako da korisnik potvrdi ili odbaci traženu akciju. Drugi nedostatak je u tome što noviji virusi pristupaju ključnim resursima zaobilazeći operativni sustav računala (direktna komunikacija s hardverskim komponentama), čime se zaobilazi i ova metoda.

Heuristička dijagnostika predstavlja najnoviju metodu za otkrivanje virusa. Antivirusni paket koji koristi ovu metodu analizira izvršni kod svake datoteke tražeći pri tome "opasne" dijelove: pristup boot sektoru, operacije niskog nivoa s diskom, pokušaj pristupa .EXE datotekama i druge operacije karakteristične za viruse. Prednost metode je mogućnost otkrivanja "opasnosti" i u novim virusima, a nedostatak prijavljivanje lažnih zaraza. Komercijalni paketi također mogu sadržavati slične rutine (npr. različiti uslužni programi, programi za analiziranje perfomansi računala itd.)

Nakon otkrivanja zaraze potrebno je ukloniti virus s računala. Tu je također moguće koristiti različite metode. Najpouzdaniji način je ponovno formatiranje disketa i diska (niskog nivoa) te naknadna instalacija svih programskih paketa i restauracija datoteka s podacima. Nažalost ovo je i najmukotrpniji način koji od korisnika zahtijeva najviše uloženog vremena te postojanje rezervnih kopija svih potrebnih datoteka. Alternativno, mnogi antivirusni paketi sadrže mogućnost uklanjanja virusa bez uništavanja sadržaja diska. Metoda se svodi na prepoznavanje načina na koji virus mijenja objekte na računalu, te pokušaja restauracije originalnog oblika objekta. Pri tome se najbolji rezultati postižu kod metoda koje se temelje na diferencijalnoj dijagnostici i spremaju originalne informacije o objektima kod prvog pregleda. Kasnije je iz ovih informacija olakšano restauriranje originalnog sadržaja. U nekim slučajevima obnavljanje originalnog sadržaja nije moguće, pa se kao rezultat dobije aplikacija koja ne funkcionira korektno. U tom slučaju potrebno je ipak reinstalirati aplikaciju s originalnih disketa. Svaka od dosad navedenih metoda za detektiranje i uklanjenje virusa, osim već ranije opisanih, ima i dodatnu negativnu osobinu. Korištenje aktivirusnog paketa, u koji je ugrađena, zahtijeva od korisnika svladavanje čitavog niza novih termina, kao i samog antivirusnog paketa. Na primjer, književnik želi na svojem računalu koristiti program za obradu teksta, za pisanje svojih budućih djela. Prijatelji su ga obavijestili da na MS DOS računalima postoje virusi koji u trenutku mogu uništiti višemjesečne napore. Da bi se osigurao pred katastrofom, književnik je odlučio da nabavi neki od antivirusnih paketa. Da bi ga uspješno koristio i zaista spriječio zarazu, mora upoznati način na koji se paket instalira na računalo, kako se pokreće, u kojim slučajevima može doći do lažnih alarma, što uraditi kada program javi zarazu itd. Nakon dugo, dugo vremena, može zaista početi i s pisanjem samog teksta.

4 Novi pristup zaštiti

Da bi antivirusna zaštita bila zaista djelotvorna, ona treba biti za korisnika transparentna. Korisnik treba koristiti svoju aplikaciju ne razmišljajući o virusima, ili antivirusnoj zaštiti, ali ako do zaraze slučajno dođe, o tome odmah treba biti obaviješten, a

zaraza po mogućnosti uklonjena. U nastavku je objašnjeno kako je ovu transparentnost moguće postići za parazitske viruse.

Novi pristup zaštiti svodi se na ugrađivanje posebnih rutina u aplikaciju, koje automatski detektiraju svaku promjenu na aplikaciji, što je najvjerojatnije posjedica virusne zaraze. Po tome je metoda slična diferencijalnoj dijagnostici, ali ima i jednu bitnu prednost. Korisnik ne mora koristiti posebni antivirusni program da bi provjerio svoje računalo, nego o tome brine proizvođač aplikacije. Time zaštita za korisnika postaje transparentna.

Proizvođač aplikacije, koji želi koristiti ovu metodu, mora uraditi slijedeće:

1) razviti aplikaciju na standardni način korištenjem nekog od programskih alata, uz dodatak rutine koja čita podatke iz datoteke opisane u slijedećoj točki i uspoređuje ih s trenutnim,

2) razviti posebni alat koji posebnim algoritmom pronalazi podatke karakteristične za aplikaciju (jednaki postupak kao i kod diferencijalne dijagnostike) i upisuje ih u posebnu datoteku koja se isporučuje uz aplikaciju.

Nakon instaliranja ovakve aplikacije na korisnikovo računalo, kod svakog korištenja prvo se pronalaze karakteristični podaci za aplikaciju i uspoređuju s originalnim zapisanim u datoteci koju isporučuje proizvođač aplikacije (točka 2). Ako dolazi do odstupanja, o tome se obavještava korisnik. Glavna prednost metode je potpuna transparentnost za korisnika, koji se virusima mora baviti tek kada se zaista pojave. Također, unapređivanjem tehnike (istraživanja su u toku), moguće je u popratnu datoteku upisati i dodatne informacije, koje bi trebale omogućiti automatsko uklanjanje virusa s računala. Osnovni nedostatak metode je u tome što nije moguće detektirati vrstu virusa koja je zarazila računalo, ali kako se teži za što većom jednostavnošću i automatizmom u upotrebi, ovo za korisnika nije bitno. Drugi nedostatak je u određenom usporenju pokretanja aplikacije (prije izvođenja glavnog dijela aplikacije potrebno je provjeriti njezinu autentičnost), ali razvoj sve bržih hardverskih komponenti (mikroprocesora, memorije i diskova) kao i programske podrške na nivou operativnog sustava (programi za ubrzivanje rada diska), uklanja ovu negativnost.

5 Programski primjer

Da bi metoda bila što bolje objašnjena, u nastavku je naveden i programski primjer njezine implementacije. Primjeri su pisani u programskom jeziku C (Borlandova verzija Turbo C), iako je metodu moguće implementirati na svakom jeziku koji ima pristup datotekama na niskom nivou (umjesto čitanja logičkih podataka iz datoteke, potrebno

je čitati nizove bajtova). Također, postoji i jedna razlika u odnosu na prije navedeni opis metode. Umjesto upisivanja karakterističnih podataka o aplikaciji u posebnu datoteku, ove informacije se dodaju na kraj same .EXE datoteke, što je dozvoljeno u MS DOS operativnom sustavu.

Programski primjer sastoji se od dva dijela: PROTECT.C i PROGRAM.C . Prvi program pretražuje datoteku PROGRAM.EXE, nastalu prevođenjem glavne aplikacije (PROGRAM.C) i dodaje ključne informacije na kraj datoteke. U ovom slučaju radi se samo o jednom bajtu u kojem su spremljene djelomične informacije o dužini datoteke i zbroju svih bajtova u datoteci. Radi se o primitivnoj zaštiti koja je ugrađena da bi se što više skratio i pojednostavio programski primjer, ali je na isti način moguće ugraditi i rutinu koja, na primjer, izvodi kriptografski kodiranu provjeru autentičnosti datoteke.

Kod pokretanja glavne aplikacije izvodi se jednaka rutina koja dobivene podatke uspoređuje s dodanim bajtom na kraju datoteke. Ako vrijednosti nisu jednake, došlo je promjene u aplikaciji, što signalizira mogućnost virusne zaraze.

Ovako zaštićena datoteka provjerena je u praksi s nekoliko parazitskih virusa: CASCADE, YANKEE, JERUSALEM. Svakim od nabrojenih virusa namjerno je zaraženo računalo za testiranje, nakon čega je na njega instaliran PROGRAM.EXE. Rezultati su bili slijedeći:

1) virus CASCADE - napada samo na izvršne datoteke tipa .COM što znači da nije imao utjecaja na datoteku PROGRAM.EXE pa ni metoda za otkrivanje zaraze nije pronašla virusnu zarazu,

2) virusi YANKEE i JERUSALEM su zarazili datoteku PROGRAM.EXE mijenjajući joj dužinu, a samim tim i zbroj vrijednosti bajtova. U oba slučaja ugrađena provjera otkrila je virusnu zarazu i ispisala potrebnu obavijest.

```
/* PROTECT.C - inicijalizacija zastite pred virusima */
# include < stdio.h >
# include < math.h >
main()
{
        FILE datoteka;
        long duzina=0, zbroj=0;
        long pom1, pom2;
        unsigned char znak,znak1,znak2,znak3;

        if ((datoteka = fopen("PROGRAM.EXE","rb")) = = NULL) {
            printf("Greska u otvaranju datoteke ! ");
            return 1;
{
while (!(feof(datoteka))) {
            znak = fgetc(datoteka);
            duzina++;
            zbroj += znak;
}
fclose(datoteka); znak1 = duzina - floor(duzina /10) * 10;
znak2 = zbroj - floor(zbroj /10) * 10;
znak3 = 10 * znak1 + znak2;

if ((datoteka = fopen("PROGRAM.EXE","a")) = = NULL) {
    printf("Greska u otvaranju datoteke! ");
    return 1;
}
fwrite(&znak3,1,1,datoteka);
fclose(datoteka);
return 0;
}
```

```
/* PROGRAM.C - Glavna aplikacija */
# include < stdio.h >
# include < math.h >

main()
{
        FILE *datoteka;
        long duzina=0, zbroj=0;
        long pom1, pom2, i;
        unsigned char znak,znak1,znak2,znak3,zpom;

        printf("Ovo je demo program za autodetekciju zaraze ! \ n");
        if ((datoteka = fopen("PROGRAM.EXE","rb")) == NULL) {
            printf("Greska u otvaranju datoteke ! ");
            return 1;
while ( ! (feof(datoteka))) {
        zpom = znak;
        znak = fgetc(datoteka);
        duzina ++;
        zbroj + = znak;
}
duzina--;
zbroj -= zpom;
fclose(datoteka);
znak1 = duzina - floor(duzina /10) * 10;
znak2 = zbroj - floor(zbroj /10) * 10;
znak3 = 10 * znak1 + znak2;
if(znak3 ! = zpom)
        printf("Kod je izmijenjen, mogucnost zaraze! \ n  \ n ");
return 0;
}
```

6 Zaključak

Ugradnjom metode za otkrivanje virusne zaraze u aplikaciju postiže se visok stupanj zaštite, koji je za korisnika potpuno transparentan. Briga oko virusa se prenosi na proizvođača aplikacije, ali kako je opisana metoda vrlo jednostavna za implementaciju, ne bi trebala u većoj mjeri utjecati na vrijeme potrebno za razvoj novih produkata, kao ni na povećanje njihove cijene.

Metoda, za sada, omogućava samo otrivanje virusne zaraze, a ne i njezino uklanjanje. Također, ograničena je samo na parazitski oblik virusa, dok prisutnost ostalih oblika na računalu ne registrira. To je posjedica početne faze u istaživanju, ali je metoda vrlo pogodna za daljnja proširenja i unapređivanja.

Programski primjeri su kratki i jednostavni, ali zato i nešto manje pouzdani. Međutim, napisani su tako da ih je vrlo jednostavno moguće unaprijediti (npr. složenijim algoritmima za pronalaženje ključnih informacija) i prevesti na druge programske jezike.

Literatura

[Goretsky A.] , "VIRUSCAN Version 9.17V106", McAfee Associates Inc., Santa Clara, U.S.A, 1993.

[Hruška J.] , "Virusi - problem koji se razmnožava", Byte (hrvatsko izdanje) 1/93, str. 50 - 58.

[Korenjak B.] , Erjavec T., "Koji najjači u zemlji je toj", Moj mikro 1/91, str. 58 - 63.

[.....] , "Thunderbyte virus detector", Thunderbyte B.V., 1993.

[.....] , "Turbo $C++-$ Library Reference", Borland International, 1990.

Primljeno: 1993-06-10

Crnko N. Ein neuer Zugang zum Schutz von Rechnern vor dem Angriff von parazitischen Viren

ZUSAMMENFASSUNG

In der vorliegenden Arbeit wird ein neuer Zugang zum Schutz von Computern vorm Angriff der parazitischen Form von Viren bescrieben. Anstatt der Benutzung von spezielen Antivirenprogrammen wird der Schutz in die ausführende Version der Aplikation, während der Programentwicklung, eingebunden und wird automatisch bei jeder Benutzung der Aplikation aktiviert. Mit dieser Methode wird ein hohes Grand an Sicherheit ereicht, dar für den Benutzer vollkommen transparent ist.

(Prijevod: Jakupek Miodrag)