Metode detekcije zlonamjernog softvera pametnih telefona

Rohlik, Andrej

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Andrej Rohlik

METODE DETEKCIJE ZLONAMJERNOG SOFTVERA PAMETNIH TELEFONA

ZAVRŠNI RAD

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH
ZNANOSTI ODBOR ZA
ZAVRŠNI RAD

Zagreb, 31. ožujka 2020.

Zavod: **Zavod za informacijsko komunikacijski promet**

Predmet: **Terminalni uređaji**

ZAVRŠNI ZADATAK br. 5781

Pristupnik: **Andrej Rohlik (0135251341)**

Studij: Promet

Smjer: Informacijsko-komunikacijski promet

Zadatak: **Metode detekcije zlonamjernog softvera pametnih telefona**

Opis zadatka:

Prikazati klasifikaciju zlonamjernog softvera pametnih telefona. Objasniti značajke detekcije zlonamjernog softvera pametnih telefona. Istražiti tehnike analize zlonamjernog softvera. Prikazati metode za detekciju zlonamjernog softvera pametnih telefona. Analizirati značajke i postupke reverznog inženjeringa.

Mentor:

dr. sc. Siniša Husnjak

Predsjednik povjerenstva
za završni ispit:

Sveučilište u Zagrebu

Fakultet prometnih znanosti

ZAVRŠNI RAD

METODE DETEKCIJE ZLONAMJERNOG SOFTVERA PAMETNIH TELEFONA

SMARTPHONE MALWARE DETECTION METHODS

Mentor: dr. sc. Siniša Husnjak

Student: Andrej Rohlik

JMBAG: 0135251341

Zagreb, rujan 2020.

SAŽETAK

Zlonamjerni ili zloćudni softver programski je specijalizirano napravljen kod koji su razvili napadači na sve vrste uređaja, a dizajniran je tako da uzrokuje veliku štetu podacima i sustavima ili da dobije neovlašteni pristup informacijsko-komunikacijskoj mreži. Rad se bavi tematikom zaštite od zlonamjernog softvera i njegovom detekcijom. Opisane su najčešće vrste zlonamjernih softvera i njihove karakteristike kako bi se omogućio uvid u aktivnosti pojedinog zlonamjernog programa koje se vrše na pametnom telefonu. U radu su opisani i neki od poznatijih antivirusnih programa pomoću kojih se obavlja detekcija zlonamjernih softvera. Kao odgovor na razne zlonamjerne softvere razvijene su raznolike tehnike i metode detekcije pomoću kojih se osigurava siguran rad uređaja. To uključuje statičke i dinamičke analize zlonamjernih programa kao i razne metode detekcije koje se temelje na potpisu, anomalijama, specifikacijama ili metodu detekcije u oblaku. Reverzni inženjering je također jedan od postupaka analize zlonamjernog softvera i pomaže u detektiranju i otklanjanju zlonamjernog programa.

KLJUČNE RIJEČI: zlonamjerni softver; pametni telefon; detekcija; reverzni inženjering

SUMMARY

Malicious software is a specialized software program developed by attackers on all types of devices and is designed to cause great damage to data and systems or to gain unauthorized access to the information and communication network. The paper deals with the topic of protection against malware and its detection. The most common types of malware and their characteristics are described in order to provide insight into the activities of a particular malware that are performed on a smartphone. The paper also describes some of the better known antivirus programs that are used to detect malware. In response to various malware, a variety of detection techniques and methods have been developed to ensure the safe operation of the device. That includes static and dynamic malware analysis as well as various detection methods that are signature-based, anomaly-based, specification-based or cloud-based. Reverse engineering is also one of the procedures of malware analysis and helps in detecting and eliminating malware.

KEY WORDS: malware; smartphone; detection; reverse engineering

Sadržaj

1. Uvod

Pametni telefoni su danas svuda oko nas i njihovo korištenje se svakim danom sve više povećava. Kada se uzme u obzir da su se prvi pametni telefoni pojavili prije nešto više od 10 godina, nevjerojatno je kako su se brzo ekspandirali na tržištu. Sa svakim novim izumom dolaze i neke prijetnje. Najčešće prijetnje pametnim telefonima su zlonamjerni softveri i još drugi razni domišljati načini napadača kako bi došli do osjetljivih informacija korisnika bez njegovog znanja. Svakim danom takvi napadi rastu, a kako nebi više dolazilo do takvih napada korisnici se trebaju educirati o mogućim napadima i kako se od njih obraniti ili ih detektirati.

Obradom ove teme pruža se uvid u razne vrste zlonamjernog softvera preko kojih napadači dolaze do žrtvinih osjetljivih informacija, te se pruža uvid u razne metode i tehnike kojima se takav softver može detektirati.

Naslov završnog rada je *Metode detekcije zlonamjernog softvera pametnih telefona*, a cilj je detaljnije objasniti prijetnje pametnim telefonima u vidu zlonamjernog softvera, te mogućnost njegovog detektiranja. Rad je podijeljen u 7 cjelina:

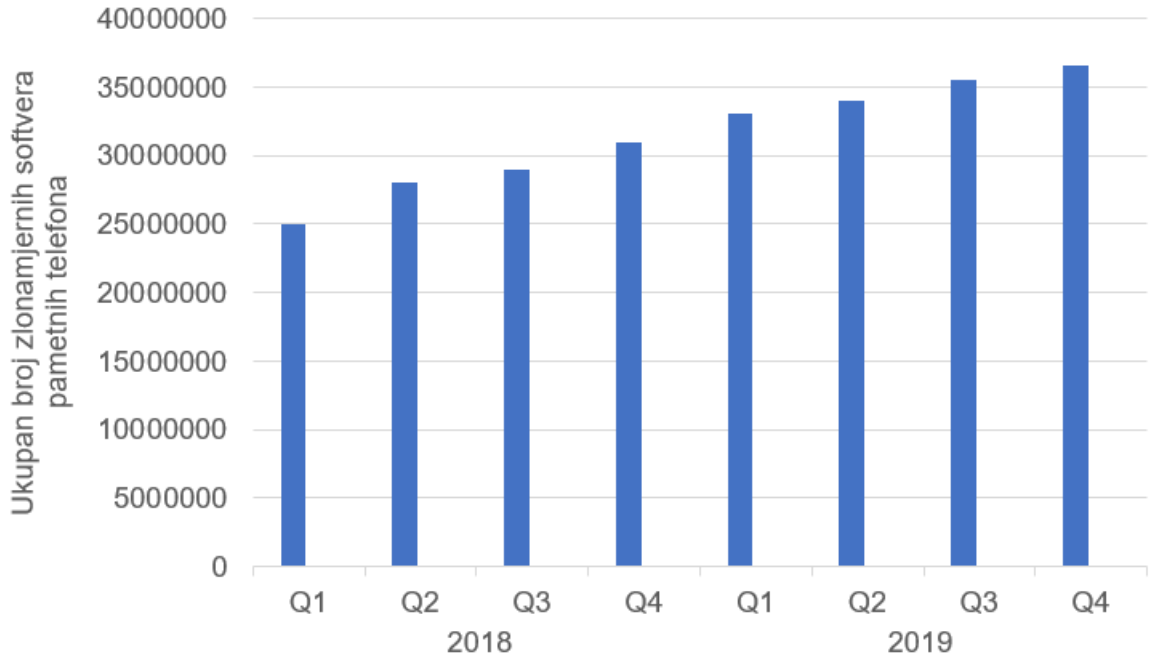
1. Uvod
2. Klasifikacija zlonamjernog softvera
3. Detekcija zlonamjernog softvera pametnih telefona
4. Tehnike analize zlonamjernog softvera
5. Prikaz metoda detekcije zlonamjernog softvera
6. Značajke i postupci reverznog inženjeringa
7. Zaključak

Poglavlje *Klasifikacija zlonamjernog softvera* definira i pobliže objašnjava razne vrste zlonamjernog softvera i njihove karakteristike. Treće poglavlje govori općenito o mogućnostima detekcije zlonamjernog softvera pametnih telefona.

U četvrtom i petom poglavlju se obrađuju razne tehnike analize i prikazuju se metode pomoću kojih se zlonamjerni softver može detektirati, a u šestom poglavlju se detaljnije govori o postupku reverznog inženjeringa i o njegovim karakteristikama.

2. Klasifikacija zlonamjernog softvera

Zlonamjerni softver za mobilne uređaje, kao što mu ime govori, predstavlja zlonamjerni softver koji posebno cilja operativne sustave na pametnim telefonima. Postoje razne vrste inačica zlonamjernog softvera pametnih telefona i različite metode distribucije i zaraze. Za organizacije koje ovise o pametnim telefonima ili koje omogućuju zaposlenima i posjetiteljima korištenje vlastitih uređaja kao dio BYOD (*Bring Your Own Device*) politike, prijetnja je stvarna. Kako se sve više korisnika udaljava od stolnih računala i prelaze većinom na korištenje pametnih telefona napadači koriste takvu priliku i osmišljaju nove napade koje mobilnim uređajima, a i samim korisnicima postaju sve veća briga. Na slici 1 se može vidjeti povećavanje zlonamjernih softvera svake godine po četvrtinama.



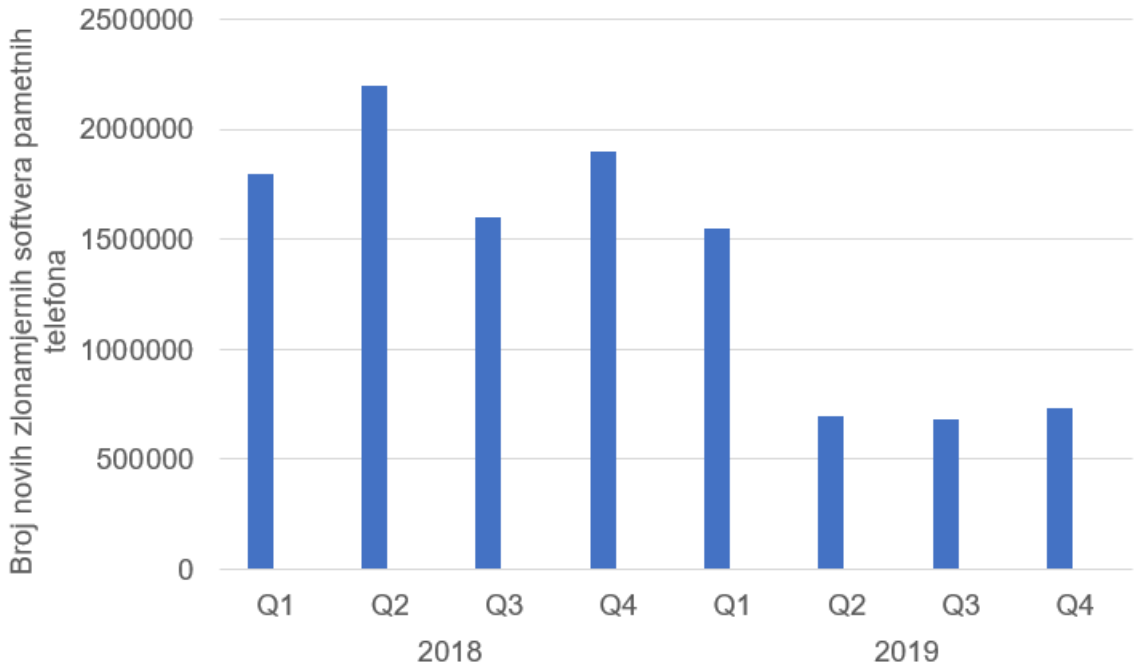
Slika 1. Ukupan broj zlonamjernih softvera pametnih telefona

Izvor: [1]

Prema [2] zlonamjerni softver se klasificira prema šteti koju nanosi zaraženom uređaju, pa tako postoje:

- virus,
- crv,
- trojanski konj,
- Špijunski softver (engl. *Spyware*),

- *Adware*, neželjeni softver koji je namijenjen izbacivanju reklama na zaslon pametnog telefona, najčešće unutar web preglednika, [3],
- *Crimeware*, izraz za softver koji se koristi za počinjenje zločina, poput krađe osobnog identiteta, novca ili vlasničkih podataka, [4],
- *Scareware*, zlonamjerni softver koji manipulira korisnicima da vjeruju da trebaju preuzeti ili kupiti zlonamjeran, ponekad beskoristan softver, [5],
- *Keylogger*, softver namijenjen tajnom praćenju i snimanju (svih) pritisnutih tipki na uređaju, [6],
- *Rootkit*, vrsta zlonamjernog softvera koja je dizajnirana tako da može ostati skrivena na korisnikovom uređaju. No iako to možda korisnik ne primjećuje, takav softver je aktivan uvijek, [7].



Slika 2. Broj novih zlonamjernih softvera pametnih

Izvor: [1]

Na slici 2 se može vidjeti broj novootkrivenih zlonamjernih softvera u 2018. i 2019. godini po četvrtinama. Vidi se da je u 2018. godini postojao veći broj novih zlonamjernih softvera, dok se u 2019. godini taj broj smanjio.

2.1. Virus

Kao i svaki drugi zlonamjerni softver, virus također ugrožava cjelokupno funkcioniranje uređaja. To je zlonamjerni entitet koji ulazi u uređaj iz nekog vanjskog izvora (Interneta ili bilo kojeg drugog uređaja). Nakon što „inficira“ uređaj, on se proširi u skladište i promijeni se u svoju idealnu funkcionalnost. Postoji nekoliko vrsta virusa s različitim stupnjem iskorištavanja ranjivosti, [8].

Ako ostane bez nadzora, virus može prouzrokovati ozbiljnu štetu na uređaju. Osim što ugrožava korisnikov doživljaj pametnog telefona, može uzrokovati i sljedeće elemente:

- može smanjiti ukupnu učinkovitost uređaja i utjecati na njegovu brzinu obrade,
- virus može uzrokovati štetu na pametnom telefonu putem neželjenih skočnih prozora, poruka i upozorenja (npr. smanjiti njegovu brzinu obrade zbog velike količine podataka koju mora obraditi zbog pojavljivanja neželjenih skočnih prozora, poruka i upozorenja),
- također se može isprazniti pohrana uređaja dobivanjem umjetnog prostora i fragmentiranjem njegove pohrane,
- virus također narušava korisnikovu privatnost i može preuzeti njegove podatke (kao što su ID e-pošte, podaci o korisničkom računu itd.),
- može se automatski pretplatiti na *premium* usluge ili slati poruke na *premium* brojeve i tako korisniku uzrokovati financijske gubitke, [8].

Postoje stotine virusa koji mogu utjecati na pametni telefon ili na njegov sustav. Neki od njih su:

- *HummingBad* virus: prvotno je otkriven 2016. godine i od tada je zahvatio preko 10 milijuna uređaja. Ovaj virus može preuzeti aplikacije bez korisnikovog dopuštenja i kupovati u Trgovini Play ili nekim drugim aplikacijama za kupovinu.
- *Gooligan*: virus uglavnom oštećuje popularne aplikacije poput pojačivača sustava, monitora baterije, YouTubea itd. Više od 70% pametnih telefona su tamo najranjiviji. Virus može na korisnikov pametni telefon instalirati neželjene elemente.
- *Gunpoder*: uglavnom utječe na ukorijenjene uređaje koji su koristili *Nintendo* iz izvora treće strane. Ovakav virus može „hakirati“ pametne telefone raznih korisnika i učiniti ih potpuno neaktivnim.

- *Shedun*: jedan je od najpopularnijih zlonamjernih programa i postoji već od 2015. godine. Može prisilno instalirati određene aplikacije na pametni telefon i koristiti korisnikove podatke.
- Policijski virus: poznat i pod nazivom FBI virus, može blokirati cijeli sustav uređaja i učiniti ga neaktivnim, [8].

2.2. Crv

Crv može iskoristiti mobilni uređaj žrtve pokretanjem zlonamjernog programa, a zaraženi mobilni uređaj će zauzvrat skenirati i zaraziti ostale mobilne uređaje u mobilnoj mreži. Crv može obavljati zlonamjerne aktivnosti, poput krađe podataka, slanja vjerodajnica napadačima i slanja *premium* SMS-ova i još puno drugih neželjenih aktivnosti. Manjak sigurnosti u mreži i mjera ublažavanja može uzrokovati širenje napada crva kroz mrežnu infrastrukturu, trošeći ukupnu propusnost i uzrokujući neku drugu štetu, što može dovesti do financijskih gubitaka. Napadači iskorištavaju destruktivno ponašanje i veliko širenje crva kroz mrežu i preuzimaju veliki broj sustava, povećavajući štetu i tako otežavajući pronalaženje crva, [9].

Trenutno je za mobilne inteligentne terminale najučinkovitiji način prevencije zaraze crvom pravodobno „krpanje“ operativnog sustava mobilnih telefona i odgovarajućih mobilnih aplikacija. Ali to je često teško postići jer:

- postoji veliki broj različitih zakrpa za različite mobilne operativne sustave,
- zlonamjerni crv se brzo širi, a broj ranjivosti također se neprestano povećava,
- sigurnosna svijest korisnika mobilnih pametnih telefona je slaba, što može dovesti do nesvjesne infekcije,
- za one mobilne pametne telefone koji su prodani, gotovo je nemoguće provesti objedinjenu nadogradnju verzije operacijskog sustava, [9].

U mobilnom okruženju širenje crva je mnogo teže kontrolirati, jer je prijenos različit. Puno je stvari o kojima treba razmišljati kada se koristi dobroćudni crv za kontrolu i uklanjanje mobilnih zlonamjernih crva, poput problema s popravkom preuzimanih zakrpa, učitavanja i zagušenja mreže dovedenih isporukom dobroćudnih crva i problema pouzdanosti dobroćudnih crva. Zbog toga je na velikom rastućem tržištu mobilnih mreža sigurnosna zaštita nužan preduvjet da se izbjegnu veliki gubici, [9].

2.3. Trojanski konj

Trojanski konj je vrsta zlonamjernog softvera koji se često prerušava u legitimni softver. Trojanski konj mogu „zaposliti“ napadači koji pokušavaju dobiti pristup korisničkim sustavima. Korisnici su obično prevareni nekim oblikom socijalnog inženjeringa da bi dopustili učitavanje i izvršavanje trojanskog konja na svojim sustavima. Nakon aktiviranja, trojanski konj može omogućiti napadačima špijuniranje, krađu osjetljivih podataka i dobivanje pristupa korisnikovu sustavu. Te radnje mogu obuhvaćati:

- brisanje podataka,
- blokiranje podataka,
- promjenu podataka,
- kopiranje podataka,
- povredu izvedbe uređaja i njegovog sustava.

Za razliku od računalnih virusa i crva, trojanski konj nije u stanju samostalno se kopirati, [10].

Trojanski konj je klasificiran prema vrsti radnje koju može obaviti na uređaju:

- *Trojan-Banker*: *Trojan-Banker* programi dizajnirani su za krađu podataka s korisnikovog računa za internetske bankarske sustave, sustave za e-plaćanje te kreditne ili debitne kartice.
- *Trojan-Dropper*: ove programe napadači koriste kako bi instalirali trojanskog konja i/ili viruse ili kako bi spriječili otkrivanje zlonamjernih programa. Nisu svi antivirusni programi sposobni skenirati sve komponente unutar ove vrste trojanskog konja.
- *Trojan-FakeAV*: *Trojan-FakeAV* programi simuliraju aktivnost antivirusnog softvera. Osmišljeni su za iznuđivanje novca od korisnika kako bi zauzvrat otkrivali i uklanjali prijetnje, iako prijetnje koje prijavljuju zapravo ne postoje.
- *Trojan-Ransom*: ta vrsta trojanskog konja može mijenjati podatke na korisnikovom uređaju tako da se uređaj ne pokreće ispravno ili više ne može koristiti određene podatke. Napadači će vratiti performanse uređaja ili deblokirati podatke, nakon što im korisnik plati novac otkupnine koji oni zahtijevaju.

- *Trojan-SMS*: ovi programi mogu korisnika koštati novca slanjem tekstualnih poruka s mobilnog uređaja na *premium* brojeve telefona.
- *Trojan-Spy*: *Trojan-Spy* programi mogu špijunirati kako korisnik koristi svoj uređaj. Na primjer, praćenjem podataka koje unosi putem tipkovnice, snimke zaslona ili dobivanje popisa pokretanih aplikacija, [10].

2.4. Rootkit

Rootkit je zlonamjeran softver koji infekcijom operativnog sustava pametnog telefona postiže svoje štetne ciljeve. Na primjer, *rootkit* se može koristiti za skrivanje zlonamjernih korisničkih procesa, za instaliranje trojanskih konja, *keylogger-a*, za onemogućivanje vatrozida, skenera virusa i sustava za prepoznavanje neovlaštenog upada. Što je još gore, budući da utječu na operativni sustav, *rootkit* može neprimjetno postići svoje zlonamjerne ciljeve i tako ostati neotkriven i zadržati dugoročnu kontrolu nad zaraženim uređajima. Prikrivene tehnike koje su usvojene razvijanjem *rootkit-a* postale su popularne među piscima zlonamjernih softvera, [11].

Iako *rootkit* već dugo predstavlja prijetnju tradicionalnim stolnim računalima jer njihovi operativni sustavi predstavljaju veliku i složenu površinu, sve veća složenost operativnih sustava pametnih telefona čini ih privlačnom metom za autore *rootkit-a*. Kao općenita računalna platforma, pametni telefoni također su ranjivi na brojne prijetnje koje predstavlja *rootkit* stolnim računalima. *Rootkit* može iskoristiti nekoliko sučelja i usluga jedinstvenih pametnim telefonima za pokretanje novih napada s ozbiljnim socijalnim posljedicama. Tri nova *rootkit* napada:

- prebacivanje putem govornog podsustava (*rootkit* inficira GSM podsustav, omogućujući tako napadaču prisluškivanje povjerljivih razgovora žrtve),
- praćenje lokacije s GPS-om (*rootkit* koji ugrožava privatnost lokacije žrtve),
- odbijanje usluge zbog iscrpljenosti baterije (pametni telefoni koriste bateriju i zato imaju ograničene resurse, a ovaj *rootkit* iscrpljuje bateriju uređaja i čini pametni telefon neupotrebljivim kada to korisniku najviše treba, npr. u nekim hitnim slučajevima), [11].

Socijalne posljedice tih napada su pogubne. Pametni telefoni postali su sveprisutni do te mjere da se ljudi oslanjaju na svoje pametne telefone u svakodnevnim aktivnostima. Kao osobni uređaj, korisnici obično vjeruju svojim pametnim telefonima i ne očekuju da će pogrešno raditi. *Rootkit* iskorištava takvo povjerenje korisnika

pametnog telefona za postizanje svojih zlonamjernih ciljeva, a istovremeno ga je i vrlo teško otkriti, [11].

2.5. Keylogger

Iako u napadačeve svrhe, *keylogger* djeluje u kontekstu zlonamjernog softvera, on nije uvijek ilegalan za instaliranje i upotrebu. *Keylogger* je uobičajeni alat za korporacije, koji odjeli informacijske tehnologije koriste za rješavanje tehničkih problema na svojim sustavima i mrežama ili za prikriveno nadgledanje zaposlenika. Isto vrijedi i za roditelje koji žele pratiti aktivnosti svoje djece. U svim takvim slučajevima, ako organizacija ili osoba koja preuzima i instalira *keylogger* zapravo posjeduje uređaj, tada je to potpuno legalno. Na Internetu postoje tisuće komercijalno dostupnih *keylogger-a* koji se oglašavaju upravo za takvu upotrebu, [12].

Ipak, zabrinutost kod *keylogger-a* postoji kada zlonamjerni akteri stoje iza njih, a definitivno ne posjeduju uređaj koji pokušavaju zaraziti. Ovisno o kakvom je *keylogger-u* riječ on može preuzeti sve lozinke koje je korisnik unio, povremeno snimati slike ekrana, snimati web stranice koje korisnik pregledava, pregledavati poslane e-poruke i bilo kakve sesije slanje poruka, kao i osjetljive financijske informacije (kao što su brojevi kreditne kartice, PIN kodovi i bankovni računi), a sve te podatke preko mreže šalje na udaljeno računalo ili web poslužitelj. Tamo osoba koja upravlja programom zapisivanja može sve to preuzeti i bez sumnje poslati nekoj trećoj strani koja to može iskoristiti u kriminalne svrhe, [12].

Ne postoji neki poznati hardverski *keylogger* za pametne telefone, ali i Android i iPhone uređaji su i dalje osjetljivi na softverski *keylogger*. Softverske *keylogger-e* je lakše instalirati na uređaje pa je zato takva vrsta *keylogger-a* mnogo češća. Osim toga, jednom kada *keylogge*r „inficira“ pametni telefon, nadgleda više od aktivnosti tipkovnice, npr. snimke ekrana (e-pošta, tekstovi, stranice za prijavu itd.), kameru telefona, mikrofon, povezane pisače i mrežni promet. Može čak blokirati i korisnikovu sposobnost odlaska na određene web stranice. Svatko tko ima pristup telefonu bez korisnikova znanja, može učitati *keylogger*. Kao što je to slučaj s prijenosnim računalima, tabletima i računalima, korisnici pametnih telefona mogu se zaraziti ako postanu plijenom lažnih poruka putem e-pošte ili nepromišljeno kliknu na neku nepoznatu poveznicu, [12].

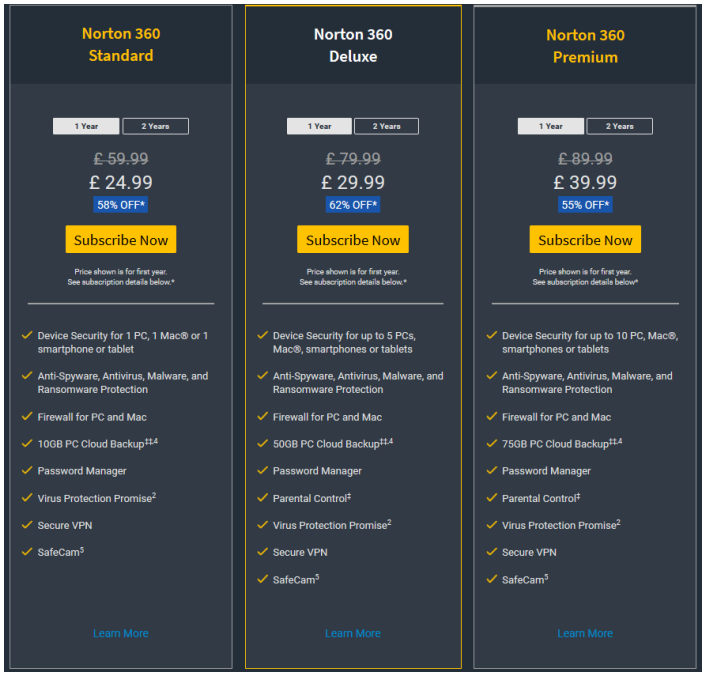
3. Detekcija zlonamjernog softvera pametnih telefona

Antivirusni alati uvelike pomažu u otkrivanju, prepoznavanju, uklanjanju zlonamjernih programa i samoj zaštiti pametnih telefona od virusa i ostalih zlonamjernih programa. Moderni antivirusni alati mogu štititi mobilne uređaje od raznih vrsta zlonamjernih softvera, a to podrazumijeva i sve softvere koji su navedeni u poglavlju *2. Klasifikacija zlonamjernih softvera pametnih telefona*.

3.1. Norton Mobile Security

Norton Mobile Security je sigurnosni alat dizajniran da zadovolji korisnikove potrebe za upotrebom mobilnog uređaja, uključujući i pametne telefone i tablet računala. Kao i mnoge druge vrste računalne opreme, tableti i drugi mobilni uređaji mogu biti izloženi riziku zbog niza prijetnji. Uz pomoć Norton Mobile Security, moguće je smanjiti neke od tih rizika i poboljšati sigurnost na mreži.

Mnogi ljudi ne shvaćaju korist ili potrebu za takvom vrstom sigurnosti. Vjeruju da, sve dok imaju lozinku i budu pažljivi, ne moraju brinuti o svojim uređajima. Međutim, postoji mnogo slučajeva u kojima to nije tako. Na primjer, mnogi ljudi preuzimaju besplatne aplikacije koje često u sebi imaju skrivene komponente koje ne samo da gledaju što korisnik radi, već bi mogle ugroziti i njegov identitet.



Slika 3. Prikaz verzija Norton Mobile Security

Izvor: [13]

Jedna od komponenti *Norton Mobile Security-a* je *App Advisor*. Ovaj ugrađeni alat automatski će odgovoriti korisniku postoje li rizici za njegovu privatnost, kao i za bilo koji slučaj zlonamjernog softvera prilikom preuzimanja iz trgovine Google Play. Čak i ako se ne preuzimaju aplikacije i dalje postoji opasnost prilikom pregledavanja interneta, igranja igara i povezivanja s ljudima na mreži.

Na slici 3 se mogu vidjeti tri verzije koje se mogu instalirati na mobilni uređaj, te se također mogu vidjeti koje pogodnosti dolaze uz svaku verziju.

Brojne su značajke i prednosti ove mobilne zaštite. Takva zaštita uključuje:

- zamišljen je za zaštitu od digitalnih prijetnji koje se mogu dogoditi prilikom preuzimanja, otvaranja ili upotrebe aplikacija na mobilnom uređaju,
- djeluje tako da osigurava višu razinu privatnosti na mreži, štiteći što korisnik radi na mobilnom uređaju,
- za one koji uređaj negdje zaborave ili im prijeti krađa, ovaj alat može pomoći da se uređaj oporavi daljinskim aktiviranjem, te u većini slučajeva pokazuje lokaciju uređaja,
- može pomoći u zaštiti i vraćanju podataka o kontaktima i omogućuje sigurno dijeljenje podataka kad je to potrebno.

3.2. Lookout

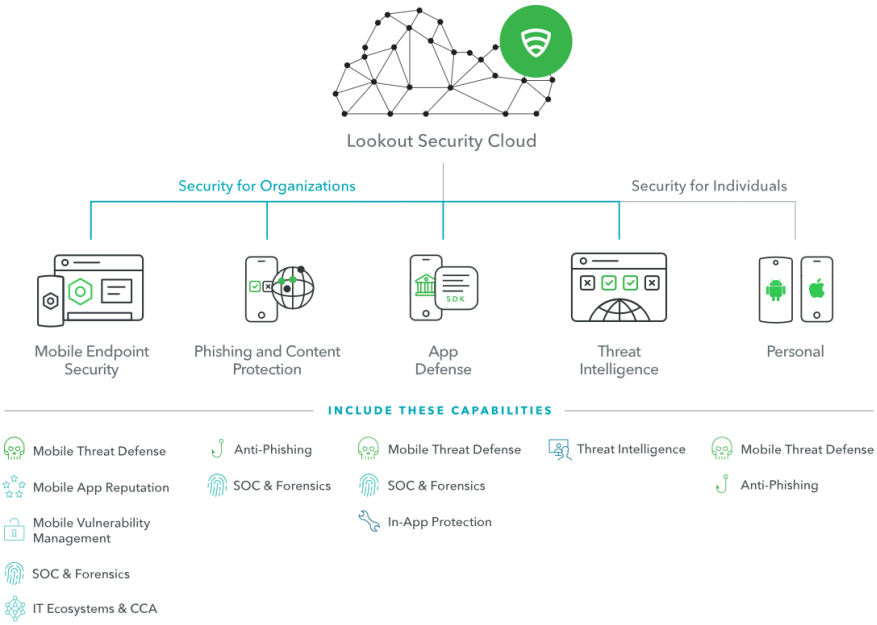
Mnoge organizacije u današnje vrijeme prihvaćaju upotrebu pametnih telefona i tableta kako bi povećale produktivnost na radnom mjestu, a kako osjetljiviji podaci postaju mobilni, sigurnosna pravila organizacije moraju se proširiti i na mobilne krajnje uređaje. *Lookout Mobile Security* olakšava uvid u čitav spektar mobilnog rizika, primjenjuju politike za smanjenje tog rizika i integriraju nova rješenja u svoja već postojeća rješenja za sigurnost i upravljanje mobilnim uređajima.

Lookout-ova tehnologija se sve više i više nadograđuje tako da aplikacija pruža razne pogodnosti za korisnika i omogućuje korisniku sigurno pretraživanje putem interneta, sigurno instaliranje drugih aplikacija itd. Neki od prijetnji protiv kojih se aplikacija „bori“ su:

- trojanski napad i špijunski softver koji mogu izvući podatke iz uređaja,
- ranjivosti u prijenosu i pohrani podataka u aplikaciji,
- rizična ponašanja u aplikacijama koje predstavljaju rizik,

- instalirane aplikacije s nekih drugih poslužitelja, a koje se ne nalaze na službenim trgovinama aplikacija.

Pristup korporativnim podacima trebao bi se odobriti na temelju pretpostavke nultog povjerenja i kontinuirane procjene zdravlja krajnjih točaka. *Lookout* djeluje iza scene, dinamično nadgledajući zdravlje krajnjih točaka dok je korisnik povezan s poduzećem. *Lookout* omogućuje povezivanje s poslovnom infrastrukturom i podacima samo pouzdanim uređajima.



Slika 4. Prikaz proizvoda Lookout Mobile Security

Izvor: [14]

Na slici 4 su prikazani proizvodi za sigurnost organizacije ili proizvodi u osobne svrhe i njihove sposobnosti, a koji su ponuđeni od strane *Lookout-a* za krajnje korisnike.

3.3. Avast Mobile Security

Avast Antivirus je skup raznih sigurnosnih aplikacija koje je Avast razvio za Microsoft Windows, macOS, Android i iOS. Proizvodi Avast Antivirus uključuju besplatnu i plaćenu verziju koje pružaju računalnu sigurnost, sigurnost preglednika, antivirusni softver, vatrozid itd. Avast je u veljači 2015. lansirao besplatni poslovni proizvod, *Avast for Business*. *Avast for Business* je rješenje za više platformi koje uključuje antivirusnu zaštitu, skeniranje web prijetnji, zaštitu preglednika i konzolu za upravljanje oblakom.

Od 2017. Avast je najpopularniji dobavljač antivirusa na tržištu i imao je najveći udio na tržištu antivirusnih aplikacija. U veljači 2018., u testiranju različitih proizvoda protiv zlonamjernih prijetnji tvrtke AV-TEST, *Avast Free Antivirus* zaradio je 6 od 6 bodova u kategoriji „Zaštita“, otkrivši 100% uzoraka zlonamjernog softvera koji su korišteni u ovom testu i zaradio *AV-TEST Certified* pečat. Avast-ova aplikacija za mobilnu sigurnost i antivirus otkrila je 100% uzoraka zlonamjernog softvera u siječnju 2018. u testiranju Android zlonamjernog softvera od strane *AV-Comparatives*.

Tablica 1. Usporedba značajki Avast Mobile Security verzija

	Besplatna verzija	Premium verzija	Ultimate verzija
Blokiranje virusa i zlonamjernih prijetnji	+	+	+
Zaštita od ransomware-a	-	+	+
Provjera sigurnosti Wi-Fi mreže	-	+	+
Izbjegavanje lažnih i nesigurnih web stranica	-	+	+
Sigurnost od krađe identiteta	-	+	+
SecureLine VPN	-	-	+
Cleanup Premium	-	-	+
Passwords Premium	-	-	+

Izvor: [15]

Avast nudi obilje značajki koji pomažu korisniku kada je njegov mobilni uređaj ukraden ili izgubljen kao što su npr.: lociranje uređaja, označivanje uređaja kao izgubljenog, pokretanje sirene, zaključavanje uređaja, brisanje podataka na daljinu i prikazivanje poruka na telefonu. Druga značajka Avast-a omogućuje korisniku šifriranje fotografija pomicanjem ili slanjem izravno u biblioteku fotografija Avast aplikacije. Avast posjeduje i značajku za skeniranje Wi-Fi mreže na koju se korisnik želi povezati i pri tome bilježi sve ranjivosti te mreže. U tablici 1 je prikazana usporedba značajki tri Avast Mobile Security verzije (besplatna, Premium i Ultimate verzija).

3.4. McAfee Mobile Security

McAfee Mobile Security ima mnogo sigurnosnih značajki i dodataka. Učinkovit je u blokiranju i otkrivanju većine napada i zlonamjernih softvera već na samom početku. Uz McAfee korisnik dobiva sve potrebne značajke antivirusa kao što su zaštita u stvarnom vremenu i siguran vatrozid i još nekoliko dodataka kao što su optimizacija izvedbe i upravitelj lozinki. McAfee štiti od virusa, zlonamjernog softvera, špijunskog softvera i *ransomware* napada, a također štiti korisnika od sumnjivih ili ranjivih web stranica. McAfee je bio 99% uspješan u detekciji i prevenciji napada zlonamjernog softvera nultog dana.

Tablica 2. Usporedba McAfee Mobile Security verzija

	Besplatna verzija	Standard verzija	Plus verzija
Sigurnosni pregled	+	+	+
Provjera privatnosti	+	+	+
Zaštita od krađe	+	+	+
Sigurna Wi-Fi mreža	+	+	+
Alat za pojačanje baterije	+	+	+
Alat za unapređivanje memorije	+	+	+
Alat za čišćenje pohrane	+	+	+
Praćenje uporabe podataka	+	+	+
Sigurnosno kopiranje medija	-	+	+
Telefonska podrška	-	+	+
Bez oglasa u aplikaciji	-	+	+
Način rada za gosta	-	+	+
Sigurnosno zaključavanje aplikacije	-	+	+
Sigurno pretraživanje weba	-	+	+
Wi-Fi Guard VPN	-	-	+

Izvor: [16]

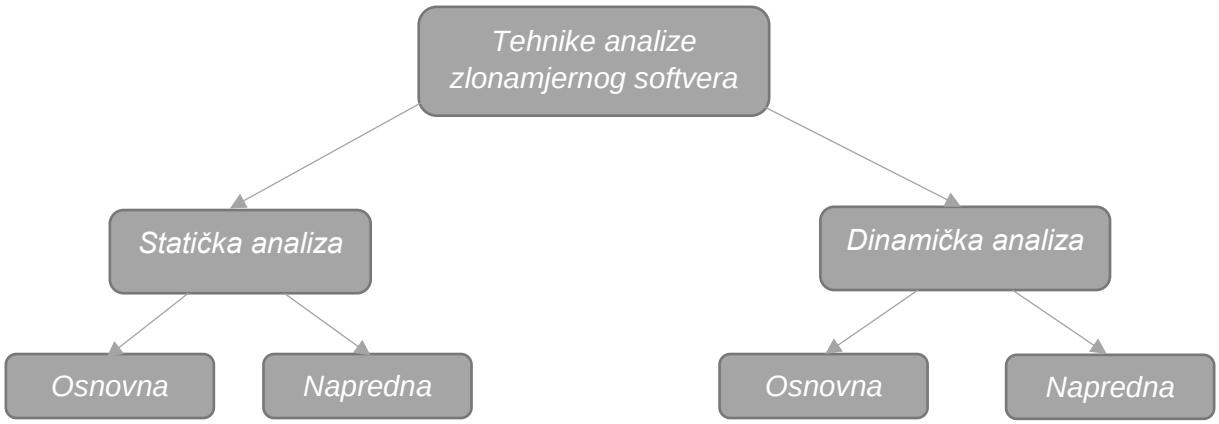
Iako su McAfee rezultati mnogo bolji sada nego prethodnih godina, Norton *360* još uvijek sveukupno ostvaruje bolje rezultate. McAfee pruža korisnicima širok raspon značajki, od sigurnog pregledavanja do trajnog brisanja datoteka, te uključuje:

- optimizaciju performansi,
- šifriranu pohranu (128-bitna enkripcija),
- Home Network Security (vatrozid),
- upravitelj lozinki,
- kompatibilnost sa više uređaja.

Takva uspješnost u detekciji je ista kao i kod drugih vrhunskih brendova kao što su Avast i Bitdefender. U tablici 2 prikazana je usporedba tri verzije McAfee Mobile Security (besplatna verzija, standardna verzija i Plus verzija).

4. Tehnike analize zlonamjernog softvera

Na temelju značajki korištenih za klasificiranje aplikacije, analiza se može kategorizirati kao statička i dinamička. Statička analiza vrši se bez pokretanja aplikacije. Primjeri statičkih značajki uključuju: dozvole, API pozive koji se mogu izdvojiti iz datoteke AndroidManifest.xml. Dinamička analiza bavi se značajkama koje su izvučene iz aplikacije tijekom rada, uključujući: mrežni promet, potrošnju baterije, IP adresu itd. Treći tip analitike je hibridna analiza koja kombinira značajke statičke i dinamičke tehnike.



Slika 5. Podjela tehnika analize zlonamjernog softvera

4.1. Statička analiza

U statičkoj analizi, značajke se izdvajaju iz datoteke aplikacije bez izvršavanja aplikacije. Ova metodologija je ekonomična i vremenski učinkovita jer se aplikacija ne izvršava. Ali istodobno, ova analiza „pati“ zbog tehnika obijanja koda koje autori zlonamjernog softvera koriste kako bi izbjegli statičke tehnike detekcije. Jedna od vrlo popularnih tehnika je *Update Attack*: na mobilnom uređaju instalira se dobroćudna aplikacija, a kada se aplikacija ažurira, zlonamjerni sadržaj se preuzima i instalira kao dio ažuriranja. To se ne može otkriti statičkim tehnikama analize jer će se skenirati samo dobroćudna aplikacija.

Statička analiza se može podijeliti u dvije kategorije: osnovna statička analiza i napredna statička analiza. Osnovna statička analiza sastoji se od pregledavanja izvršne datoteke bez uvida u stvarne upute. Ova vrsta analize može provjeriti jesu li podaci zlonamjerni, dati podatke o njegovoj funkcionalnosti i ponekad dati podatke koji omogućuju stvaranje jednostavnih mrežnih potpisa. Osnovna statička analiza je elementarna i prilično brza, ali uglavnom je beskorisna u odnosu na složeni zlonamjerni

softver. Napredna statička analiza sastoji se od obrnutog razvoja komponenti zlonamjernog softvera, preuzimanja izvršne datoteke i pregleda programa kako bi se saznalo što program radi. Procesor izvršava upute, pa ova analiza točno opisuje što program radi. Ipak, za naprednu statičku analizu potrebno je puno iskustva u rastavljanju i konstrukciji koda, [17].

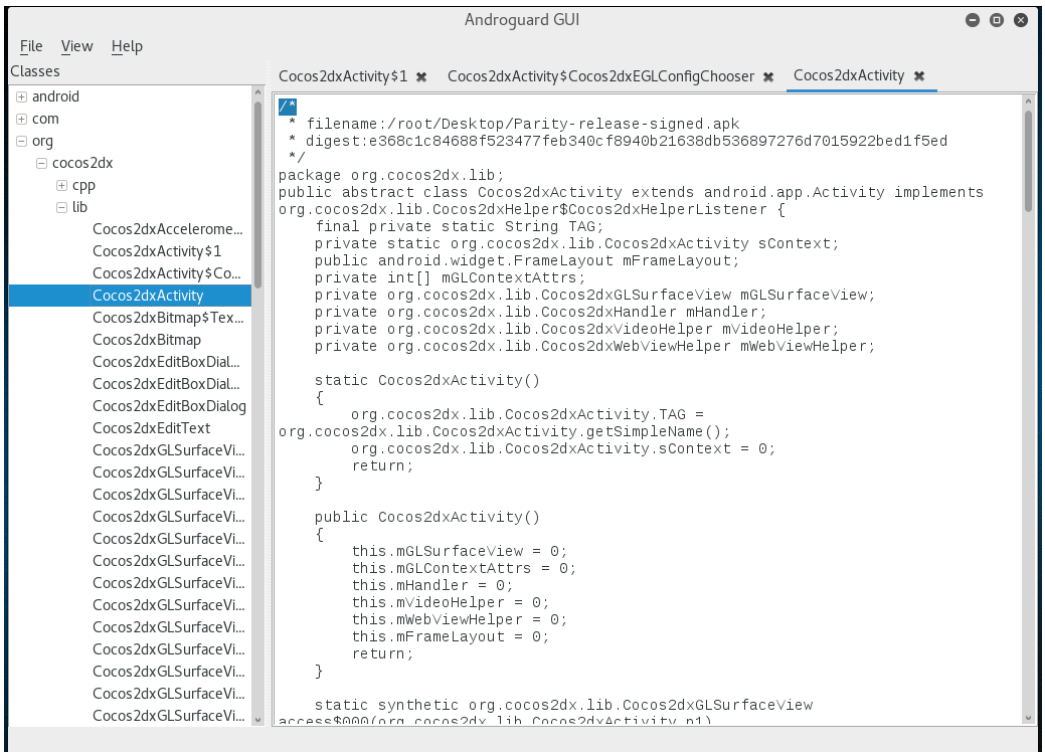
Autori u [18] predložili su metodu koja otkriva i klasificira Android zlonamjerni softver koristeći statičku analizu s kombinacijom informacija o napadačima. Učinkovitost otkrivanja zlonamjernog softvera Android poboljšana je integriranjem podataka napadača i kategoriziranjem nelegitimnih aplikacija u homogene klase. Sustav je mogao prepoznati zlonamjerni softver s 98% točnosti. Opći nedostatak te metode može se poboljšati dodavanjem funkcionalnosti dinamičke analize.

Prema [19] autori su predložili integrirani statički okvir koristeći tehniku filtriranja koja se sastoji od četiri sloja za identificiranje i procjenu mobilnog zlonamjernog softvera na Androidu s gotovo 99% točnosti. Također su predstavili pristup koji spaja statičke logičke strukture i dinamičke informacije o vremenu pokretanja kako bi se otkrio zlonamjerni softver. Rezultati su pokazali da je pristup jednostavan za implementaciju i da ima male troškove. Predstavili su i novi pristup pomoću dozvole i API-ja. Zatraženi metapodaci izvlače se iz svake datoteke i klasificiraju se u uobičajenu ili zlonamjernu aplikaciju pomoću najbližeg susjeda. Model je postigao 97,87% točnosti.

Statički pristup u osnovi se temelji na potpisu zlonamjernog programa i dozvoli aplikacije. U pristupu utemeljenom na potpisu, dojmovi zlonamjernog softvera generiraju se i pohranjuju u bazu podataka kao model. Nepoznata aplikacija uspoređuje se s bilo kojom postojećom aplikacijom i ako rezultat sličnosti premaši postavljeni prag, označava se zlonamjernim softverom. Metoda temeljena na potpisu ne može učinkovito otkriti novi nepoznati zlonamjerni softver. Pristup temeljen na dozvoli koristi se i na strani poslužitelja i na strani klijenta za provjeru autentičnosti instanci aplikacija. Ponašanje instanci aplikacije kategorizirano je ili kao normalno ili kao zlonamjerno na strani poslužitelja. Za izvršavanje statičke analize koriste se brojni alati, te su zbog toga u sljedećim navodima opisani neki od takvih alata.

Alat Androguard

Androguard je alat koji se temelji na programskom jeziku Python i koristi se za reverzni inženjering Android aplikacija. To podrazumijeva uzimanje sirovih datoteka Android paketa (.apk) aplikacije i njihovo raščlanjivanje kako bi se mogla obaviti analiza. Tu se može provesti testiranje prodora zlonamjernog softvera i ranjivosti. Androguard podržava Linux, Windows i OSX sve dok je Python instaliran u sustav. Pokretanje Androguarda u sustavu Windows bavi s velikim brojem ovisnosti, a radi jednostavnosti, preporučuje se da se alat pokreće na Linux operacijskom sustavu korištenjem Virtual Machine.



Slika 6. Grafičko sučelje alata Androguard

Izvor: [20]

Alat APKTool

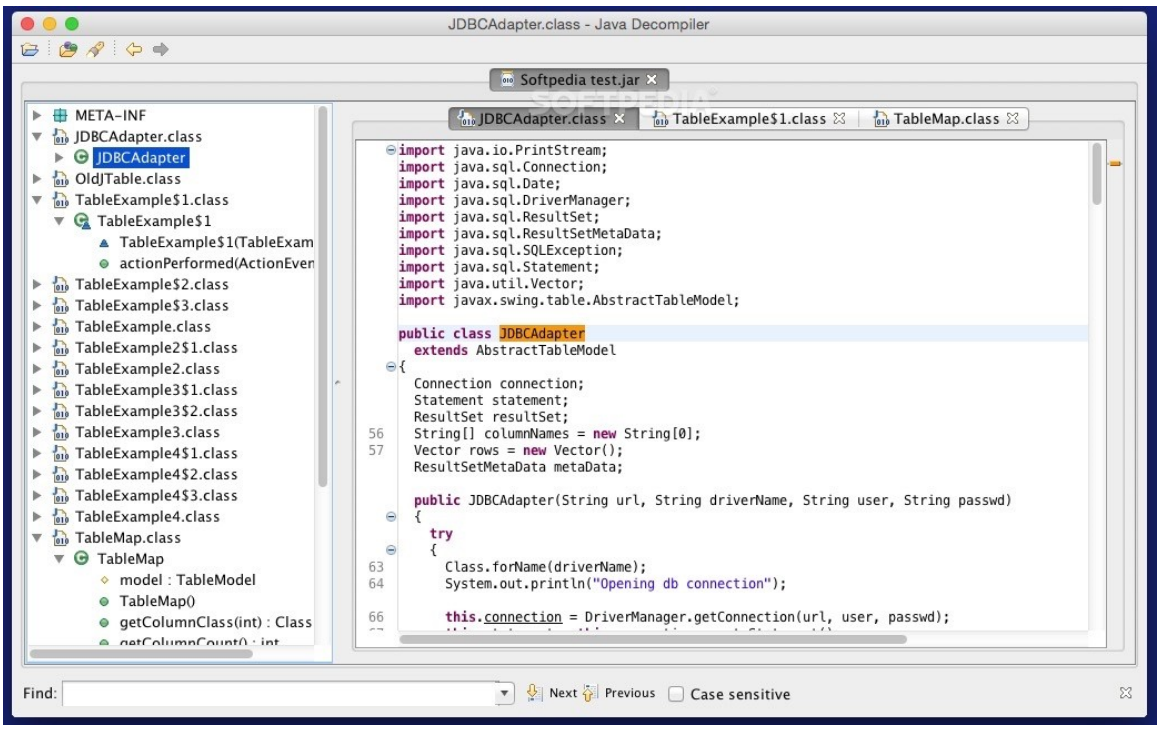
Alat za reverzni inženjering zatvorene, binarne Android aplikacije. Može dekodirati resurse do gotovo originalnog oblika i obnoviti ih nakon što se unesu neke izmjene te omogućuje uklanjanje pogrešaka malog koda korak po korak. Također olakšava rad s aplikacijom zbog strukture datoteka poput projekata i automatizacije nekih ponavljajućih zadataka kao što je izgradnja apk-a itd. Može se koristiti za lokalizaciju, dodavanje nekih značajki ili podrške za prilagođene platforme i u druge svrhe.

Alat Dex2jar

Dex2Jar je slobodno dostupan alat za rad s Android *.dex* i Java *.class* datotekama. Temeljna značajka Dex2Jar je pretvoriti datoteku *groups.dex* APK-a u class.jar ili obrnuto. Dakle, moguće je pogledati izvorni kod Android aplikacije pomoću bilo kojeg Java dekompajlera, a da on je u potpunosti čitljiv. Dobivaju se *.class* datoteke, a ne stvarni Java izvorni kod koji je napisao programer aplikacije. Također, moguće je dobiti *.smali* datoteke izravno iz datoteke *class.dex* ili obrnuto. To znači da se može promijeniti izvorni kod aplikacije koja izravno radi s ovim formatom.

Alat JD-GUI

JD-GUI samostalni je grafički program koji prikazuje Java izvorne kodove datoteka *.class*. Mogu se pregledavati obnovljeni izvorni kodovi s JD-GUI-om za trenutni pristup metodama i poljima. Ako se otvori *.jar* datoteka s JD-GUI-om, može se pogledati izvorni kod aplikacije koji je u čitljivom formatu, a također je vrlo lako kretati se kroz kod.



Slika 7. Grafičko sučelje alata JD-GUI

Izvor: [21]

4.2. Dinamička analiza

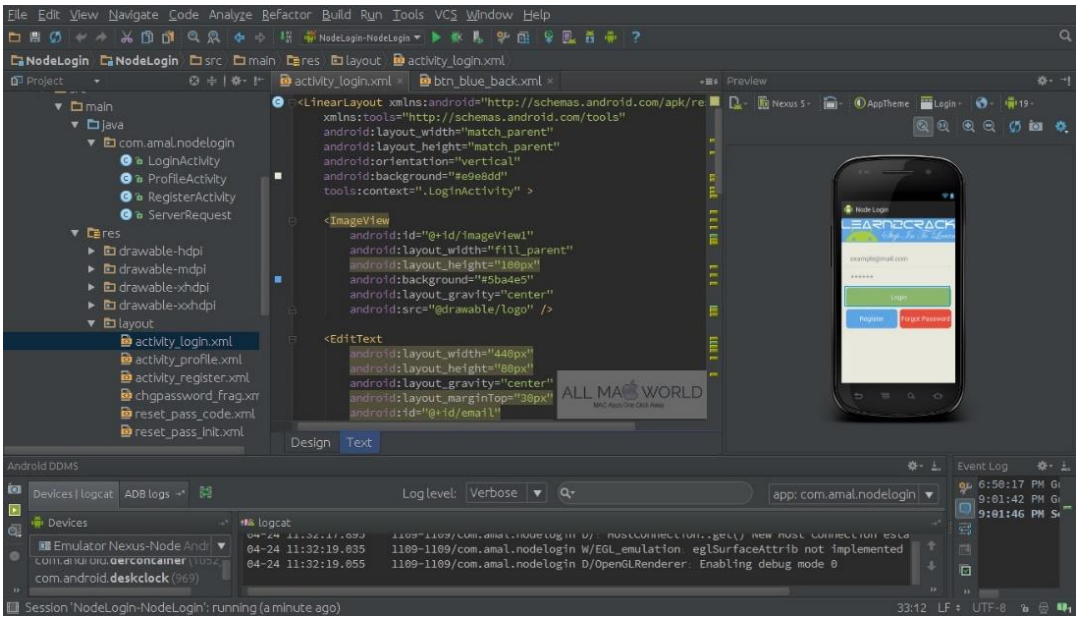
Dinamička analiza se također kao i statička analiza može podijeliti u dvije kategorije: osnovna dinamička analiza i napredna dinamička analiza. Glavne metode dinamičke analize uključuju pokretanje zlonamjernog softvera i nadziranje njegovog ponašanja u sustavu radi uklanjanja infekcije, stvaranja učinkovitih potpisa ili oboje.

Prije sigurnog pokretanja zlonamjernog softvera, mora se stvoriti okruženje koje će omogućiti da se zlonamjerni programi izvode bez opasnosti od oštećenja sustava ili mreže. Kao i osnovne metode statičke analize i osnovne tehnike dinamičke analize može koristiti većina ljudi bez detaljnog znanja o programiranju, ali neće biti korisne kod svih zlonamjernih programa. Napredna dinamička analiza koristi program za uklanjanje pogrešaka kako bi se provjerilo interno stanje zlonamjernog programa. Napredne metode dinamičke analize pružaju još jedan način za izvlačenje detaljnih informacija iz izvršne datoteke. Ove su metode najkorisnije kada se pokušavaju dobiti informacije koje je teško sastaviti s drugim metodama. U priručniku *OWASP Mobile* može se naučiti kako koristiti naprednu dinamičku analizu zajedno s naprednom statičkom analizom za cjelovitu analizu sumnjivih zlonamjernih programa, [17].

Autori u [22] proučavali su redoslijed pokretanja sistemskih poziva duboko zasnovan na prepoznavanju uzoraka kako bi otkrili zlonamjerni softver na Android uređaju. Rezultat je pokazao 95,8% točnosti. Također su predstavili pristup koji promatra dinamično ponašanje aplikacija temeljeno na zapisima poziva sustava. Zapisnici sistemskih poziva svake promatrane aplikacije koriste se za izgradnju skupa podataka koji klasificiraju aplikaciju kao zlonamjernu ili normalnu. Novi pristup koji je temeljen na ponašanju koristi se za identificiranje Android zlonamjernog softvera koristeći sekvence sistemskih poziva kao ponašanje zlonamjernog softvera. Njegov je cilj otkriti i zaustaviti bilo koji zlonamjerni softver za vrijeme izvođenja. Iako je točnost otkrivanja vrlo visoka, stopa performansi je vrlo niska. Za izvršavanje dinamičke analize također se koriste brojni alati, a u sljedećim navodima opisani su neki od njih.

Alat Android SDK Manager

Android SDK Manager je alat koji se osim u sigurnosti koristi i kod razvijanja Android aplikacija. Alat omogućava skidanje i instaliranje različitih API-ja te omogućuje i njihovo pokretanje na emulatoru. Emulator se koristi kako bi se pokretale na njemu zlonamjerne aplikacije i kako bi se pomoću njega pratilo ponašanje takve aplikacije.

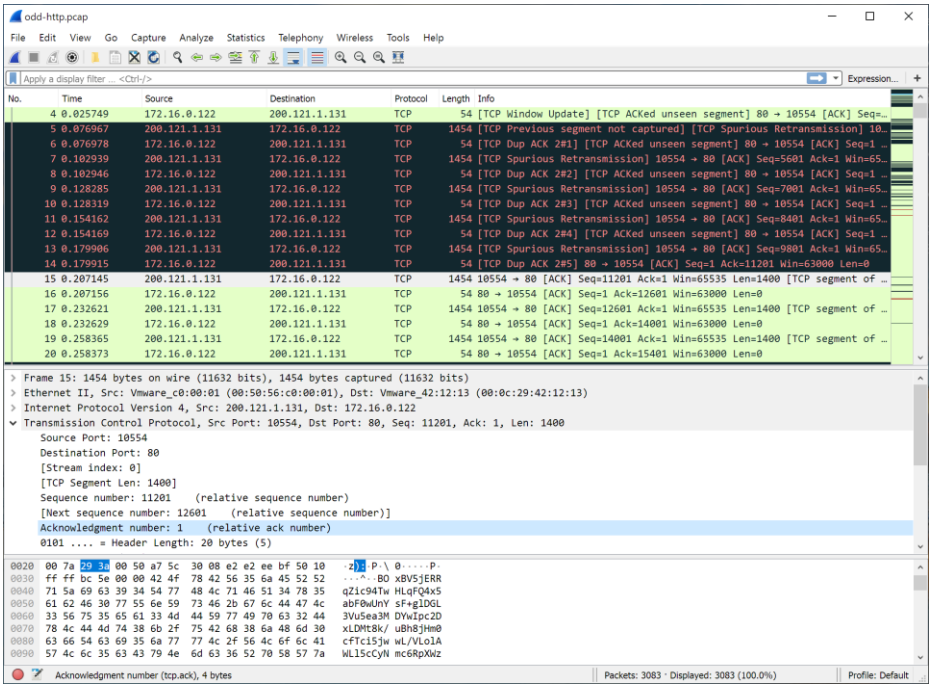


Slika 8. Grafičko sučelje alata Android SDK Manager

Izvor: [23]

Alat Wireshark

Wireshark je besplatni analizator paketa otvorenog koda. Koristi se za rješavanje problema s mrežom, analizu, razvoj softvera i komunikacijskih protokola i obrazovanje. Wireshark je program za prikupljanje podataka koji "razumije" strukturu (enkapsulaciju) različitih mrežnih protokola. Može raščlaniti i prikazati polja, zajedno s njihovim značenjima kako je specificirano u različitim mrežnim protokolima. Wireshark može obojiti pakete na temelju pravila koja se podudaraju s određenim poljima u paketima kako bi korisnik u prvom redu mogao prepoznati vrste prometa. Tako zelena boja označava TCP (eng. *Transmission Control Protocol*) pakete, tamnoplava DNS (eng. *Domain Name System*), svijetloplava UDP (eng. *User Datagram Protocol*) i crna označava TCP pakete kod kojih je došlo do neke pogreške.



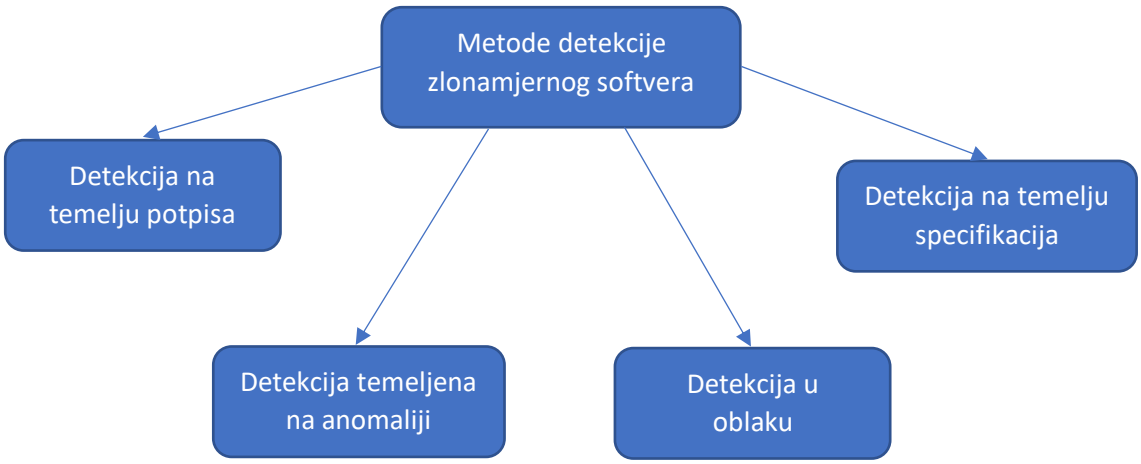
Slika 9. Grafičko sučelje alata Wireshark

Izvor: [24]

Zbog takvih karakteristika Wireshark se koristi za analizu mrežnog prometa mobilnih uređaja prilikom dinamičke analize aplikacije kako bi se utvrdilo pristupa li aplikacija Internetu i da li prima/šalje pakete koji su zlonamjerni.

5. Prikaz metoda detekcije zlonamjernog softvera

Metode detekcije mogu se klasificirati na: detekciju na temelju potpisa, detekciju temeljena na anomaliji, detekciju na temelju specifikacija i detekciju u oblaku. Danas većina metoda detekcije zlonamjernog softvera upotrebljava statički izvađene podatke iz datoteke AndroidManifest.xml, kao i dinamički dobivene informacije iz mrežnog prometa. Štoviše, većina trenutnih sustava za detekciju su opremljeni bazom podataka redovitih izraza koji određuju sekvence bajtova ili uputa koji se smatraju zlonamjernim, a uglavnom se temelje na sintaktičkim potpisima i koriste tehnike statičke analize. Nažalost, statičkim tehnikama se zlonamjerne aplikacije mogu „sakriti“ koristeći tehnike poput polimorfizma, metamorfizma i dinamičkog učitavanja koda.



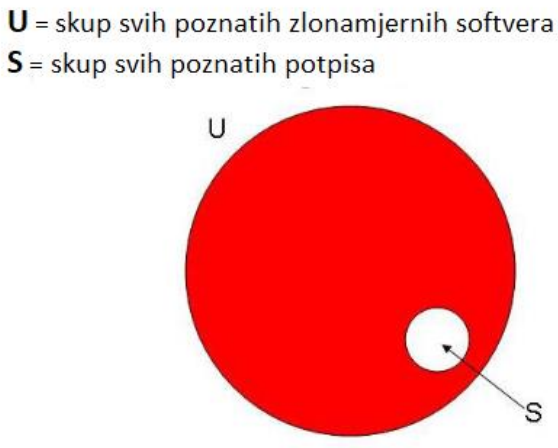
Slika 10. Podjela metoda detekcije zlonamjernog softvera

5.1. Detekcija na temelju potpisa (Signature-based detection)

Detekcija na temelju potpisa pokušava modelirati zlonamjerno ponašanje zlonamjernog softvera i koristi taj model u otkrivanju zlonamjernog softvera. Zbirka svih tih modela predstavlja znanje za detekciju na temelju potpisa. Takav model zlonamjernog ponašanja često se naziva potpisom. U idealnom slučaju, potpis bi trebao biti u stanju prepoznati bilo koji zlonamjerni softver koji pokazuje zlonamjerno ponašanje određeno potpisom. Kao i svi podaci koji postoje u velikim količinama za koje je potrebno pohranjivanje i za potpise je potrebno spremište. To spremište predstavlja svo znanje koje metoda bazirana na potpisu ima, a odnosi se i na otkrivanje zlonamjernog softvera.

Trenutno se prvenstveno oslanja na ljudsko znanje u stvaranju potpisa koji predstavljaju zlonamjerno ponašanje izloženo u programima. Jednom kada je potpis

kreiran, on se dodaje znanju metode zasnovane na potpisu (tj. spremištu). Jedan od glavnih nedostataka metode za otkrivanje zlonamjernog softvera utemeljenog na potpisu je taj što ne može otkriti napade nultog dana, to je napad za koji u spremištu nema odgovarajućeg potpisa. Slika 11 prikazuje glavni nedostatak metoda temeljenih na potpisu. Budući da je skup mogućih zlonamjernih ponašanja, U, beskonačno velik, ne postoje poznate tehnike za točno predstavljanje U putem potpisa. Nadalje, spremište potpisa nije ni približno veličini U, [25].



Slika 11. Ilustracija zašto je detekcija temeljena na potpisu nedovoljna

Izvor: [25]

Drugi nedostatak metoda temeljenih na potpisu je ta što je za razvijanje potpisa obično potrebno ljudsko uključivanje/stručnost. To ne samo da omogućuje uvođenje ljudske pogreške, već zahtijeva znatno više vremena nego ako je razvoj potpisa potpuno automatiziran. S obzirom na to da neki zlonamjerni softver ima mogućnost izuzetno brzog širenja, mogućnost brzog razvijanja točnog potpisa postaje najvažnija. Automatizirani razvoj potpisa postoji, ali na tom području je potrebno još puno posla, [25].

Neki od primjera metoda detekcije na temelju potpisa pokušavaju utjecati na to da je velik dio novih zlonamjernih softvera derivat prethodnih zlonamjernih softvera, tj. novi softveri su dobiveni preradom prethodnih. Kako bi se iskoristilo takvo zapažanje, potpisi zlonamjernog softvera moraju biti izrađeni na način da bilježe zlonamjernu suštinu ili nepromjenljivost zlonamjernog softvera koji je modeliran. Razlog zbog kojeg se koristi ova paradigma stvaranja potpisa je taj da detektori zlonamjernog softvera budu manje podložni obmanama. Drugi istaknuti razlog za izgradnju potpisa na ovaj

način je minimiziranje broja potpisa zlonamjernog softvera koji su pohranjeni u spremištu. Iako trenutno pohranjivanje nije problem, pohranjivanje s vremenom potencijalno može postati ozbiljno jer će to izravno utjecati na vremensku složenost detektora zlonamjernog softvera. U sljedećim navodima prikazuju se alati ili tehnike pomoću kojih su autori provodili detekciju zlonamjernog softvera, [25].

Metoda na osnovu potpisa za otkrivanje crva

Autori u [26] predlažu metodu na osnovu potpisa za otkrivanje crva koja se temelji na poznatom zlonamjernom ponašanju. Autori predstavljaju četiri različita ponašanja. Osnovni potpisi su oni koji se mogu prepoznati nadgledanjem protoka podataka koji ulaze i izlaze iz jednog čvora. Osnovni potpis je kad se server promijeni u klijenta. Budući da se crv mora razmnožavati, nakon ugrožavanja poslužitelja, mora se ponovno ponašati kao klijent drugom domaćinu u nadi da će putem iste ranjivosti tipično zaraziti više strojeva. Ovaj pristup otkrivanju nije toliko učinkovit ako se primjenjuje u okruženju „peer-to-peer“. Autori su također analizirali potpis poslužitelja-klijenta. Otkriveno je da je potpis poslužitelja na klijenta savršeno osjetljiv na aktivne crve koji mijenja poslužitelj u klijenta.

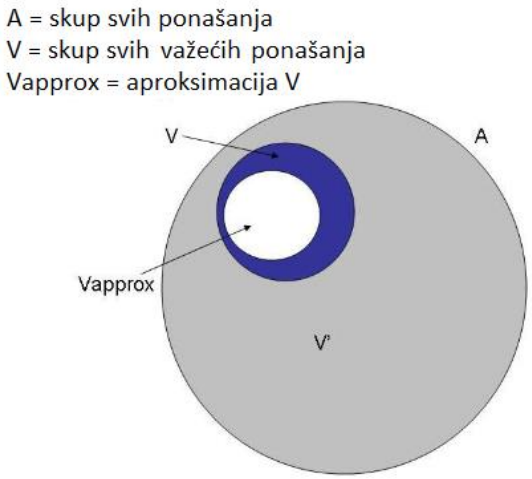
SAVE

Autori su u [27] predložili metodu koja se naziva *Static Analysis for Vicious Executables* (SAVE). Oblik potpisa određenog virusa daje redoslijed Windows API poziva. Svaki API poziv predstavljen je 32-bitnim brojem. Najznačajniji 16 bita odgovara modulu kojem API poziv pripada, dok najmanje značajnih 16 bita odgovara položaju funkcije API-ja u vektoru API funkcija. Euklidska udaljenost izračunava se između poznatih potpisa i slijeda API poziva koji se nalaze u ispitivanom programu. Prosjek triju funkcija daje sličnost API sekvence programa koji se pregledava s potpisom iz spremišta. Ako je razlika 10 posto ili manja, tada se program koji se pregledava označava kao zlonamjerni. Autori su uspoređivali SAVE s 8 detektora zlonamjernog softvera. SAVE je uspoređivan sa detektorima Norton, McAfee Unix Scanner, McAfee, Dr. Web, Panda, Kasperksy, F-Secure i Anti Ghostbusters. Testirali su sve skenere na varijante W32.Mydoom, W32.Bika, W32.Beagle i W32.Blaster.Worm. SAVE je bio jedini detektor koji je u studiji uspio otkriti sve varijante spomenutih zlonamjernih softvera.

5.2. Detekcija temeljena na anomaliji (Anomaly-based detection)

Detekcija temeljena na anomaliji obično se odvija u dvije faze - fazi treninga (fazi učenja) i fazi otkrivanja (nadgledanja). Tijekom faze treninga detektor pokušava naučiti normalno ponašanje. Detektor može učiti ponašanje domaćina ili programa koji se pregledava ili kombinaciju oboje. Ključna prednost detekcije temeljene na anomaliji je njezina sposobnost otkrivanja napada nultog dana. Napadi nultog dana su napadi koji su detektoru zlonamjernog softvera nepoznati. Dva temeljna ograničenja ove tehnike su njena visoka stopa lažnog alarma i složenost uključena u određivanju značajki koje treba naučiti u fazi treninga, [25].

Slika 12 prikazuje zašto samo otkrivanje na temelju anomalije nije dovoljno za otkrivanje zlonamjernog softvera. Kao što je prikazano, V je skup svih važećih ponašanja sustava izvedenih iz skupa nekonfliktnih zahtjeva, a V 'je skup svih nevažećih ponašanja.



Slika 12. Karakterizacija ponašanja u detekciji temeljenoj na anomaliji

Izvor: [25]

Približna vrijednost svih valjanih ponašanja načinjenih metodama detekcije temeljene na anomaliji prikazana je na slici kao skup Vapprox-a. Budući da je Vapprox aproksimacija, valjano ponašanje može biti označeno kao zlonamjerno pomoću metoda detekcije temeljene na anomaliji. Na primjer, ako se izuzetak nikad ne primijeti tijekom faze treninga, iznimka koja predviđa fazu praćenja izazvala bi pogrešan alarm. To doprinosi visokoj lažnoj pozitivnoj stopi koja je obično povezana s metodama detekcije na temelju anomalije. Mogućnost da sustav pokaže prethodno neviđeno ponašanje tijekom faze otkrivanja nije nula. Stoga vjerojatnost tehnike utemeljene na

anomaliji lažne pozitivne vrijednosti nije jednaka nuli. Razvijanje boljih aproksimacija za normalno ponašanje računalnog sustava otvoren je problem informatike. U sljedećim navodima prikazuju se alati ili tehnike pomoću kojih su autori provodili detekciju zlonamjernog softvera, [25].

PAYL

Autori u [28] predstavljaju PAYL, alat koji izračunava očekivani teret za svaku uslugu (port) u sustavu. Stvara se raspodjela frekvencija bajta koja omogućava da se razvije centroidni model za svaku od usluga domaćina. Taj centroidni model izračunava se tijekom faze učenja. Detektor uspoređuje dolazna korisna opterećenja sa centroidnim modelom, mjereći mahalanobijsku udaljenost između njih. Mahalanobijska udaljenost uzima u obzir ne samo srednje vrijednosti karakterističnog svojstva, već i varijancu i kovarijancu, što dovodi do snažnije statističke mjere simultanosti. Ako je dolazni korisni teret predaleko od modela centroida (velika mahalanobijska udaljenost), tada se korisni teret smatra zlonamjernim. Autori su koristili alat PAYL u laboratoriju gdje se od 201 napada, njih 97 trebalo otkriti njihovom tehnikom. Njihova tehnika je mogla detektirati 57 od 97 napada što je bilo otprilike 60%.

Fileprint analiza

Autori u [29] opisuju *Fileprint* analizu (n-gram) kao sredstvo za otkrivanje zlonamjernog softvera. Pretpostavka autora je da benigne datoteke imaju predvidljive redovne bajt sastave za svoje vrste. Tako, na primjer, benigne .pdf datoteke imaju jedinstvenu distribuciju bajtova koja se razlikuje od .exe ili .doc datoteka. Svaka datoteka koja se pregledava i za koju se smatra da se previše razlikuje od određenog modela ili skupa modela, označena je kao sumnjiva. Te sumnjive datoteke označene su za daljnju inspekciju nekim drugim mehanizmom ili odlučivanjem kako bi se utvrdilo ako je zapravo zlonamjerna. Autori su otkrili da je primjena 1-gram analize na PDF (Portable Document Format) datoteke s ugrađenim zlonamjernim softverom prilično učinkovita u odnosu na COTS AV (Commercial Off-The Shelf Anti-Virus) skener. 1-gram analiza pokazala je stopu otkrivanja između 72,1 posto i 94,5 posto za PDF datoteke s ugrađenim zlonamjernim softverom, dok je COTS AV skener imao efektivnu stopu detekcije nula.

5.3. Detekcija na osnovi specifikacija (Specification-based detection)

Detekcija na temelju specifikacija je vrsta detekcije temeljene na anomaliji koja pokušava uklopiti tipičnu visoku stopu lažnog alarma povezanu s većinom tehnika detekcije na temelju anomalije. Budući da je detekcija temeljena na specifikacijama derivat detekcije temeljene na anomaliji, slika 12 također vrijedi za detekciju na temelju specifikacija. Umjesto da pokuša približiti implementaciju aplikacije ili sustava, detekcija na temelju specifikacija pokušava približiti zahtjeve za aplikaciju ili sustav. Detekcija temeljena na inspekcijskom nadzoru, faza treninga je postizanje nekog skupa pravila, koji određuje sva valjana ponašanja koja bilo koji program može pokazati za zaštićeni sustav ili program u inspekciji, [25].

Glavno ograničenje detekcije temeljeno na specifikacijama je ta što je često teško potpuno i točno odrediti čitav niz valjanih ponašanja koje sustav treba pokazati. Može se zamisliti da čak i za umjereno složeni sustav, potpuna i točna specifikacija njegovih valjanih ponašanja može biti neusporediva. Čak i kada je lako izraziti specifikacije za sistemski prirodni jezik, često je to teško izraziti u obliku pogodnom za stroj. U sljedećim navodima prikazuju se alati ili tehnike pomoću kojih su autori provodili detekciju zlonamjernog softvera, [25].

Automatizirano otkrivanje ranjivosti u privilegiranim programima

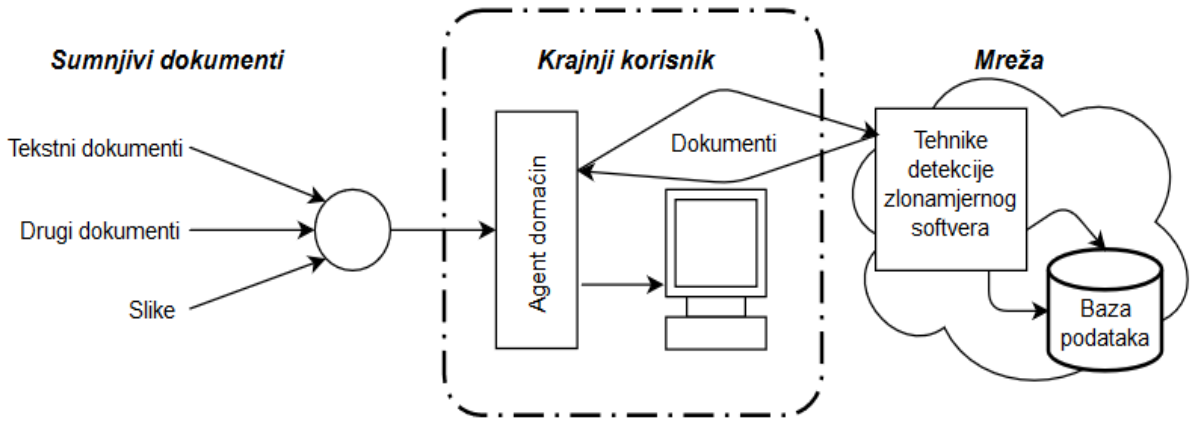
Autori su u [30] predstavili jezik za specificiranje ponašanja privilegiranih programa. Ova se tehnika oslanja na operativni sustav za stvaranje revizijskih zapisa koji se zatim koriste u provjeri ponašanja programa. Revizija je proces bilježenja "zanimljivih" aktivnosti, koji je u ovom slučaju svaki put kada se poziva sistemski poziv. Na temelju specifikacija programa, može se utvrditi da li je njegovo ponašanje zlonamjerno ili ne. Specifikacije programa prevode se u tragove revizije koji se uspoređuju s revizijskim tragovima programa koji se pregledava. Tragovi revizije programa koji se pregledava osmišljeni su u operativnom sustavu. Potencijalni nedostatak ove tehnike je taj da će se zlonamjerni softver otkriti nakon napada. Drugi mogući nedostatak je taj što tehnika može biti jednako značajna kao mehanizam revizije operativnog sustava.

Otkrivanje zlonamjernog koda u upravljačkom programu

Autori u [31] predlažu metodu u kojoj je meta zlonamjerni softveri za pokretanje koji inicilizira hardver i učitava operativni sustav. Ovo je područje ranjivosti jer se radi o kodu koji se izvršava prije učitavanja operativnog sustava. Modul neprovjerenog upravljačkog programa provjerava se u skladu s sigurnosnim pravilima prije nego što ga učita u memoriju. Općenito, ove sigurnosne politike identificiraju kako je dopušteno pokretačima uređaja da se međusobno povezuju s ostatkom sustava. Kompajler za ovjeru sastavlja i bilježi nepouzdane module upravljačkog programa. U tom istraživanju prevodilac je prihvatio Java bajt kod. Izlaz prevodioca predstavlja reprezentaciju koja provjerava verifikator kako bi se utvrdilo može li se vjerovati modulu upravljačkog softvera. Njihova se metoda temelji na učinkovitom certificiranju koda koje osigurava sljedeće: sigurnost protoka, sigurnost memorije i sigurnost snopa.

5.4. Detekcija u oblaku (Cloud-based detection)

Sigurnost u oblaku pružaju mnoge tvrtke za otkrivanje zlonamjernog softvera s vodećim stopama detekcije u industriji. Skeniranje visokih performansi otkrivaju najnovije zlonamjerne uređaje. *Cipher Cloud* je tvrtka koja pruža ovakve usluge. Osnovna infrastruktura svih ovih usluga je ista, [32]. Na slici 13 se može vidjeti osnovna infrastruktura detekcije zlonamjernog softver u oblaku.



Slika 13. Osnovna infrastruktura detekcije zlonamjernog softvera u oblaku

Izvor: [32]

Kao i kod *Fileprint* analize kod detekcije temeljene na anomaliji koriste se binarni n-gram kao značajka formalnog otkrivanja softvera. Međutim, budući da je ukupan broj mogućih n-grama nevjerojatno velik, odabire se n-gram koji imaju najveću diskriminacijsku snagu. Kako tok napreduje, pojavljuju se noviji n-grami koji dominiraju starijim n-gramima. Ovi noviji n-grami zamjenjuju stare u modelu kako bi se utvrdile najbolje značajke za određeno razdoblje, [32].

6. Značajke i postupci reverznog inženjeringa

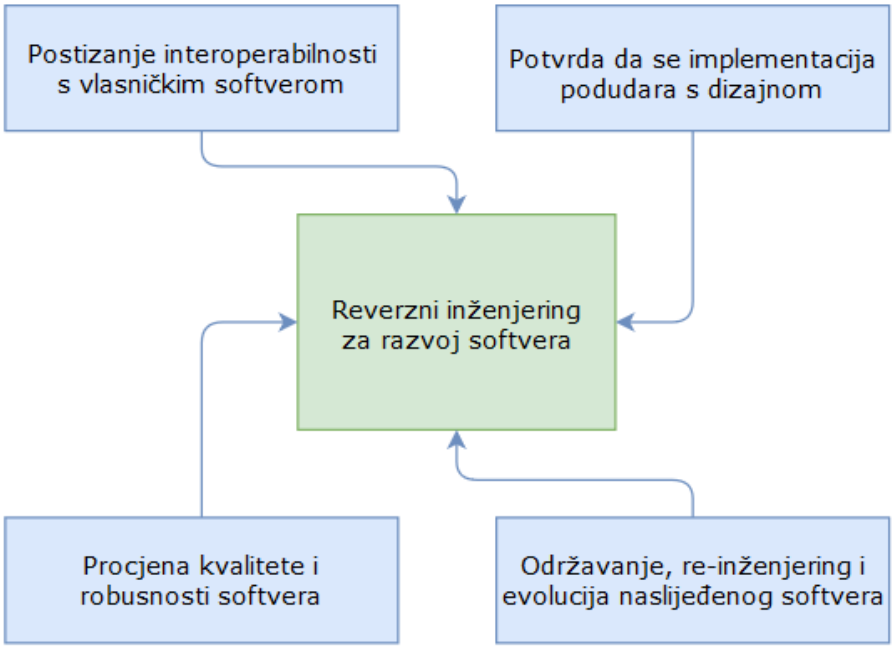
Reverzni inženjering je postupak analize softverskog sustava, u cjelini ili djelomično, kako bi se izvukli podaci o dizajnu i implementaciji. Tipičan scenarij reverznog inženjeringa uključivao bi softverski modul koji radi godinama i nosi nekoliko pravila poslovanja u svojim linijama koda. Reverzne inženjerske vještine također se koriste za otkrivanje i neutraliziranje virusa i zlonamjernog softvera i zaštitu intelektualnog vlasništva. Potrebni su računalni programeri koji su vješti u reverznom inženjeringu kako bi se softverske komponente mogle održavati, poboljšavati ili ponovo upotrebljavati.

Tijekom krize Y2K (kvar koji se odnosi na događaje povezane s oblikovanjem i pohranjivanjem kalendarskih podataka za datume koji počinju u 2000. godini) postalo je očito da reverzne inženjerske vještine nisu obično održane kod programera. Od tog vremena provodi se mnogo istraživanja kako bi se formaliziralo koje vrste aktivnosti spadaju u kategoriju reverznog inženjerstva, kako bi se tim vještinama mogli naučiti računalni programeri i testeri. Kako bi se riješili nedostatci u obrazovanju o reverznom inženjeringu, prikupljeno je nekoliko recenziranih članaka o reverznom inženjeringu, ponovnoj uporabi softvera, održavanju softvera, evoluciji softvera i sigurnosti softvera s ciljem razvoja odgovarajućih, praktičnih vježbi u svrhe podučavanja. Istraživanje je pokazalo da je prilično dobro opisan i sve povezane aktivnosti uglavnom spadaju u dvije kategorije: razvoj softvera i softver povezan sa sigurnošću, [33].

Iako se velik dio napisanog softvera više ne koristi, značajan iznos preživio je desetljećima i nastavlja voditi globalnu ekonomiju. Teško bi bilo ovih dana steći stručno obrazovanje za naslijeđene programske jezike kao što su COBOL, PL/I i FORTRAN. Složenu situaciju čini činjenica da je velik dio naslijeđenih kodova slabo dizajniran i dokumentiran. Kad god se računalni znanstvenici ili softverski inženjeri bave razvojem postojećeg sustava, 50–90% radnog napora troši se na razumijevanje programa. Nakon što inženjeri provode tako veliku količinu svog vremena pokušavajući razumjeti sustav prije poboljšavanja sustava to nije ekonomski održivo jer softverski sustav i dalje raste u veličini i složenosti. Iako već postoji nekoliko alata koji pomažu softverskim inženjerima u procesu razumijevanja programa, alati se usredotočuju na prijenos podataka o dizajnu softverskog sustava. Očekivano je da programer ima dovoljno vještina da učinkovito integrira informacije u svoj model arhitekture sustava, [33].

Prema [34], postoje četiri scenarija reverznog inženjeringa povezanih s razvojem softvera, te scenariji pokrivaju širok spektar aktivnosti koje uključuju: održavanje softvera, ponovnu upotrebu, re-inženjering, evoluciju, interoperabilnost i testiranje. Na slici 14 sažeti su reverzni inženjerski scenariji u vezi s razvojem softvera. Slijede zadaci koje bi se mogli obaviti u svakom od scenarija:

- Postizanje interoperabilnosti s vlasničkim softverom: razviti programe ili upravljačke programe koji interoperativno koriste vlasničke biblioteke u operacijskim sustavima ili aplikacijama.
- Provjera odgovara li implementacija dizajnu: provjerite da se kod proizveden tijekom razvojnog procesa podudara s predviđenim dizajnom pretvaranjem koda u apstraktni dizajn.
- Procjena kvalitete i robusnosti softvera: osiguranje kvalitete softvera prije kupnje vršenjem heurističke analize binarnih datoteka kako bi se provjerilo postoje li upute s nizom nekvalitetnih kodova.
- Održavanje, re-inženjering i evolucija naslijeđenog softvera: oporavak dizajna starih softverskih modula kad izvorni kod nije dostupan kako bi se omogućilo održavanje, evolucija i ponovna upotreba modula, [34].



Slika 14. Scenariji reverznog inženjeringa za razvoj softvera

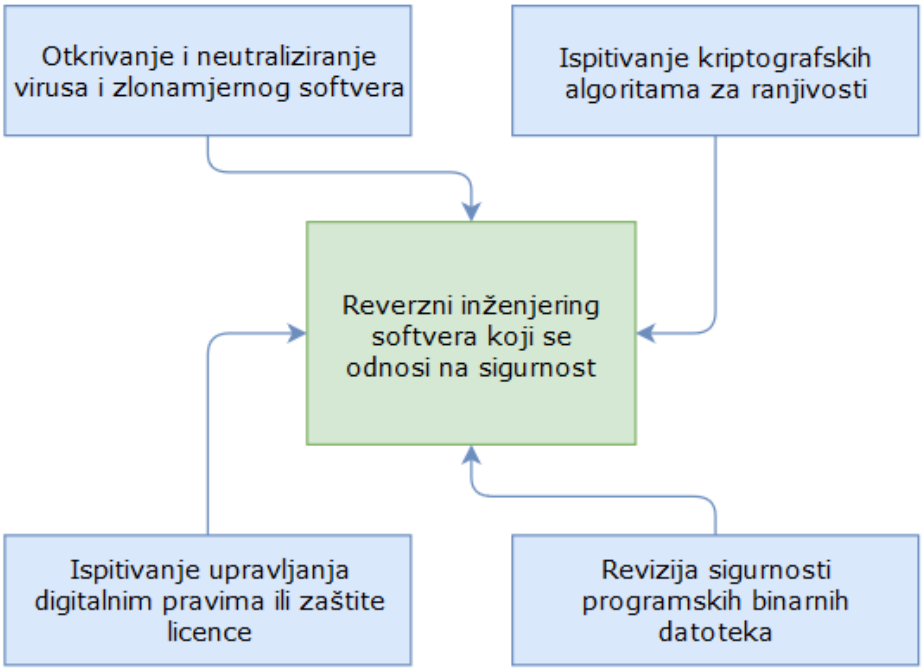
Izvor: [34]

Iz perspektive softverske tvrtke, vrlo je poželjno da su njeni proizvodi zaštićeni od neželjenih aktivnosti i da je na njima teško napraviti postupak reverznog inženjerstva. Napraviti softver tako da je na njemu teško napraviti postupak reverznog inženjerstva u sukobu je s idejom da bi se kasnije mogao oporaviti dizajn softvera radi održavanja i evolucije. Stoga proizvođači softvera obično ne primjenjuju tehnike reverznog inženjeringa na softver dok ne budu isporučeni kupcima, čuvajući kopije čitljivog i održivog koda. Proizvođači softvera obično ulažu vrijeme samo u otežavanje reverznog softvera ako postoje posebno zanimljivi algoritmi zbog kojih se proizvod izdvaja od konkurencije. Programski jezici kao što je Java, koji od računalnih programera ne zahtijevaju da upravljaju detaljima sustava na nižoj razini, postali su sveprisutni. Kao rezultat toga, računalni programeri sve su više izgubili dodir s onim što se događa u sistemskom izvođenju programa, [34].

U [34] sugerira se da programeri mogu steći bolje i dublje razumijevanje softvera i hardvera, kroz učenje koncepata reverznog inženjeringa. „Hakeri“ i kradljivci bili su prilično glasni i aktivni u dokazivanju da posjeduju dublje razumijevanje detalja o sustavu na niskoj razini od svojih profesionalnih kolega. Postoje četiri scenarija reverznog inženjeringa povezanih sa softverskom sigurnošću, baš poput reverznog inženjeringa vezanog za razvoj. Scenariji pokrivaju širok spektar aktivnosti koje uključuju osiguranje da je softver siguran za upotrebu, štiteći pametne algoritme ili poslovne procese, sprječavanje krađe softvera i digitalnih medija poput glazbe, filmova i knjiga i osiguravanjem kriptografskih algoritama koji nisu podložni napadima.

Na slici 15 sažeti su reverzni inženjerski scenariji u vezi sa softverskom sigurnošću. Zadaci koji bi se mogli obaviti u svakom od scenarija:

- Otkrivanje i neutraliziranje virusa i zlonamjernog softvera: otkrivanje, analiziranje ili neutraliziranje (čistog) zlonamjernog softvera, virusa i špijunskog softvera.
- Ispitivanje kriptografskih algoritama za ranjivosti: ispitivanje razine sigurnosti podataka koju pruža određeni kriptografski algoritam analizirajući je na slabosti.
- Testiranje upravljanja digitalnim pravima ili zaštita licence : zaštita digitalnih prava na softveru i medijima primjenom i testiranjem anti reverzibilnih tehnika.
- Revizija sigurnosti programskih binarnih datoteka: program revizije za sigurnosne ranjivosti bez pristupa izvornom kodu skeniranjem niza uputa radi potencijalnih iskorištavanja, [34].



Slika 15. Scenariji reverznog inženjeringa softvera koji se odnosi na sigurnost

Izvor: [34]

Reverzni inženjering zlonamjernog softvera uključuje rastavljanje (a ponekad i dekompajliranje) softverskog programa. Kroz ovaj se postupak binarne upute pretvaraju u kodne mnemonike (ili konstrukcije više razine) kako bi inženjeri mogli pogledati što program radi i na koje sustave utječe. Tek poznavanjem njegovih detalja, inženjeri će tada moći stvoriti rješenja koja mogu ublažiti željene štetne efekte programa. Reverzni inženjer upotrijebit će niz alata kako bi otkrio kako se program širi kroz sustav i što je projektirano. A čineći to, on bi tada znao koje ranjivosti program namjerava iskoristiti. Da bi poništili kod zlonamjernog softvera, inženjeri će često koristiti mnoge alate:

- Rastavljači (npr. IDA Pro). Rastavljač će rastaviti aplikaciju za izradu asemblerskog koda. Dekompajleri su također dostupni za pretvaranje binarnog koda u izvorni kod, iako nisu dostupni za sve arhitekture.
- Debugeri (npr. X64dbg, Windbg, GDB). Reverzni inženjeri koriste uređaje za uklanjanje pogrešaka da bi manipulirali izvršenjem programa kako bi stekli uvid u ono što radi tijekom izvođenja programa. Također omogućavaju inženjeru da kontrolira određene aspekte programa dok se izvodi, poput područja memorije programa. To omogućava veći uvid u program koji radi i kako utječe na sustav ili mrežu.

- Mrežni analizatori (npr. Wireshark). Mrežni analizatori govore inženjeru kako program komunicira s drugim uređajima, uključujući veze koje program uspostavlja i koje podatke pokušava poslati, [35].

Inženjerima je potrebno više vremena da shvate rastavljeni ili dekompajlirani kod. I to je vrijeme za koje zlonamjerni softver može „pustošiti“ na mreži. Zbog toga je sve veća pažnja posvećena dinamičkoj analizi zlonamjernog softvera. Dinamička analiza zlonamjernog softvera oslanja se na zatvoreni sustav kako bi se zlonamjerni program pokrenuo u sigurnom okruženju i kako bi se gledalo što on čini, [35].

Reverzni inženjering ima različite primjene, ali svakako se najviše koristi kod analize i detekcije potencijalnih zlonamjernih softvera pametnih telefona. Putem reverznog inženjeringa se može vidjeti izvorni kod potencijalno „zaražene“ aplikacije nekim virusom i programeri mogu prema kodu ili putem raznih alata vidjeti postoji li u aplikaciji nešto zlonamjerno što može korisniku naštetiti u bilo kakvom smislu. Zlonamjerni softver Anubis djelovao je između 2018. i 2019. godine i napravio je najviše problema korisnicima. U daljnjim navodima moguće je vidjeti reverzni inženjering zlonamjernog softvera Anubis.

Anubis se uglavnom sastoji od dva dijela: preuzimanjem i korisnim opterećenjem. Ako se zlonamjerni softver širi na web mjesta trećih strana, poput *flash* ažuriranja, on preuzima samo korisni teret Anubisa. Ali ako se zlonamjerni softver proširi Google Play trgovinom, on koristi program za preuzimanje. Anubis se koristi malim, ali snažnim koracima kako bi korisnici povjerovali da je to legitimna aplikacija. Budući da autori takvih softvera žele uhvatiti „vrijedne“ žrtve, obično će ove lažne aplikacije biti povezane s financijama. Anubis može pratiti korisničke aktivnosti i imati mogućnost ispitivanja određenih stvari, poput okvira za poruke. Svaki događaj pristupačnosti ima izvornu komponentu koja definira koja je aplikacija pokrenula trenutni događaj. Postoje različite vrste događaja koje Anubis koristi:

- TIP_VIEW_CLICKED
- TIP_VIEW_FOCUSED
- TIP_VIEW_TEXT_CHANGED
- TIP_WINDOW_STATE_CHANGED, [36].

Anubis prati sve događaje pristupačnosti i provjerava vrste događaja. Dakle, ako korisnik otvori novi prozor, stanje će se promijeniti, a događaj će se pokrenuti. Vrsta

događaja TYPE_WINDOW_STATE_CHANGED je prva provjera. Kako bi korisnik uklonio zlonamjerni softver, vjerojatno ide u postavke. Postavke su Android aplikacija pod nazivom *com.android.settings*. Druga je provjera dolazi li pokrenuti događaj iz *com.android.settings*. Prva provjera odnosi se na naziv aplikacije koju je korisnik kliknuo, a zatim se provjeravaju: *uninstall* (this.f) i *to remove* (this.g). Ako su svi uvjeti ispunjeni, pokreće se aktivnost a(). Ova aktivnost samo otvara okvir upozorenja koji kaže da se sistemske aplikacije ne mogu brisati. Budući da aplikacija nema sadržaj koji se može pokrenuti, Android otvara okvir upozorenja na početnom zaslonu. Dakle, kad god korisnik pokušava otvoriti detalje zlonamjernog softvera u postavkama, zlonamjerni ga softver proslijedi na početni zaslon s okvirom upozorenja i da se aplikacija ne može izbrisati. A ovo je samo jedan od mnogih problema koje zlonamjerni softver Anubis može uzrokovati korisniku, [36].

```
if (source != null) {
    for (AccessibilityNodeInfo accessibilityNodeInfo : source.findAccessibilityNodeInfosByText(this.a.i(this))) {
        for (AccessibilityNodeInfo accessibilityNodeInfo2 : source.findAccessibilityNodeInfosByText(this.f)) {
            if (accessibilityNodeInfo2.toString().contains("com.android.settings")) {
                a();
                cVar6 = this.a;
                stringBuilder4 = new StringBuilder();
                stringBuilder4.append("p=");
                cVar = this.a;
                stringBuilder5 = new StringBuilder();
                stringBuilder5.append(this.a.q(this));
                stringBuilder5.append("|Attempt to remove malware 2|");
                stringBuilder4.append(cVar.c(stringBuilder5.toString()));
                cVar6.b(this, "4", stringBuilder4.toString());
            }
        }
        for (AccessibilityNodeInfo accessibilityNodeInfo22 : source.findAccessibilityNodeInfosByText(this.g)) {
            if (accessibilityNodeInfo22.toString().contains("com.android.settings")) {
                a();
                cVar6 = this.a;
                stringBuilder4 = new StringBuilder();
                stringBuilder4.append("p=");
                cVar = this.a;
                stringBuilder5 = new StringBuilder();
                stringBuilder5.append(this.a.q(this));
                stringBuilder5.append("|Attempt to remove malware 3|");
                stringBuilder4.append(cVar.c(stringBuilder5.toString()));
                cVar6.b(this, "4", stringBuilder4.toString());
            }
```

Slika 16. Prikaz zlonamjernog koda Anubis zlonamjernog softvera

Izvor: [36]

Na slici 16 se može vidjeti primjer zlonamjernog koda Anubis zlonamjernog softvera. Napredni programi zlonamjernog softvera imaju skup alata koji koriste za nadmudrivanje zatvorenog sustava i izbjegavanje otkrivanja: mogu odgoditi svoje zlonamjerne aktivnosti, djelovati samo kad je korisnik aktivan, sakriti zlonamjerni kod na područjima gdje ga neće otkriti, zajedno s nizom drugih tehnika sakrivanja. To znači da se reverzni inženjeri ne mogu oslanjati samo na dinamičke tehnike nego uz njih moraju koristiti i neke druge raspoložive tehnike, a u isto vrijeme, reverzni inženjering svake nove prijetnje nekog zlonamjernog softvera nije realan.

7. Zaključak

Sve više i više raste trend da se neke aplikacije skrivaju, kradu dragocjene resurse i podatke s mobilnih uređaja koji su važan dio svakodnevnice i uvode korisnike u digitalni svijet. Sve je veća prijetnja jer se gotovo polovica svih zlonamjernih programa na mobilnoj platformi sastoji od skrivenih aplikacija. Zločinci sakrivajući ikone svojih aplikacija pokušavaju obmanuti korisnike i zato korisnici moraju poduzeti više koraka za pronalaženje i uklanjanje neželjenih aplikacija. Kako bi ostali neotkriveni, kriminalci se koriste i raznim tehnikama kako bi njihove aktivnosti izgledale što legitimnije. Iako se taktike prijetnji i dalje mijenjaju kriminalci se prilagođavaju i otkrivaju tehnike detekcije zlonamjernih programa, a korisnici mogu poduzeti nekoliko koraka kako bi ograničili svoju izloženost i rizik.

Iako neke zlonamjerne aplikacije to čine tijekom postupka provjere, čini se da većina napada dolazi iz društvenih medija, lažnih oglasa i drugih neslužbenih izvora aplikacija. Prije nego što korisnik preuzme nešto na svoj uređaj najbolje bi bilo napraviti kratko istraživanje o izvoru i razvojnom programeru jer su mnogi od njih označili druge korisnike. Sveobuhvatni sigurnosni softver na svim uređajima, bilo da se radi o računalima, tabletu ili pametnim telefonima, i dalje je snažna obrambena mjera zaštite podataka i privatnosti od raznih napada kriminalaca. Također je najbolje korištenje alata za praćenje ID-a kako bi korisnik bio svjestan promjena ili radnji koje su se dogodile na njegovom pametnom telefonu, a da nije bio ni svjestan tih promjena.

Uz naravno sveprisutne antivirusne aplikacije koje razne tvrtke nude sa dobrim značajkama i uz dobre cijene postoje i razne analize zlonamjernog softvera kao i razne metode detekcije. Kako su razvijanjem zlonamjerni programi sve više sofisticiraniji i otporniji na takve metode i analize, nije moguće koristiti samo jednu tehniku nego je najbolje detektirati zlonamjerni softver kombinacijom različitih tehnika. Uz takve kombinacije postotak uspješnosti detekcije raste i s time su uređaji u vlasništvu korisnika sve sigurniji. Zbog toga je potrebno vršiti bolju i efikasniju edukaciju korisnika kako se obraniti od napadača i raznih prijetnji te sve više koristiti provjerene aplikacije koje pomažu kod sigurnosti njihovih mobilnih uređaja. Također je potrebno razvijati nove alate za rano detektiranje prisutnosti zlonamjernog softvera kako bi se osigurala pravovremena reakcija na napad i kako bi se time spriječio nastanak štete na uređaju.

Literatura

[1] “McAfee Mobile Threat Report”, Preuzeto sa: 2020-Mobile-Threat-Report.pdf [Pristupljeno: kolovoz 2020.]

[2] „Zlonamjerni softver – CERT“, Preuzeto sa: https://www.cert.hr/19795-2/malver/ [Pristupljeno: kolovoz 2020.]

[3] „Adware – What Is It & How To Remove It | Malwarebytes“, Preuzeto sa: https://www.malwarebytes.com/adware/ [Pristupljeno: kolovoz 2020.]

[4] „What is crimeware“, Preuzeto sa: https://www.pcmag.com/encyclopedia/term/crimeware [Pristupljeno: kolovoz 2020.]

[5] „What is Scareware“, Preuzeto sa: https://www.forcepoint.com/cyber-edu/scareware [Pristupljeno: kolovoz 2020.]

[6] „What is a Keylogger? | How Hackers Install a Keylogger“, Preuzeto sa: https://enterprise.comodo.com/what-is-a-keylogger.php [Pristupljeno: kolovoz 2020.]

[7] „What is rootkit, and how to stop them | Norton“, Preuzeto sa: https://us.norton.com/internetsecurity-malware-what-is-a-rootkit-and-how-to-stop-them.html [Pristupljeno: kolovoz 2020.]

[8] „Everything You Need To Know About Android Virus“, Preuzeto sa: https://tunesgo.wondershare.com/android-tips/what-is-android-virus-and-how-to-remove-virus-android.html [Pristupljeno: kolovoz 2020.]

[9] „Spread and Control of Mobile Benign Worm“, Preuzeto sa: https://www.hindawi.com/journals/jam/2014/746803/ [Pristupljeno: kolovoz 2020.]

[10] „What is Trojan Virus?“, Preuzeto sa: https://www.kaspersky.com/resource-center/threats/trojans [Pristupljeno: kolovoz 2020.]

[11] Bickford J., O'Hare R., Baliga A., Ganapathy V., Iftode L.: *Rootkits on Smart Phones: Attacks, Implications and Opportunities*, Department of Computer Science, Rutgers University, 2010.

[12] „What is keylogger“, Preuzeto sa: https://www.malwarebytes.com/keylogger/ [Pristupljeno: kolovoz 2020.]

[13] „Norton 360“, Preuzeto sa: https://uk.norton.com/products?inid=hho_supp_header_products [Pristupljeno: kolovoz 2020.]

[14] „Security products for a post-perimeter world“, Preuzeto sa: https://www.lookout.com/products [Pristupljeno: kolovoz 2020.]

[15] „Avast Store“, Preuzeto sa: https://www.avast.com/store#all [Pristupljeno: kolovoz 2020.]
[16] „McAfee Mobile Security“, Preuzeto sa: https://www.mcafee.com/enterprise/en-us/solutions/mvision-endpoint-security.html [Pristupljeno: kolovoz 2020.]
[17] „Android Application Malware Analysis“, Preuzeto sa: https://hacken.io/research/industry-news-and-insights/android-application-malware-analysis/ [Pristupljeno: kolovoz 2020.]
[18] Kang H., Jang J.W., Mohaisen A., Kim H.K.: *Detecting and classifying android malwareusing static analysis along with creator information*, Int. J, Distrib, Sens, Netw.11(6),479174, 2015.
[19] Song J., Han C., Wang K., Zhao J., Ranjan R., Wang L.: *An integrated static detectionand analysis framework for Android*, Pervasive Mob, Comput.32,15–25, 2016.
[20] „Androguard GUI“, Preuzeto sa: https://androguard.readthedocs.io/en/latest/tools/androgui.html [Pristupljeno: rujan 2020.]
[21] „JD-GUI for Windows and Mac“, Preuzeto sa: https://mac.softpedia.com/get/Development/Java/JD-GUI.shtml [Pristupljeno: rujan 2020.]
[22] Vidal J.M., Monge M.A.S., Villalba L.J.G.: *A novel pattern recognition system for detecting Android malware by analyzing suspicious boot sequences*, Knowl., Based Syst.150, 198–217, 2018.
[23] „Android Studio and SDK Tools“, Preuzeto sa: https://developer.android.com/studio [Pristupljeno: rujan 2020.]
[24] „Working with Wireshark“, Preuzeto sa: https://www.networkcomputing.com/networking/working-ring-buffer-wireshark-files [Pristupljeno: rujan 2020.]
[25] Idika N., Mathur A. P.: *A Survey of Malware Detection Techniques*, Department of Computer Science, Purdue University, West Lafayette, 2007.
[26] Ellis D., Aiken J., Attwood K., Tenaglia S.: *A behavioral approach to worm detection*, 2004 ACM Workshop on Rapid Malcode, str. 43–53, 2004.
[27] Sung A., Xu J., Chavez P., Mukkamala S.: *Static analyzer of vicious executa-bles*, 20th Annual Computer Security ApplicationsConference (ACSAC ’04), 00:326–334, 2004.
[28] Wang K., Stolfo S. J.: *Anomalous payload-based network intrusion detection*, 7th International Symposium on (RAID), str. 201–222, 2004.

[29] W. Li, K. Wang, S. Stolfo, and B. Herzog. Fileprints: *Identifying file types by n-gramanalysis*, 6th IEEE Information Assurance Workshop, June 2005.
[30] Ko C., Fink G., Levitt K.: *Automated detection of vulnerabilities in privilegedprograms by execution monitoring*, 10th Annual ComputerSecurity Applications Conference, str. 134–144, 1994.
[31] Adelstein F., Stillerman M., Kozen D.: *Malicious code detection for open firmware*, 18th Annual Computer Security Applications Conference, 2002.
[32] Gupta K. M., Shaw S., Chakraborty S.: *Pattern Based Malware Detection Technique in Cloud Architecture*, Institute of engineering & Management, Kolkata, India, 2016.
[33] Ali M.R.: *Why teach reverse engineering?*, ACM SIGSOFT SEN 30(4), 1-4, 2005.
[34] Eliam E.: *Secrets of Reverse Engineering*, Wiley, Indianapolis, 2005.
[35] Nugroho H. A., Hamid: *Reverse Engineering Technique fo Malware Analysis*, Islamic University of Indonesia, Yogykarta, 2013.
[36] „Mobile Malware Analysis: Tricks used in Anubis“, Preuzeto sa: https://eybisi.run/Mobile-Malware-Analysis-Tricks-used-in-Anubis/ [Pristupljeno: rujan 2020.]

Popis kratica

TCP (Transmission Control Protocol) protokol za kontrolu prijenosa podataka

DNS (Domain Name System) protokol za davanje imena mrežnim adresama

UDP (User Datagram Protocol) protokol koji se nalazi u dijelu transportne razine referentnog OSI modela

PDF (Portable Document Format) format zapisa dokumenata

API (Application Programming Interface) aplikacijsko programsko sučelje

Popis slika

Popis tablica