SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4588

ZAŠTITA OD UCJENJIVAČKOG ZLOĆUDNOG KODA SPREČAVANJEM PREPISIVANJA PODATAKA NA DATOTEČNOM SUSTAVU

Tibor Žukina

Zagreb, lipanj 2016.

Sadržaj

1.Uvod

Zloćudni kod (engl. malware) [1] tip je računalnog programa koji djeluje na korisnikovom računalu bez njegovog pristanka te ima neku vrstu nepoželjnog ili štetnog učinka na sustav, kao što je oštećenje programa i podataka koji se nalaze na sustavu, krađa podataka, prikazivanje reklamnih poruka, sudjelovanje u napadima na druga računala putem mreže ili omogućavanje neovlaštenog udaljenog pristupa na računalo. Posebno opasna i česta vrsta takvih programa koja ju u posljednjih nekoliko godina doživjela porast i protiv koje se neprestano razvijaju različite nadogradnje na sigurnosnim sustavima mehanizmi zaštite naziva se ucjenjivački zloćudni kod (engl. ransomware). Ucjenjivački zloćudni kod tip je zloćudnog koda posebno namijenjenog blokiranju ili kriptiranju podataka nakon kojeg slijedi iznuda određenog iznosa za otkupljivanje jedinstvenog ključa. Napad ucjenjivačkog zloćudnog koda obično počinje kad je PC napadnut od iznuđivača koristeći različite tehnike društvenog inženjeringa ili napada na sustav. Nakon kriptiranja žrtva ima izbor hoće li platiti iznuđivačima kako bi spasila svoje kriptirane podatke. Obično se prikaže poruka da će pokušaj onemogućavanja ili usporavanja funkcionalnosti ucjenjivačkog zloćudnog koda na bilo koji način odmah dovesti do gubitka svih podataka.

Motivacija za izradu ovog završnog rada upravo je obrana žrtava zloćudnog koda koji zbog svojeg neiskustva ili neznanja najčešće budu prisiljeni platiti značajnu otkupninu. Njihovi su podaci, koji mogu biti vrlo važni za poslovni ili privatni život tada ozbiljno ugroženi. Cilj je postići imunost na ucjenjivače, odnosno omogućiti im da na bilo koji drugi način vrate kontrolu nad svojim sustavom i oporave svoje podatke ne morajući pri tome pristati na njihove uvjete. Pri tome je veliki izazov predvidjeti sve moguće mehanizme kojima ucjenjivački program želi održati kontrolu nad sustavom žrtve te je potrebno detaljno poznavati mnoge funkcije i svojstva tog sustava kako bi se to moglo izvesti. Razvoj takve zaštite nije jednokratni posao koji nakon svoga završetka čini sustave korisnika savršeno otpornima na bilo koji pokušaj napada takvog zloćudnog programa. To je kontinuirani, dugotrajni i postupni proces koji tijekom svojeg rada nailazi na propuste nastale zbog nedostataka u prethodim zaštitama ili unaprijeđenju samog zloćudnog koda. Te se propuste nastoji ispraviti u što kraćem roku i osigurati pritom što pouzdaniju zaštitu.

Tema ovog završnog rada je program CryptoLocker, način njegova rada, šteta koju uzrokuje i načini zaštite koji podrazumijevaju poništavanje ili sprječavanje njegova učinka. CryptoLocker je zloćudni ucjenjivački kod koji napada računala s operacijskim sustavom Windows te je jedan od najopasnijih i najučinkovitijih programa toga tipa koji je svojim autorima donio veliku zaradu ugrozivši veliku količinu podataka na računalima [9]. Obično se širi mrežom postojećih botneta koji šalju poruke elektroničke pošte velikom broju nasumično odabranih korisnika. Poruka elektroničke pošte koju naivni korisnik otvori naizgled je bezopasnog sadržaja, no zapravo je zaražena CryptoLocker izvršnom datotekom. Čim se pokrene, zloćudni se program učita u radnu memoriju i izbriše s diska te je nakon toga nemoguće naći njegov binarni kod. Način djelovanja i svrhu većine zloćudnog koda moguće je analizirati tehnikama reverznog inženjeringa kojima se ispituje njihovo ponašanje, no time što se izbriše Crytolocker pokušava otežati pristup svome izvršnome kodu, a time i otkrivanju konkretnog načina na koji radi. Korištenjem alata za snimanje mrežnog prometa te nadzorom aktivnih procesa i memorije računala, detalji trenutne verzije CryptoLockera ipak nakon nekog vremena budu otkriveni, no u opticaj uvijek bude puštena nova verzija s brojnih poboljšanjima koja je ponovno korak ispred svojih žrtava i stručnjaka informacijske sigurnosti kojima je glavna zadaća preventivno djelovati protiv te vrste sigurnosnih prijetnji.

Rad je strukturiran na sljedeći način: Prvo poglavlje završnog rada opisuje okolnosti u kojima će se ispitivati način rada CryptoLockera i pokušati razviti rješenje za različite scenarije njegova napada. U njemu se uvode osnovni pojmovi potrebni za razumijevanje postavljenog zadatka i odgovarajućeg rješenja. Potom se u drugom poglavlju objašnjavaju mehanizmi CryptoLockera u svim njegovim fazama, što uključuje njegovo širenje, prikrivanje djelovanja, kriptiranje podataka, komunikaciju s poslužiteljem te plaćanje i dekriptiranje kriptiranih datoteka. Na kraju se u trećem poglavlju opisuje osnovna zaštita od CryptoLockera te navode mogući scenariji napada s odgovarajućim poboljšanjima zaštite od njih, korištenje različitih algoritama za kriptiranje, pisanje samo po određenim blokovima podataka i višestruko pisanje po istim podacima. Rad završava zaključkom i literaturom.

2.Virtualna okolina za ispitivanje ucjenjivačkog programa

Sigurno ispitivanje CryptoLockera podrazumijeva otkrivanje njegovih svojstava sa smanjenim ili gotovo nikakvim rizikom gubitka dokumenata za ispitivača na njegovom radnom računalu. To je moguće postići koristeći izolirano virtualizacijsko okruženje. Postoji nekoliko različitih programskih paketa koji omogućuju implementaciju virtualizacijskog okruženja, primjerice VMware [21], Oracle VM VirtualBox [19], XenServer [22], Parallels Desktop [23] , QEMU [24] i Virtual PC [25] , a za ispitivanje Cryptolockera u ovom se slučaju koristi Oracle VM VirtualBox. Osim standardnih alata za ispitivanje instaliranih unutar samog virtualnog stroja te na računalu domaćinu koristit će se dodatne napredne mogućnosti koje okolina za virtualizaciju nudi, a koje su nemoguće ili puno teže izvediv direktno na fizičkom računalu. Korištenjem virtualizacijskog rješenja trenutno stanje sustava prije i nakon kriptiranja datoteka i između svih ostalih bitnih međukoraka bit će moguće spremiti te nakon toga prema potrebi vratiti. Prije početka samog ispitivanja bilo je potrebno pribaviti primjerke programa CryptoLockera koji će se ispitivati te razumjeti koncepte virtualizacije i snimki stanja sustava koji će se pritom koristiti.

2.1. Virtualizacija računala

Sustav samog virtualnog stroja unutar kojeg se vrši ispitivanje izoliran je unutar posebne datoteke nad kojom CryptoLocker djeluje. Postoje različite vrste virtualnih strojeva, svaka sa drugačijim funkcijama.

Procesni virtualni strojevi (engl. *process virtual machines*) [26] namijenjeni su izvršavanju samog računalnog programa pružajući apstraktnu okolinu za izvođenje programa potpuno neovisnu o platformi. Neki virtualni strojevi također su namijenjeni emuliranju različitih arhitektura te dopuštaju izvođenje aplikacija i operacijskih sustava za drugačiji CPU i arhitekturu. Primjer rješenja koje koristi ovu vrstu virtualizacije je Java virtualni stroj (engl. *Java virtual machine*) koji omogućuje pokretanje Java aplikacija na operacijskom sustavu računala koje su vidljive kao zasebni procesi.

Sustavski virtualni strojevi (engl. *system virtual machines*) [26] predstavljaju potpunu zamjenu za određeni stvarni stroj te pružaju nivo funkcionalnosti potreban za

izvođenje potpunog operacijskog sustava. Virtualizacija na razini operacijskih sustava dopušta raspodjelu resursa računala pomoću funkcija jezgre operacijskog sustava za više korisničkih instanci koje se obično zovu kontejneri te se krajnjim korisnicima doimaju poput pravih strojeva. Za rad sustavskih virtualnih strojeva nužan je hipervizor koji dijeli i upravlja sklopovljem dopuštajući da se više različitih okruženja izoliranih jedni od drugih izvodi na istom fizičkom stroju. Moderni hipervizori koriste sklopovski potpomognutu virtualizaciju koja omogućuje efikasnu i potpunu virtualizaciju pomoću specifičnih sklopovskih mogućnosti, prvenstveno CPU-a fizičkog računala. Primjer rješenja koje koristi ovakvu virtualizaciju su virtualna računala stvorena programskim paketom Oracle VM VirtualBox [19] na kojima su instalirani izolirani operacijski sustavi.

CryptoLocker djeluje na sve korisničke datoteke određenih sufiksa unutar datotečnog sustava i za rad mu potreban cjeloviti operacijski sustav unutar kojeg će se pokrenuti. Zato je za ispitivanje rada CryptoLockera i svih njegovih učinaka nužna potpuna virtualizacija na razini operacijskog sustava koju omogućuje sustavski virtualni stroj.

Sustavski virtualni strojevi kao takvi donose određene prednosti koje će se prilikom ispitivanja nastojati maksimizirati i nedostatke koje će se što više nastojati izbjeći.

Prednosti sustavskih virtualnih strojeva:

1) Više okolina operacijskih sustava mogu istovremeno postojati na istom tvrdom disku, s virtualnim particijama koje omogućuje dijeljenje datoteka stvorenih bilo na fizičkom računalu na kojeg se virtualni strojevi instaliraju, bilo na virtualnim računalima. Stoga instalacije programa, internetska povezivost i funkcije poput ispisa i faksiranja mogu biti generirane bilo na fizičkom, bilo na virtualnom računalu. Bez obzira na sustav svojeg nastanka, sve su datoteke spremljene na tvrdom disku fizičkog računala. Ova će funkcionalnost biti poželjna pri snimanju internetskog prometa virtualnog stroja alatom Wireshark koji će biti instaliran na operacijskom sustavu fizičkog računala.
2) Mogu pružati emulirane sklopovske okoline različite od procesorskog seta instrukcija fizičkog računala. Ova prednost nije potrebna u konkretnom slučaju budući da virtualno računalo jednostavno uzima dio procesorskih i memorijskih

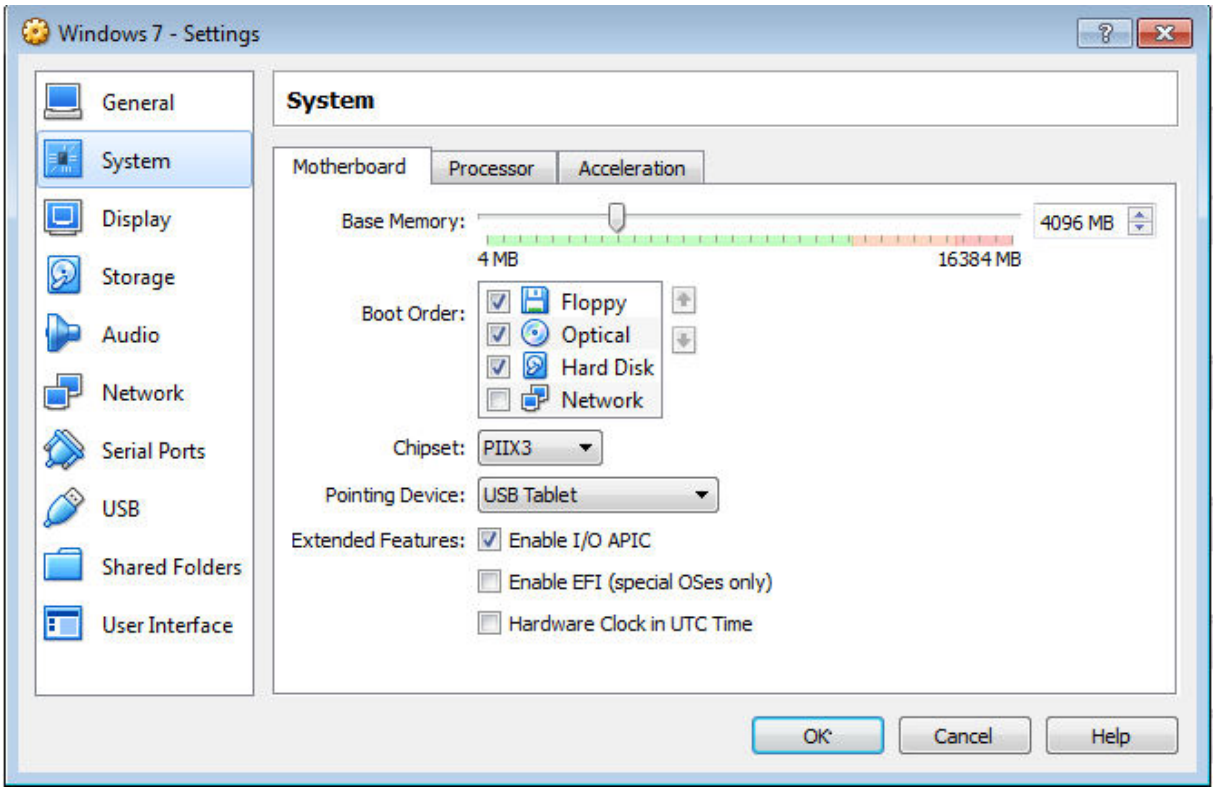
resursa fizičkog računala, pri čemu je na njemu pokrenut isti operacijski sustav.

Glavni nedostatci sustavskih virtualnih strojeva:

1) Virtualni je stroj manje efikasan od pravoga stroja kad indirektno pristupa tvrdom disku fizičkog računala. Ovaj nedostatak neće biti izražen tijekom ispitivanja jer je ionako cilj da CryptoLocker pristup isključivo datotekama izoliranog virtualnog stroja na kojem je pokrenut, a njegovo izvođenje nije sklopovski zahtjevno što se tiče brzina diska, količine radne memorije i sl.
2) Dok su virtualni strojevi paralelno pokrenuti na tvrdom disku stvarnog domaćina, oni mogu ispoljavati promjenjive odnosno nestabilne performanse, primjerice u brzini izvođenja i zaštiti od zloćudnog koda. To ovisi o podacima drugih virtualnih strojeva kojima je sustava opterećen, osim ako odabrani softvare virtualnih strojeva različitim mehanizmima ne omogućuje potpunu izolaciju među tim virtualnim strojevima. Budući da u ovom konkretnom slučaju neće biti više virtualnih strojeva, neće biti konflikta između njih pa stoga ovaj nedostatak može biti zanemaren.
3) Zaštita od zloćudnih programa instalirana na fizičkom računalu najčešće ne dopire do operacijskog sustava i procesa samog virtualnog računala. Ovaj bi se navodni nedostatak u konkretnoj namjeni virtualnog stroja mogao pokazati kao prednost jer cilj je dopustiti CryptoLockeru da izvrši sve svoje uobičajene radnje do kraja kako bi se one u detalje mogle proučavati. Budući da je u konkretnom slučaju operacijski sustav virtualnog računala Windows 7 Professional isti kao i sustav fizičkog računala, program za zaštitu od zloćudnog koda mogao bi nastojati spriječiti nepoželjni internetski promet, zbog čega će ta njegova značajka morati biti isključena kako bi se CryptoLocker odvio do kraja, inače neće biti moguće poslati jedinstveni privatni ključ na poslužitelj, a time niti pokrenuti kriptiranje.

2.2. Postavljanje ispitne okoline

Za stvaranje virtualnog stroja na kojem će CryptoLocker biti pokrenut korišten je besplatni program Oracle VM VirtualBox [19] namijenjen upravo stvaranju virtualnih strojeva s velikim izborom operacijskih sustava koji se na njemu mogu pokretati, primjerice Windows, Linux, MacOS, Solaris te širokim mogućnostima dodjele računalnih resursa tome stroju. Za ovaj je zadatak odabran 64-bitni Windows 7 Professional. Budući da će na njemu biti instalirani alati koji uobičajeno postoje na Windows operacijskim sustavima te jednostavni analitički alati tipični za problemsko područje kojima se ovaj rad bavi, pretpostavlja se da će sustav zahtijevati relativno malo računalnih resursa, no budući da fizički stroj nudi radnu memoriju od 16 GB, opravdano mu je dati 4 GB radne memorije. Prilikom stvaranja virtualnog stroja ujedno se stvara i virtualni tvrdi disk za taj virtualni stroj, koji može biti u različitim formatima, a u ovom je slučaju dovoljan pretpostavljeni format .vdi (VirtualBox Disk Image) koji će kasnije biti reprezentacija cijelog sadržaja diska virtualnog stroja, uključujući i sistemske i korisničke i dijelove. Za ovaj će virtualni stroj biti odabran dinamički alocirani disk koji će koristiti prostor na fizičkom disku rastući kako se puni, iako se neće automatski smanjiti nakon što je dio prostora oslobođen. Razlog takvog odabira je što sustav neće zauzimati puno sekundarne memorije niti zahtijevati visoke performanse, pri čemu će tome disku biti dodijeljeno pretpostavljenih 25 GB prostora. Nakon što je virtualni stroj stvoren potrebno je na njemu podesiti određene postavke (Slika 2.1.) što uključuje opće postavke, postavke sustava, ekrana, sekundarne memorije, zvuka, mreže, USB-a, korisničkog sučelja i dijeljenih mapa. Najvažnije postavke koje su preduvjet za instalaciju i pokretanje operacijskog sustava su postavke spremišta (Storage) u kojima se virtualnom stroju pridružuje .iso file. U ovome slučaju to je .iso file koji sadrži instalaciju WIndowsa. Ukoliko se odgovarajući .iso file ili DVD s operacijskim sustav ne doda, a prije toga se pokrene virtualni stroj, traži se odabir start up diska. Ako se pritom ne odabere odgovarajući disk ili .iso file koji sadrži instalaciju Windowsa, bit će ispisana poruka "Fatal: No bootable medium found! System halted!". Ukoliko se doda odgovarajući .iso file nakon pokretanja će se instalirati sve bitne značajke operacijskog sustava Windows nakon čega može početi instalacija korisničkih aplikacija.



Slika 2.1. Definiranje parametara sklopovlja novog virtualnog računala

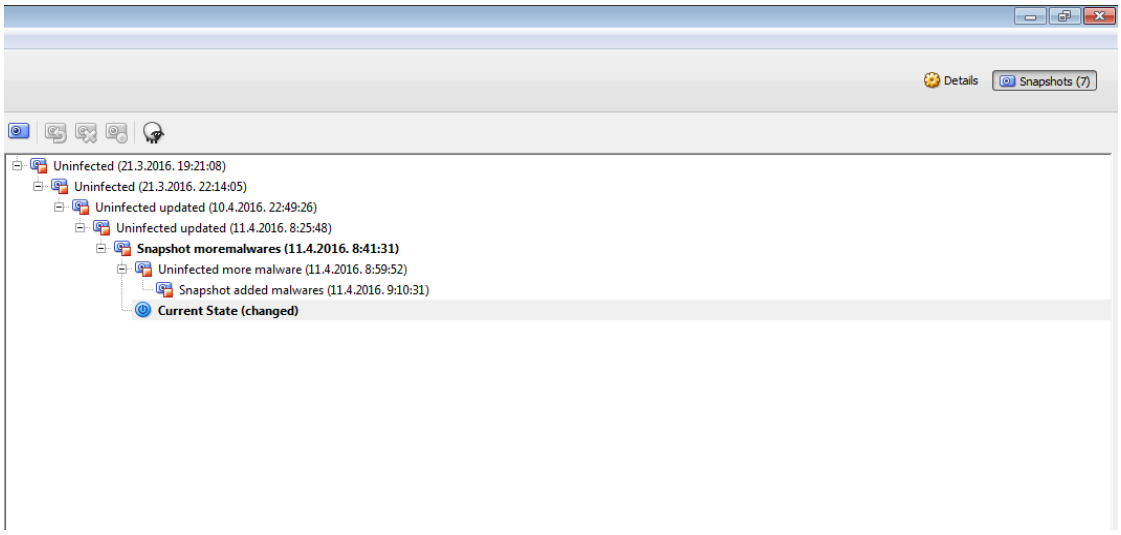
2.3. Izvori CryptoLockera za ispitivanje

Za potrebe ispitivanja rada CryptoLockera bilo je potrebno naći što noviju verziju tog malicioznog programa kako bi se uvjeti njegovog rada što vjernije simulirali i ispitali. Na internetu postoje mnoge baze zlođudnog koda namijenjene upravo toj svrsi, a nekolicina korisnika koji su uspjeli izolirati CryptoLockerovu izvršnu datoteku dobivenu elektroničkom poštom na forumima poput Reddita [14] postavili su linkove na repozitorije tih programa, uz upozorenje da se radi o stvarnim zlođudnim kodovima s kojima treba biti posebno oprezan, ne pokretati ih na stvarnome sustavu s bitnim podacima niti ih distribuirati ili koristiti za bilo kakve loše namjere. Nalaženje realističnog i pouzdanog CryptoLockera koji će omogućiti ispitivanje i otkrivanje svih njegovih značajki ipak nije bilo jednostavno iz više razloga. Kao prvo, mnogi nekad funkcionalni linkovi koji su uistinu sadržavali zip arhive CryptoLockera ili ucjenjivačkih zlođudnih kodova sličnih CryptoLockeru zbog sigurnosnih su razloga onemogućeni ili izbrisani od administratora foruma ili samog korisnika koji ih je postavio. Također postoje mnoge verzije CryptoLockera stare više od tri godine koje ne bi odražavale

stvarne principe njegovog rada. Novije su verzije izmijenjene u bitnim pojedinostima koje im omogućuju izbjegavanje zaštite osmišljene za napade starijih verzija. Kao treće, čak i noviji ucjenjivački zlćudni kod pogodan za istraživanje ove problematike za svoj rad, odnosno povezivanje sa poslužiteljima koristi php skripte na pojedinim inficiranim stranicama koje su u međuvremenu uklonjene pa program kao takav više ne može dobiti odgovor od poslužitelja i tako postaje nefunkcionalan. Najnovija pronađena verzija CryptoLockera na kojoj će se testirati većina njegovih značajki je Angler-EK-malware-payload-CryptoWall iz 2015. godine [3] pronađen kao zip arhiva referencirana na Redit forumu. Nešto starija verziju s kraja 2014. godine pronađena je unutar slobodno dostupno GitHub repozitorija s kraja te godine [2].

2.4. Snimka stanja operacijskog sustava

Snimka stanja operacijskog sustava (engl. *snapshot*) [4] pohranjuje stanje cijelog virtualnog računala i svega što se na njemu događalo u tom trenutku. Neke vIrtualizacijske okoline koje sadrže izolirani operacijski sustav unutar virtualnog stroja mogu stvarati snimka stanja cijelog operacijskog sustava spremajući cijelo stanje računala u datoteku i preusmjeravajući pisanje gosta u drugu datoteku, koja se onda ponaša kao *copy-on-write* tablica [5]. *Copy-On-Write* je strategija optimizacije korištena u operacijskim sustavima u kojoj svi dobiju pokazivače na iste resurse, a kad pozivatelj pokuša modificirati resurs, stvara se privatna kopija, kako se promjene ne bi odrazile na ostale pozivatelje. Tijekom ispitivanja CryptoLockera ovaj mehanizam koji korištena virtualizacijska okolina VirtualBox omogućuje (Slika 2.2.) bit će nužan kako promjene koje on izazove ne bi bilo nepovratne. U tom bi slučaju bilo potrebno ponovno instalirati cijeli operacijski sustav, platiti traženi iznos ili u najboljem slučaju obrisati sve zahvaćene datoteke te ih zamijeniti pričuvnom kopijom s drugog medija ukoliko ona uopće postoji. Na sučelju za upravljanje snikama stanja virtualnog računala vidljive su sve snimke stanja od njegova stvaranja, naznačeno je trenutno stanje sustava te je za svaku pojedinu snimku stanja sustava vidljivo od koje prijašnje snimke stanja je stvorena.



Slika 2.2. Snimke stanje sustava u okolini VirtualBox

Za rad sa snimkama stanja sustava obično se koriste tri osnovne operacije koje to sučelje također omogućuje. Prva operacija najčešće nazvana *Take snapshot* omogućuje spremanje trenutnog stanja sustava neovisno o svim budućim izmjenama. Druga operacija čije je najčešće ime *Restore snapshot* služi za obnovu stanja spremljenog unutar snimke stanja sustava, odnosno ponovno pridjeljivanje tog stanja virtualnom stroju. Treća operacija imena *Delete snapshot* omogućuje trajnje brisanje pojedinog spremljenog stanja sustava i nakon te operacije to konkretno stanje nikad neće moći biti obnovljeno. Nakon što su podešene sve sklopovske i programske postavke virtualnog stroja te su na njega instalirane sve bitne korisničke aplikacije i alati potrebni za ispitivanje ove problematike, WinRAR [15] i Sysintenals [16], s Interneta su skinute zip arhive najnovijih verzija CryptoLockera [2] [3].Prije njihovog pokretanja bilo je potrebno izraditi snimku stanja sustava virutualnog stroja kako bi se mogli jednostavno poništiti svi nepovoljni učinci koje CryptoLocker izazove i vratiti sustav u najnovije stanje prije njegovog pokretanja operacijom *Restore snapshot*. No prije nego što se to učini poželjno je spremiti snimku stanja zaraženog operacijskog sustava virtualnog stroja operacijom *Take snapshot* kako bi se lako mogli vratiti na to stanje i nastaviti ispitivanje bez da se prije toga opet mora pokrenuti CryptoLocker.

3. Analiza rada zloćudnog koda CryptoLocker

Kako bi se mogla razviti efikasna i funkcionalna zaštita od nekog zloćudnog programa, u ovome slučaju CryptoLockera, naprije je potrebno detaljno ispitati i razumjeti sva načela i mehanizme njegova rada. To podrazumijeva mehanizme širenja, odnosno načine na koje CryptoLocker dospijeva do računala žrtve pri čemu dolazi do infekcije. Nužno je i razumijevanje načina na koje CryptoLocker prikriva svoje djelovanje jer to omogućuje pronalaženje pogodnih alata za ispitivanje kojima će ono ipak biti vidljivo. Kako bi se u potpunosti istražio sam rad CryptoLockera i u skladu s time osmislilo rješenje koje će ga u njemu spriječiti, treba proučiti pojedinosti njegove komunikacije s poslužiteljem, pronalaženja, evidentiranja i kriptiranja bitnih datoteka te omogućavanja plaćanja nakon kojeg slijedi dekriptiranje podataka.

3.1. Mehanizmi širenja

Kao i velika većina ucjenjivačkih zloćudnih kodova, Crytolocker se prenosi kao dio instalacijskih datoteka lažnih nadogradnji programa koje se predstavljaju kao nadogradnje za Adobe Acrobat, Java i Flash Player [11]. Na stranicama poput Torrentza [17] koje služe za dijeljenje različitih piratskih programa često se može naići na oglase koji se koriste za distribuciju zloćudnog koda. Ti oglasi ne pripadaju operacijskom sustavu, nego su lažirani unutar web preglednika te obično tvrde da je potrebno nadograditi Adobe Acrobat.

Drugi način prijenosa CryptoLockera i zloćudnog koda općenito je na različitim neprovjerenim web stranicama u obliku bannera koji naizgled vodi na drugu web lokaciju ili nudi zanimljive dodatne sadržaje, popuste ili ponude, a zapravo posredno ili neposredno sadrži vezu za skidanje zloćudnog koda koji će se automatski pokrenuti i kojem je ponekad dovoljno samo dopuštenje lakovjernog korisnika za pokretanjem aplikacija iz neprovjerenog izvora.

Ucjenjivački zloćudni kod također je moguće preuzeti putem phishing poruka elektroničke pošte, iako oni najčešće završe u mapi za neželjenu poštu. Takve vrste poruka elektroničke pošte imaju u sebi lažne linkove koji će naivna žrtva phishinga kliknuti i koji će ju dovesti do skidanja CryptoLockera. Kako bi njihov identitet ostao

skriven, pošiljatelji takvih poruka ne koriste standardne pružatelje usluga elektroničke pošte, nego to čine preko slabo kontroliranih poslužitelja kakvi se najčešće nude u pojedinim azijskim zemljama. Kriminalci poruke ne šalju osobno i ciljano, nego se to obavlja automatski i masovno, a mnogi od njih to rade putem botneta [12], odnosno mreže velikog broja međusobno povezanih zaraženih računala na Internetu. Pritom korisnici tih računa ili uređaja nisu svjesni za što se njihova računala koriste, a identitetu stvarnog napadača jako je teško ući u trag.

Neke druge vrste zloćudnog koda, najčešće trojanci imaju svrhu oslabiti sigurnost žrtvinog računala i privući, odnosno skinuti na njega druge zloćudne programe, među kojima se nalaze i ucjenjivački zloćudni kodovi kao CryptoLocker ili različiti programi nastali po uzoru na njega.

Vjerojatnost infekcije CryptoLockerom može se smanjiti posjećivanjem isključivo legitimnih stranica za skidanje nadogradnji programa, izbjegavanjem neprovjerenih stranica s piratskim programima te ignoriranjem phishing poruka elektroničke pošte, bilo kakvih drugi linkova i bannera koji potencijalno vode do dohvata zloćudnog koda. Isto vrijedi i za prevenciju infekcije drugim zloćudnim kodovima koji su sami po sebi štetni, a mogu olakšati i prodor CryptoLockera u računalo.

3.2. Prikrivanje djelovanja i načini ispitivanja

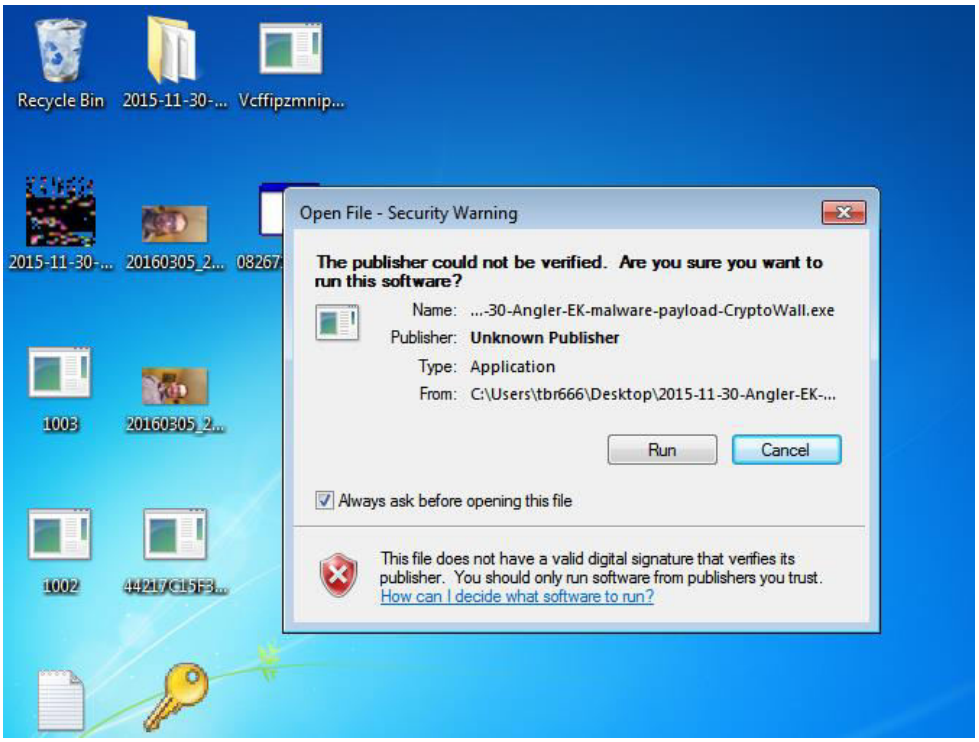
Prije nego započne sa kriptiranjem datoteka, CryptoLocker nastoji sakriti svoje prisustvo od korisnika sustava kako bi značajno smanjio vjerojatnost da njegovo djelovanje bude prekinuto prije nego su sve ciljane datoteke kriptirane. Osim ako se ne radi o verzijama CryptoLockera koje se automatski pokreću korisnik može preuzeti s malicioznim porukama elektroničke pošte, operacijski sustav prilikom pokretanja programa najprije upozorava da se radi o programu iz nepouzdanog izvora koji može biti štetan za sustav (Slika 3.1.), no ako neoprezni korisnik unatoč upozorenju dopusti pokretanje programa, on počinje sa svojim radom. Upozorenje navodi ime, tip i lokaciju datoteke tvrdeći da se ne može potvrditi njen izvor odnosno izdavač te nudi opciju odustajanja od pokretanja i njegovog dopuštanja unatoč riziku.

Nakon pokretanja, CryptoLocker se puni u radnu memoriju i briše s diska (Slika 3.2.), odnosno sekundarne memorije računala te se na slici jasno vidi da je njegova izvršna

datoteka nestala s radne površine ubrzo nakon pokretanja. To će stručnjacima također značajno otežati razumijevanje i ispitivanje njegovog načina rada jer će njegov binarni kod biti teško pronaći. Ispitivanje rada CryptoLockera svodi se na analizu njegovih popratnih pojava i promjena koje uzrokuje na sustavu korištenjem nekoliko naprednih alata. Iako je podatke u radnoj memoriji vrlo teško čitati s korisničkim ovlastima, to je moguće sa administratorskim ovlastima alatom Volatility [20] za memorijsku forenziku.

Drugi mogući način ispitivanja analiza je trenutno aktivnih procesa alatom SysInternals [16] ili jednostavno pomoću Task managera za neke starije verzije zlćudnog koda. Pri tome se primjećuje da je količina radne memorije koju CryptoLocker zauzima vrlo mala iako je kriptiranje koje izvršava vrlo zahtijevno za procesor. CryptoLocker zato može utjecati na brzinu izvođenja ostalih aplikacija tijekom svoga rada i pad performansi računala, što tijekom njegova izvršavanja onemogućuje potpuno uklanjanje sumnje na prisustvo zlćudnog koda na računalu. Neke novije verzije CryptoLockera iz 2015. godine uspijevaju dodatno sakriti svoje djelovanje tako da na popisu aktivnih procesa uopće nije vidljiv niti jedan proces povezan s tim programom.

Još jedan način kojim bi se moglo ispitivati princip rada Crypolockera jest snimanje njegovog mrežnog prometa i analiza njegove komunikacije sa poslužiteljem, odnosno lokacija i tipova resursa s kojima komunicira i tipova podataka koje s njima razmjenjuje, a za to je najpogodniji programski alat Wirehshark [18].



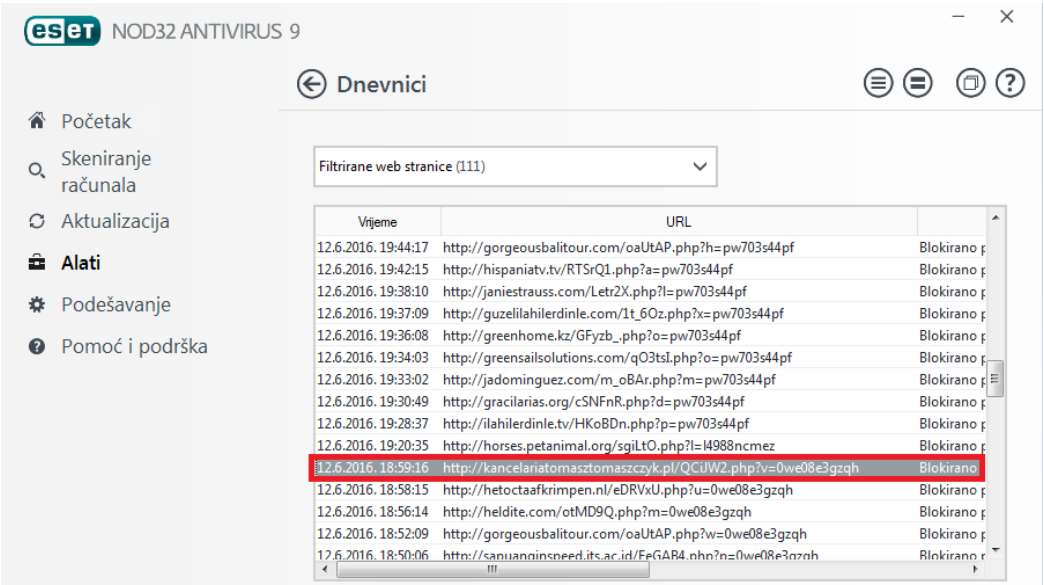
Slika 3.1. Pokretanje potencijalno neželjene aplikacije

Slika 3.2. Brisanje CryptoLockera s diska

3.3. Komunikacija sa poslužiteljem

Za komunikaciju s bazom podataka na svome centralnom poslužitelju u kojoj su pohranjeni ključevi CryptoLocker ne koristi resurse na jednom stalnom poslužitelju koji je u njegovom vlasništvu, nego inficira različite poslužitelje na kojima su smještene javne stranice različitih namjena. Te domene na Internetu naizgled nemaju nikakve povezanosti s CryptoLockerom, no na njima su zapravo smještene php skripte kojima je omogućena pohrana jedinstvenog privatnog ključa s lokalnog računala i signalizacija poslužitelju da je CryptoLocker pokrenut na pojedinom računalu žrtve.

Veza virtualnog stroja s Internetom predstavljena je unutar njegovog sučelja kao povezanost nezavisnim mrežnim kablom, no jasno je da virtualni stroj nema fizičku komponentu povezanosti s Internetom i da svi podaci koji su do njega preneseni moraju do njega proći internetskom vezom računala domaćina. Zbog toga antivirusni program ESSET NOD32, odnosno njegova komponenta koja skenira mrežni promet blokira svaku potencijalno neželjenu mrežnu komunikaciju, bez obzira ide li ona direktno preko veze fizičkog računala ili preko fiktivnog mrežnog kabla virtualnog stroja (Slika 3.3.) Poruka antivirusnog programa koja se pritom prikazuje navodi URL neželjenog resursa kojem se pokušava pristupiti te IP adrese servera na kojem se taj resurs nalazi.



Slika 3.3. Blokirani internetski pristup CryptoLockera

Primjer GET upita kojeg je antivirusni program blokirao prikazan je u prvom retku Tablice 3.1. Adresa http://kandilmesajlari.org pokazuje na domenu na kojoj se nalazi php skripta kojom CryptoLocker komunicira s poslužiteljem, no sama domena nije ni na koji način povezana s CryptoLockerom te postoji neovisno o njegovom komunikacijskom kanalu. Puni URL GET upita upisan u web pretraživač također dovodi do neprovjerene web lokacije koju isti antivirusni program Eset NOD32 blokira. Osim te adrese s kojom je CryptoLocker prvom komunicirao postoji još niz domena i odgovarajućih IP adresa na koje je poslan GET upis s istom vrijednošću argumenta 0we08e3gzqh, za kojeg se može zaključiti da predstavlja jedinstveni privatni ključ za ispitivanje dodijeljen virtualnom računalu tijekom te konkretne ispitivane infekcije (retci 2.-5.) Pokretanjem CryptoLockera drugi puta na oporavljenoj snimki stanja operacijskog sustava generira se novi jedinstveni ključ za taj virtualni stroj (retci 6.-9.)

Tablica 3.1. GET upiti Cryptolockera

Redni broj	**Url upita**	**IP adresa poslužitelja**
1.	http://kandilmesajlari.org/g750fy.php?e=0we08e3gzqh	37.187.152.165
2.	http://horses.petanimal.org/sgiLtO.php?z=0we08e3gzqh	64.14.68.59
3.	http://jadominguez.com/m_qBAr.php?o=0we08e3gzqh	178.62.99.85
4.	http://iaspr.org/5CK4p2.php?f=0we08e3gzqh	66.241.145.10
5.	http://infectiousdiseases.mit.edu/3qUdQ9.php?d=0we08e3gzqh	64.14.68.59
6.	http://horses.petanimal.org/sgiLtO.php?z=c1bz2plb245	52.20.149.53
7.	http://infectiousdiseases.mit.edu/3qUdQ9.php?d=c1bz2plb245	64.14.68.59
8.	http://jadominguez.com/m_qBAr.php?o=c1bz2plb245	178.62.99.85

Jasno je kako je IP adresa poslužitelja na kojem se pojedina skripta nalazi uvijek ista bez obzira na vrijednost ključa, odnosno instancu infekcije. Jedna od slabosti samog CryptoLockera u ovom slučaju je što nakon uklanjanja malicioznih php skripti s inficiranih poslužitelja on više ne radi jer program pokušava komunicirati s nepostojećim resursom te prima kod odgovora 404 koji označava da traženi resurs

ne postoji na toj lokaciji. CryptoLocker ima u sebi ugrađen niz adresa php skripti kojima može pristupiti pa zato u slučaju nepostojanja pojedine skripte ne prestaje sa svojim radom, nego pokušava s toga popisa odabrati novu skriptu kojoj opet šalje privatni ključ korisnika. Ovakav sustav koji ne ovisi o samo jednoj, nego o nizu međusobno neovisnih skripti, od kojih je svaka samo po sebi dovoljna za izvršavanje neophodnog dijela algoritma rada CryptoLockera, vrlo je robustan i otporan na različite mehanizme zaštite jer čak i nakon što je većina njih pronađena i otklonjena on i dalje ima mogućnost komunikacije sa serverom.

Nakon slanja privatnog ključa na server i njegova brisanja iz računala CryptoLocker će početi skenirati sve fizičke ili mapirane mrežne diskove tražeći datoteke koje bi na temelju sufiksa mogle sadržavati potencijalno bitne podatke za korisnika. Svaki puta kada pronađe datoteka ciljanog sufiksa kriptirat će ju koristeći generirani privatni ključ i dodati cijelu putanju do datoteke i njeno ime kao jednu od vrijednosti unutar CryptoLocker registra. Nazivi registara razlikuju se kod pojedinih verzija CryptoLockera, a njihov je naziv primjerice HKEY_CURRENT_USER\Software\CryptoLocker_0388\Files. Popis sufiksa datoteka na koje je usmjeren napad CryptoLockera: odt, ods, odp, odm, odc, odb, doc, docx, docm, wps, xls, xlsx, xlsm, xlsb, xlk, ppt, pptx, pptm, mdb, accdb, pst, dwg, dxf, dxg, wpd, rtf, wb2, mdf, dbf, psd, pdd, pdf, eps, ai, indd, cdr, jpg, jpe, jpg, dng, 3fr, arw, srf, sr2, bay, crw, cr2, dcr, kdc, erf, mef, mrw, nef, nrw, orf, raf, raw, rwl, rw2, r3d, ptx, pef, srw, x3f, der, cer, crt, pem, pfx, p12, p7b, p7c.

3.4. Plaćanje i otključavanje podataka

Kako bi napadači očuvali anonimnost tijekom primanja iznuđenog iznosa, CryptoLocker je uveo sustav plaćanja poznatom virtualnom valutom Bitcoin koja je posljednjih nekoliko godina postala relativno često sredstvo plaćanja na Internetu za sve vrste usluga. Iznos koji treba platiti kako bi se podaci otključali iznosi 2 Bitcoina, odnosno 400 $. CryptoLocker daje određene internetske adrese koje dovode do web mjesta za plaćanje ucjenjivaču, a to su zapravo linkovi za vršenje transakcija Bitcoina koje se inače koriste za takav oblik plaćanja. Starije vrste CryptoLockera uključivale su statičke linkove za plaćanje Bitcoina ucjenjivaču [13]. U kasnijim se verzijama ti linkovi generiraju dinamički za svaki primjerak infekcije, zbog čega je puno teže uči u

trag napadaču i njegovom računu. Ucjenjivačka poruka CryptoLockera (Slika 3.4.) koja se prikazuje na žrtvinom ekranu objašnjava da su sve važne datoteke kriptirane te da je potrebno doći do privatnog ključa kako bi ih se dekriptiralo. Također se navodi traženi iznos i rok do kojeg ga je potrebno platiti, uz upozorenje da će pokušaj uklanjanja ili oštećivanja toga programa odmah uništiti privatni ključ na poslužitelju.

Slika 3.4. Poruka ucjenjivačkog programa

Nakon isteka određenog vremenskog roka jedinstveni privatni ključ najčešće se briše sa poslužitelja, nakon čega više nije moguće pokrenuti dekriptiranje. No ponekad je moguće odgoditi rok plaćanja manipulirajući sustavskim vremenom, budući da se vrijeme do isteka roka dok je računalo ponovno upaljeno može odrediti jedino pomoću njegovog BIOS sustava. Iako sam ucjenjivački zloćudni kod tako tvrdi, privatni ključ ne mora nužno biti izbrisan nakon isteka vremenskog roka pa je plaćanje ipak moguće obaviti ako se oporavi registarski ključ CryptoLocker, nakon toga se odmah ugasi računalo te se u BIOS postavkama vrijeme vrati u prošlost.

Nakon što se plati iznuđeni iznos novca prikazat će se ekran koji tvrdi da se uplata trenutno verificira i taj proces verifikacije može u prosjeku trajati čak 3-4 sata. Nakon što je uplata verificirana, program će početi dekripciju datoteka čiji se popis zajedno s pripadnim putanjama nalazi unutar registra CryptoLockera. Ponekad se može prikazati pogreška da nije moguće dekriptirati pojedinu datoteku i taj se problem vrlo

teško ili gotovo nikako ne može riješiti, ali Crypolocker unatoč tomu nastavlja dekripciju ostatka.

4. Zaštita od CryptoLockera

Implementacija zahtijevanog rješenja podrazumijeva stvaranje dodatka operacijskom sustavu koji mijenja ponašanje datotečnog sustava i uvodi nova ograničenja i zabrane kako bi CryptoLocker spriječio u njegovoj osnovnoj zadaći. Umjesto da se datotečnom sustavu pristupa izravno na nižoj razini, puno je jednostavnije za to koristiti gotovu biblioteku koja će omogućiti izmjene s više razine apstrakcije. Pritom se radi o izmjenama koje se tiču samih struktura podataka na vrlo niskoj razini pa korištenje višeg programskog jezika poput Jave koji ne dopušta manipuliranje pokazivačima i u kojem se sve odvija na visokoj razini apstrakcije, neće biti moguće u implementaciji ovog rješenja. Zato će se koristiti gotova biblioteka Dokan [27] za programski jezik C koja će omogućiti realizaciju potrebnih izmjena.

Kako bi se izvorne datoteke kasnije mogle oporaviti, nužno je u određenim strukturama podataka voditi evidenciju o svakoj kriptiranoj datoteci, odnosno znati pokazivače na izvornu i pripadnu kriptiranu verziju datoteke te kasnije izbrisati kriptirane i vratiti izvorne podatke prelazeći listu evidentiranih parova datoteka. Evidencija se vodi na taj način da se prilikom pokušaja kriptiranja određene datoteke spremi pokazivač na njenu izvornu vrijednost, a nakon preusmjeravanja pisanja kriptirane verzije na drugo mjesto na disku sprema se i pokazivač na adresu kriptirane verzije. Osnovni rješenje za napad Cryptolockera koje će biti osmišljeno i implementirano u okviru zadatka završnog rada ipak neće pružiti zaštitu protiv naprednijih verzija CryptoLockera koji ju različitim mehanizmima pokušavaju zaobići. Zato je potrebno predvidjeti scenarije napada poput korištenja različitih algoritama za kriptiranje, kriptiranja samo nekih blokova i višestrukoj pisanja po istim blokovima radi zauzimanja svog prostora na disku te nadograditi osnovno rješenje s ciljem zaštite od tih scenarija. Za svaki od tih scenarija bit će navedeni algoritam i odgovarajuće programske strukture u C-u pomoću kojih je opisani sigurnosni rizik moguće ukloniti. Prvo će biti opisana C biblioteka Dokan koja omogućuje lakšu implementaciju dodataka datotečnom sustavu.

4.1. C biblioteka Dokan

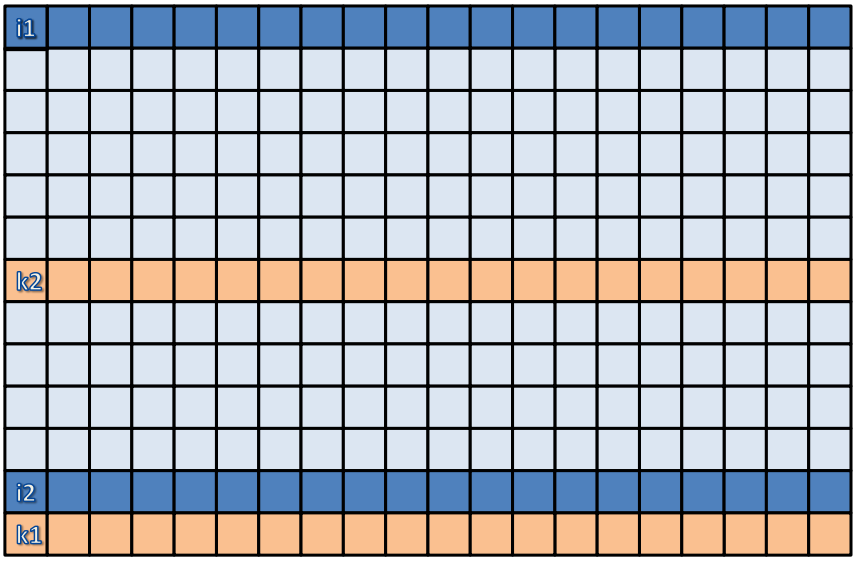
Prije razvoja dodatka datotečnom sustavu koji će onemogućiti pisanje preko postojećih blokova te njihove izmijenjene, odnosno kriptirane verzije pisati na drugom mjestu na disku ukoliko ima još mjesta, potrebno je najprije razviti osnovni datotečni sustav koji će različite naredbe i pokušaje pristupa jednostavno prosljeđivati datotečnom sustavu NTFS. Kasnije će se na tom sustavu raditi izmjene i dodaci koji će omogućiti implementaciju rješenja zadatka. Pri tome nije potrebno u potpunosti razvijati datotečni sustav već za to postoji gotova C biblioteka Dokan [27], odnosno njena poboljšana verzija Dokany [28] koja omogućava korisniku različite prilagodbe kontrole pristupa i dozvola u datotečnom sustavu te izmjenu njegovog ponašanja, odnosno načina na koji reagira na različite korisničke zahtjeve i operacije.

4.2. Osnovno rješenje za napad CryptoLockera

Princip rada CryptoLockera uključuje kriptiranje datoteka ciljanih sufiksa koje se ne mogu dekriptirati ni na koji drugi način osim uz pomoć privatnog ključa s CryptoLockerovog poslužitelja i automatskim dekriptiranjem inverznim algoritmom od onog koji se koristio za kriptiranje. Bit osnovnog rješenja je onemogućavanje CryptoLockera u njegovom izvornom djelovanju koje inače ostavlja isključivo jednu kopiju svake datoteke na računalu i to samo onu kriptiranu koja je potpuno nečitljiva. Ukoliko antivirusni program instaliran na korisnikovom operacijskom sustavu ne detektira i spriječi njegovo djelovanje potrebno je u datotečni sustav ugraditi dodatnu funkcionalnost koja će onemogućiti pisanje preko pojedinih blokova na disku ukoliko ima još mjesta na disku te omogućiti vraćanje izvornih datoteka pri oporavku sustava koje je omogućeno jedino bootanjem posebnog dodatka, a ne unutar samog operacijskog sustava kako CryptoLocker ne bi zlouporabio i tu funkcionalnost.

Kako bi sustav tijekom oporavka znao razlikovati izvorne i kriptirane verzije datoteka te povezati pojedinu kriptiranu datoteku s onom koju korisnik trenutno vidi s izvornom datotekom koju je CryptoLocker prethodno pokušavao napasti, potrebno je u upravljačkoj strukturi datotečnog sustava voditi evidenciju o parovima pokazivača na početak pojedinih izvornih datoteka i kriptiranih datoteka koje njima odgovaraju (Slika 4.1.). Na slici je prikazana pojednostavljena blokovska struktura pohrane datoteka na

disku na kojoj postoje pokazivači na početke dvaju izvornih datoteka (i1,i2) i pokazivači na početke njihovih kriptiranih verzija (k1, k2). To se može postići listom odnosno tablicom struktura koje se sastoje od dva pokazivača, prvog na izvornu datoteku i drugog na kriptiranu datoteku nastalu pokušajem njenog kriptiranja (Ispis 4.1.).



Slika 4.1. Shematski prikaz evidencije kriptiranih datoteka za osnovno rješenje napada CryptoLockera

Takvo bi osnovno rješenje trebalo biti realizirano u dvije etape, odnosno dva zasebna, ali uzročno-posljedično povezana scenarija:

1) CryptoLocker počinje pisati kriptirane blokove pojedine datoteke preko njenih izvornih blokova, no dodatak operacijskom sustavu ga u tome sprječava. Nakon pisanja prvog bloka na određenu lokaciju na disku u tablicu se sprema struktura podataka koja za adresu izvornog bloka bilježi izvorni blok, a za adresu kriptiranog bloka onu lokaciju na disku gdje se taj blok zapisao. Nastavlja se slijedno kriptirati datoteka i slijedno pisati kriptirani blokovi na disk dok se ne kriptiraju svi njeni blokovi. Nakon toga ucjenjivački zloćudni kod kriptira novu datoteku s popisa datoteka sufiksa od interesa te se već nakon kriptiranja i pisanja njenog prvog bloka u polje struktura bilježi novi zapis. Postupak se ponavlja dok se ne prođu sve ciljane datoteke.
2) Na vanjskom prijenosnom mediju za pohranu podataka pokreće se drugi operacijski sustav unutar kojeg je omogućena funkcionalnost oporavka

kriptiranih datoteka. Dodatak sustavu koji služi za oporavak prolazi tablicom zapisanih struktura i briše datoteku s lokacije na koju pokazuje pokazivač njene kriptirane verzije, odnosno sve njene blokove od adrese vrijednosti pokazivača. Pritom datoteku s lokacije na koju pokazuje pokazivač izvorne verzije počinje korisniku pokazivati kao trenutnu verziju datoteke na disku.

```
typedef struct RecoveryFileData{
     FILE * original;
     FILE * crypted;
     RecoveryData  *nextentry;
}  RecoveryData;
RecoveryData startentry;
```

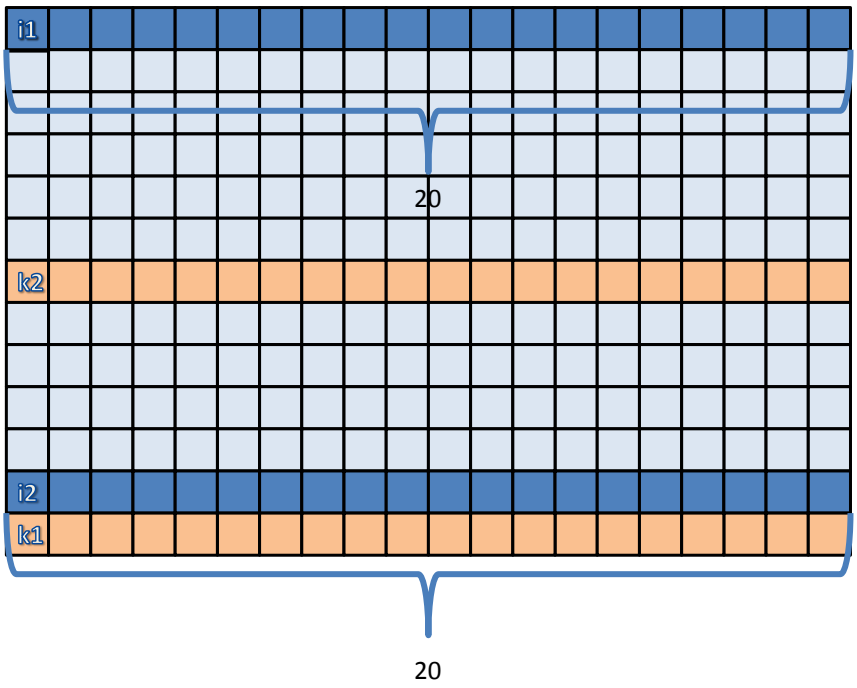
Ispis 4.1.Struktura podataka evidencije kriptiranih datoteka za osnovno rješenje napada CryptoLockera

4.3. Scenarij napada: Korištenje različitih algoritama za kriptiranje

Dosad opisan način rješenja pretpostavlja da je veličina izvornog bloka jednaka veličini kriptiranog bloka, no ovisno o konkretnom algoritmu kriptiranja koje koristi pojedina verzija CryptoLockera to ne mora biti točno, odnosno veličine mogu biti varijabilne. Iako izvorni CryptoLocker za kriptiranje datoteka koristi algoritam AES (*Advanced encryption standard*) [29], moguće je da pojedine verzije koriste različite, odnosno ponešto izmijenjene algoritme. Pri izmijenjenom algoritmu blok kriptiranja ne mora nužno biti blok unutar datotečnog sustava, odnosno može se pri kriptiranju uzimati dva ili više blokova datotečnog sustava odjednom ili njihovi određeni dijelovi. To implicira različite veličine izvornih blokova koje ucjenjivački zloćudni kod kriptira, odnosno kriptiranih blokova nastalih kao rezultat kriptiranja, dakle veličina izvornih blokova ne mora biti jednaka veličini kriptiranih blokova.

Zato je potrebno modificirati ovo rješenje i u svaku strukturu podataka dodati cijeli broj koji će označavati broj blokova pojedine datoteke (Slika 4.2.). Osim oznake početka izvorne (i1, i2) i kriptirane datoteke (k1, k2) na slici je vidljiva i oznaka broja blokova u svakoj pojedinoj izvornoj datoteci. Prvi će se dio rješenja modificirati na taj način da će se nakon početka pojedine datoteke i evidentiranja pokazivača na obje njene verzije u strukturu kao broj blokova trenutne datoteke zapisati početna vrijednosti 1 te će se zapisivanjem svakog novog kriptiranog bloka na disk ta

vrijednost povećati za 1 (Ispis 4.2.). Tijekom operacije oporavka vrijednost u brojaču će se iskoristiti za određivanje kraja kriptirane verzije datoteke.



Slika 4.2. Shematski prikaz evidencije kriptiranih datoteka za rješenje napada korištenjem različitih algoritama za kriptiranje

Ovo rješenje je nužno za stabilnost datotečnog sustava pa ga je svakako potrebno uvesti, no problem je što bi njegova izmjena povećala potreban memorijski prostor za 25%, s 16 bajta za 2 pokazivača po strukturi podataka na 20 bajtova za 2 pokazivača i jednu cjelobrojnu vrijednost, a sama složenost izvođenja u fazi kriptiranja značajno je povećana jer je umjesto evidentiranja samo dvaju vrijednosti pokazivača za svaku datoteku potrebno i za svaki kriptirani blok povećati brojač za 1, dakle umjesto 2 operacije za evidenciju jednu datoteke potrebno je $2+n$ operacija, gdje je n prosječni broj blokova u pojedinoj kriptiranoj datoteci. No taj je algoritam ipak moguće optimizirati tako da se tijekom kriptiranja prve datoteke ustanovi omjer veličine izvornog i kriptiranog bloka. Tijekom oporavljanja datoteka tim se omjerom pomnoži veličina pojedine izvorne datoteke kako bi se ispravno odredila veličina, odnosno kraj njene kriptirane verzije koju je potrebno obrisati.

```
typedef struct RecoveryFileData{
     FILE * originalfile;
     FILE * cryptedfile;
     RecoveryData  *nextentry;
     int size;
    } RecoveryData;

RecoveryData startentry;
```

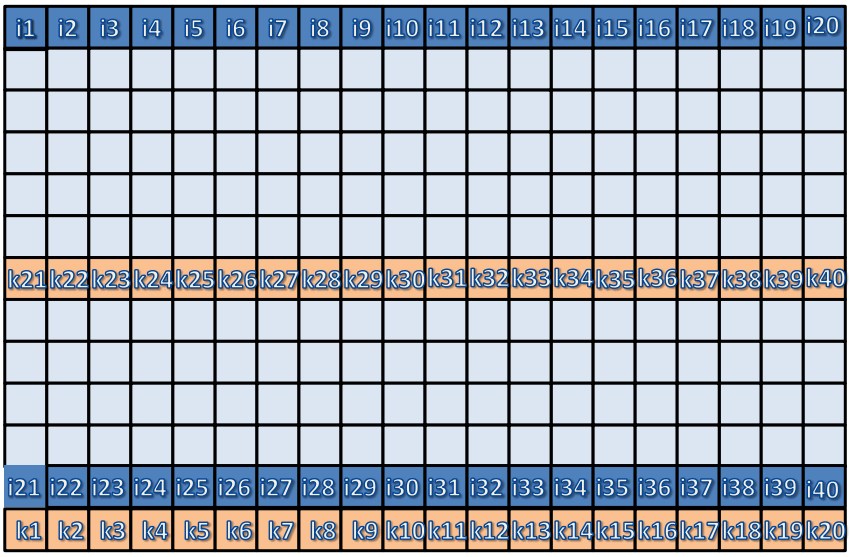
Ispis 4.2. Struktura podataka evidencije kriptiranih datoteka za rješenje napada korištenjem različitih algoritama za kriptiranje

4.4.Scenarij napada: Pisanje samo po određenim blokovima datoteka

Osnovno rješenje pretpostavlja da CryptoLocker datoteke kriptira slijedno i kao cjelinu, dakle da kriptira i pokušava prepisati sve njihove blokove redom te da su blokovi kriptiranih verzija datoteka također smješteni slijedno, točno onim redosljedom kako su smješteni i blokovi izvornih datoteka. No u ovom slučaju CryptoLocker može koristiti činjenicu postojanja određenih ključnih blokova neke datoteke čija je ispravnost nužna za njeno interpretiranje i otvaranje.

Kako bi se identificirali neki uobičajeni formati datoteka, računalo ponekad treba provjeriti samo prvih nekoliko okteta datoteke koju pokušava interpretirati. Ta se skupina okteta najčešće zove identifikacijski okteti (engl. magic bytes) [6], a taj se pojam odnosi upravo na blok okteta uz pomoć kojih aplikacije mogu otkriti je li datoteka koju planiraju parsirati u pravome formatu. Zato bi za potpunu nečitljivost datoteke bilo dovoljno kriptirati samo njen prvi blok koji sadrži identifikacijske oktete. U slučaju takve vrste naprednijeg napada vjerojatno će se osim prvog bloka zbog kriptirati i svakih nekoliko blokova ili će se uz ključne blokove podataka dodatno odabrati više nasumičnih blokova za kriptiranje. Jedino datoteke jednostavnog formatiranja poput običnih tekstualnih datoteka (txt), C datoteka, Java datoteka, zaglavlja (h), Perl skripti (pl), Python skripti (py) i slično koje nemaju identifikacijske oktete otpornije su na kriptiranje te je potrebno kriptirati svaki njihov znak kako bi se tekst učinio nečitljivim i time neuporabljivim.

Takav oblik napada nikako ne bi mogao biti spriječen samo bilježeljem pokazivača na izvorne i kriptirane verzije datoteke i njihovom kasnijom primjenom za oporavak tih datoteka jer bi to moglo dovesti do nepotpunog oporavka datoteka, povratka izvornih blokova krivim redoslijedom ili čak miješanja kriptiranih i izvornih blokova nakon oporavka. Zato je nužno posebno bilježiti pojedine blokove koji se kriptiraju kako bi se oni kasnije pojedinačno mogli vratiti u izvorni oblik na pravu adresu, odnosno u ispravnom redoslijedu (Slika 4.3.). Na slici je vidljivo da sada postoje pokazivači na svaki pojedini izvorni blok (i1-i40) i njegovu kriptiranu inačicu (k1-k40). Taj se problem može riješiti stvaranjem tablice, odnosno liste struktura podataka koje se sastoje od pokazivača na izvorni blok podataka, kojeg korisnik kasnije treba vratiti tijekom oporavka i odgovarajućeg pokazivača na kriptirani blok podataka kojeg korisnik trenutno vidi u sustavu (Ispis 4.3.). Nakon što se izvorni blok datoteke kriptira te njegova kriptirana verzija spremi na neko drugo mjesto na disku, potrebno je u tablicu pokazivača upisati strukturu sa sljedećim vrijednostima: pokazivač na izvorni blok podataka bit će adresa bloka u kojeg je CryptoLocker pokušao pisati, a pokazivač na kriptirani blok bit će na novoj lokaciji na disku.



Slika 4.3. Shematski prikaz evidencije kriptiranih blokova za rješenje napada pisanjem samo po određenim blokovima

Nedostatak ovog rješenja svakako je značajno povećanje prostora potrebnog za spremanje upravljačkih struktura ovakvo nadograđenog datotečnog sustava. Dosad je za to bilo dovoljno *m* struktura u polju pokazivača, gdje je *m* broj kriptiranih datoteka, a sada se ta vrijednost množi s prosječnim brojem (kriptiranih) blokova u

pojedinoj datoteci *n* te iznosi *m*n*. Ovdje je potrebno postići kompromis između povećanja sigurnosti i štednje memorije, no u ovom je slučaju zbog što bolje zaštite potrebno pretpostaviti najgori mogući scenarij u kojem će CryptoLocker kriptirati sve blokove neke datoteke i to u potpuno nasumičnom redoslijedu i zato je ipak nužno žrtvovati toliku količinu memorije kako bi se algoritam u potpunosti realizirao.

```
typedef struct RecoveryFileData{
     void * originalblock;
     void * cryptedblock;
     RecoveryData  *nextentry;
}  RecoveryData;
int blocksize;
RecoveryData startentry;
```

Ispis 4.3. Struktura podataka evidencije kriptiranih blokova za rješenje napada pisanjem samo po određenim blokovima

4.5. Scenarij napada: Višestruko pisanje po istim blokovima

CryptoLocker bi mogao detektirati da mu je pisanje kriptiranih blokova preko izvornih onemogućeno te da se oni pišu na nekoj drugoj lokaciji na disku. Nakon toga bi ucjenjivački zloćudni kod nastavio kriptirati isti blok i pisati ga na disk sve dok se sav slobodni prostor diska ne ispuni kopijama kriptirane verzije toga bloka i time bi prepisivanje izvornog bloka njegovom kriptiranom verzijom uspjelo i CryptoLocker bi nastavio kriptirati ostale blokove. Kako bi se zadovoljilo rješenje prethodno opisanog naprednijeg scenarija napada, opet je potrebno pokazivače na izvorne i njima odgovarajuće kriptirane blokove spremati u tablicu, odnosno listu stuktura, no to rješenje ipak ne pokriva slučaj višetrukog pisanja po istim podacima te ga je potrebno nadograditi.

Cilj je spriječiti da na disku postoji više kopija kriptiranog bloka koje pripadaju istom izvornom bloku, odnosno u kontekstu ovog rješenja da unutar tablice postoji više struktura s međusobno jednakim pokazivačima na izvorne blokove. Kako bi se to postiglo, pri svakom pokušaju pisanja potrebno je realizirati jednostavan algoritam kojim će se tablica ispuniti na ispravan način te ograničiti odnosno minimizirati

ispunjenost diska potrebnu za zadržavanje izvornih blokova podataka. Taj se algoritam sastoji od nekoliko koraka (Ispis 4.4.):

1) provjerava se da li u tablici postoji zapis u kojem pokazivač na izvorni blok podataka pokazuje na lokaciju bloka kojeg se trenutno pokušava prepisati svojom kriptiranom verzijom
2) ako takav zapis u tablici ne postoji, njegova se kriptirana verzija piše na neko drugo mjesto na disku, dok izvorni blok ostaje na svojoj prvotnoj poziciji, u tablicu se pohranjuje zapis koji kao pokazivač na izvorni blok ima adresu bloka na koji se pokušava pisati, a kao adresu kriptiranog bloka odgovarajući adresu na disku koja mu je dodijeljena
3) ako takav zapis postoji izvorni se blok kopira na drugo mjesto na disku te se pokazivač na izvorni blok mijenja na tu adresu, nakon čega se kriptirani blok piše na mjesto gdje je nekad bio izvorni i odgovarajući pokazivač u strukturi unutar tablice nakon toga se ažurira.

Glavna je prednost ovog algoritma što veličina prostora na disku potrebna za stvaranje pričuvnih kopija odnosno naknadni oporavak datoteka postaje konačna, odnosno sustav zaštite sam po sebi postaje izvediv, za razliku od sustava bez toga dodatka kojeg bi ispunio disk bilo koje veličine te od kriptiranja ne bi uspio spasiti niti jedan blok podataka. Prvi je njegov nedostatak to što je za zaštitu pojedine količine podataka odnosno pojedinog broja blokova potreban dvostruko veći prostor na disku. Točnije, to će rješenje stvoriti dvije kopije kriptiranog bloka, jednu na proizvoljnom mjestu na bloku tijekom prvog pokušaja pisanja te jednu na mjestu nekadašnjeg izvornog bloka u drugom pokušaju pisanja kad se ustanovi da se na mjesto tog izvornog bloka već pokušalo pisati. Tada je dodatni prostor potreban na disku za čuvanjem datoteka prosječne veličine n blokova jednak *2*m*n* umjesto *m*n* kao u izvornom rješenju koje ne uzima u obzir ovakav oblik napada.

To bi se moglo jednostavno riješiti izmjenom algoritma u slučaju kada postoji zapis u tablici s adresom izvornog bloka koji se kriptira. Umjesto kopiranja izvornog bloka na drugo mjesto na disku, pri čemu će se adresa izvornog bloka u tablici ažurirati, a adresa kriptiranog bloka postati jednaka nekadašnjoj adresi izvornog, puno je efikasnije prebrisati prvu kopiju kriptiranog bloka kopiranim izvornim blokom. To znači da postoji samo jedan primjerak kriptiranog bloka, pri čemu su adrese izvornog i

kriptiranog bloka podataka zamijenjene u tablici te su oni zamijenjeni fizički na disku. Na taj se blok CryptoLocker tada više neće vraćati jer mu je dopušteno pisanje na njegovu izvornu lokaciju te se ponaša kao da na disku više nema mjesta, a zapravo će se na drugome mjestu stvoriti pričuvna kopija koju će se kasnije moći vratiti.

Drugi je nedostatak algoritma velika složenost provjere postojanja zapisa u tablici koji već sadrži zapis s određenim pokazivačem na izvorni blok podataka, odnosno za svaku pojedinu datoteku veličine od n blokova pri svakom novom pisanju trebalo bi se izvršiti k provjera u tablici za tu datoteku , pri čemu k iznosi broj dotad napadnutih blokova te datoteke, odnosno zapisa u tablici za nju. Zato je broj operacija provjere za datoteku veličini blokova jednak $\sum_{k=0}^{n-1} k$, što iznosi $\frac{n^2}{2} - n$ provjera za cijelu datoteku, odnosno $m*(\frac{n^2}{2} - n)$ za skupinu od m datoteka prosječne veličine n blokova.

Taj bi se problem mogao riješiti provjerom samo zadnjeg zapisa u tablici, pretpostavljajući da CryptoLocker blokove kriptira slijedno i da će se sigurno zadržati na jedno te istom bloku dok god ne ispuni sav prostor diska njegovim kriptiranim verzijama, odnosno neće krenuti na sljedeće blokove nakon neuspjelog pisanja na njegovu lokaciju, no naprednije i inteligentnije verzije CryptoLockera mogle bi se vratiti na taj blok i pokušati ponovno pisati po njemu nakon što je u tablici već stvoreno nekoliko novih zapisa pa zato to poboljšanje nikako nije sigurno i mogli bi na račun smanjenja složenosti postati potpuno neefikasno. Štoviše, CryptoLocker bi se mogao vraćati na blokove ne samo iste datoteke, nego datoteke čije je blokove odavno pokušao neuspješno prepisati te su njihovu kriptirane verzije završile na slobodnom prostoru na disku. U krajnjem slučaju, CryptoLocker bi mogao nakon skupljanja popisa svih datoteka ciljnih sufiksa u memoriju spremiti popis pokazivača na njihove blokove te nasumično odabrati nekoliko blokova koje će kriptirati i ciklički se vraćati na njih sve dok ne zauzme sav slobodan prostor na disku.

To bi impliciralo vrlo nepovoljnu složenost koja bi iznosila čak $\sum_{k=1}^{m*n} k = \frac{(m*n)^2}{2} - m * n$ što bi moglo iznositi jako puno za veliki broj datoteka sastavljenih od puno malih blokova. Zato se može zaključiti da primitivni slijedni zapis zahtijevanih struktura podataka u tablicu ili polje nije pogodan za ovo rješenje jer može uzrokovati veliku složenost dodatke na operacijski sustav i sam mehanizam potreban za njegovo

upravljanje može zahtijevati količinu memorije koja ne postoji u radnoj memoriji računala. Zato bi se vrlo vjerojatno morala koristiti sekundarna memorija s diska, usporediva s veličinom kriptiranih datoteka, što bi značajno usporavalo čitav operacijski sustav.

Rješenje za to je pak napredni način pohrane potrebnih struktura, odnosno korištenjem strukture slične tipu podataka *HashMap* koja je u C-u realizirana tablicom s raspršenih adresiranjem. Takva će struktura za pojedini ključ, odnosno pokazivač na izvornu adresu najbrže naći odgovarajuću adresu kriptiranog bloka, pri čemu je to vrijeme to kraće što je algoritam pronalaženja bloka za pojedini ključ više optimiziran. Glavni su zahtijevi takvog algoritma da se uzastopne izvorne adrese preslikavaju u što udaljenije blokove te da se uzastopne adrese što ravnomjernije rasporede na različite blokove, kako bi veličina pojedinog bloka bila što manja, a broj preljeva iz jednog u drugi blok što veći. Idealni algoritam hashiranja u kojem ne dolazi do prelijevanja iz jednog u drugi blok tablice složenost kriptiranja za skup datoteka smanjio bio na $k*n*m$, gdje je k veličina bloka u tablici hashiranja, a $m*n$ ukupni broj kriptiranih blokova. U ovom bi slučaju trebalo adresi izvornog bloka jednoznačno pridružiti adresu pridruženog bloka pa bi veličina bloka u tablici s raspršenim adresiranjem trebala biti $k=1$, što daje složenost algoritma od $m*n$. Nakon završenog kriptiranja žrtvi će biti vidljiva samo kriptirana verzija podataka, a izvorni će blokovi biti skriveni od CryptoLockera i sačuvani na disku. Pri oporavku kriptiranih datoteka jednostavno će se proći po raspršenoj tablici i za svaku adresu izvornog bloka taj blok prikazivati korisniku kao trenutno, a kriptirane blokove izbrisati.

```
typedef struct RecoveryFileData{
     void * originalblock;
     void * cryptedblock;
     RecoveryData  *nextentry;
}  RecoveryData;

int blocksize;
RecoveryData startentry;
```

Ispis 4.4. Struktura podataka evidencije kriptiranih blokova za rješenje napada višestrukim pisanjem po istim blokovima

Tablica 4.1. Pregled scenarija napada CryptoLockera i odgovarajućih rješenja

Problem	Rješenje	Nedostaci
Prepisivanje izvornih blokovima kriptiranim blokovima	Pisanje svih blokova datoteke na drugo mjesto na disku, brisanje kriptiranih verzija datoteka, korisnik opet vidi izvornu datoteku	Algoritam CryptoLockera može diktirati proizvoljni omjer veličine izvornog i kriptiranog bloka tako da veličina izvorne datoteke ne mora biti ista kao veličina kriptirane
Korištenje različitih algoritama za kriptiranje	Pisanje svih blokova datoteke na drugo mjesto na disku, evidentiranje broja blokova svake izvorne datoteke, određivanje veličine kriptirane verzije pomoću evidentirane veličine izvorne verzije, brisanje kriptiranih verzija datoteka, korisnik opet vidi izvornu datoteku	Moguće je da CryptoLocker ne kriptira sve blokove datoteke, nego samo neke ključne blokove, što uzrokuje veliku nestabilnost u ovakvom rješenju
Pisanje samo po određenim blokovima datoteka	Pisanje zasebnih napadnutih blokova na drugo mjesto na disku pri čemu se pokazivač na svaki blok evidentira , brisanje kriptiranih verzija blokova, korisnik opet vidi izvornu verziju	Moguće je da CryptoLocker namjerno piše po istim podacima više puta kako bi ispunio sav prostor na disku
Višestruko pisanje po istim podacima	Ako napadnuti izvorni blok još nema evidentiranu kriptiranu verziju, pisanje zasebnih napadnutih blokova na drugo mjesto na disku pri čemu se svaki blok evidentira, ako takav zapis postoji dopušta se pisanje preko izvorne verzije i pohranjuje njena kopija na disk, brisanje kriptiranih verzija blokova, korisnik opet vidi izvornu verziju	CryptoLocker može kriptirati blokove potpuno različitim redoslijedom i vraćati se na njih proizvoljni broj puta kako bi pokušao prevariti algoritam evidencije blokova. U tom slučaju potrebno je vršiti provjeru cijele tablice koja treba biti realizirana nekom vrstom raspršenog adresiranja zbog veće efikasnosti

4.6. Scenarij napada: Čuvanje upravljačkih struktura datotečnog sustava

U slučaju svih mogućih rješenja i scenarija napada postavlja se pitanje čuvanja svih struktura podataka i programa potrebnih za njihovo izvršavanje. Kod potreban za drugi dio algoritma, odnosno oporavak sadržan je na vanjskom mediju kao dio posebnog operacijskog sustava tako da mu CryptoLocker tijekom kriptiranja nikako ne može pristupiti. Strukture podataka ili zapise poput struktura podataka u C-u kojima je realizirana evidencija kriptiranja kasnije korištena za oporavak najbolje je spremiti također unutar određenog sistemskog registra ili komponente koja je dio dodatka datotečnog sustava jer ako se spremi unutar datoteke na određenom mjestu unutar računala, i ta datoteka sama može biti kriptirana. Potrebno je napomenuti da jedino izrazito optimiziran algoritam evidencija, poput onog koji koristi tablicu raspršenog adresiranja neće zauzimati jako puno memorije, što je uvjet da može biti spremljen u registru na taj način.

5. Zaključak

Zloćudni ucjenjivački kod ima svrhu kriptiranja svih bitnih korisničkih datoteka na računalu tako da one postaju u potpunosti nečitljive i da je jedini način njihovog oporavka plaćanje iznuđenog iznosa u zamjenu za jedinstveni ključ kojim će se izvršiti njihovo dekriptiranje. Osnovni cilj i motivacija ovog završnog zadatka je učiniti potencijalne žrtve ucjenjivačkog zloćudnog koda imunima na njegovo djelovanje, odnosno omogućiti im oporavak datoteka i uklanjanje zloćudnog koda s računala neovisno o plaćanju iznuđenog iznosa.

Ispitivanje načina rada CrytoLockera i izrada odgovarajućeg rješenja vrlu su riskantni ako se izvode na operacijskom sustavu fizičkog računala jer će u slučaju neuspjeha sve njegove datoteke biti kriptirane i to će stanje biti nepovratno kao u stvarnim situacijama napada. Zato je zadatak bilo potrebno obavljati na operacijskom sustavu virtualnog računala implementiranog unutar neke od virtualizacijskih okolina, a u ovom konkretnom slučaju za to je korišten Oracle VM VirtualBox [19]. Ta virtualizacijska okolina omogućuje spremanje stanja operacijskog sustava unutar snimki stanja operacijskog sustava te je nakon kriptiranja moguće vratiti stanje prije pokretanja CryptoLockera, kao i spremiti stanje u trenutku infekcije.

Prije svoga pokretanja Cryptolocker se učitava u radnu memoriju i briše iz sekundarne memorije diska kako njegov binarni kod ne bi mogao biti pronađen. Zbog toga se ispitivanje rada CryptoLockera vrši ispitivanjem posljedica njegovog rada snimanjem mrežnog prometa i nadziranjem procesa unutar operacijskog sustava. Za snimanje mrežnog prometa koristi se alat Wireshark [18] instaliran na operacijskom sustavu fizičkog računala, a za nadziranje procesa koristi se Windows Sysinternals [16] instaliran na operacijskom sustavu virtualnog računala. CryptoLocker nakon pokretanja s popisa adresa php skripti nasumično bira skriptu pomoću koje središnjem serveru šalje jedinstveni ključ generiran za tu infekciju. Potom se traže datoteke ciljanog formata te se njihove putanje zapisuju u jednom od registara računala. Nakon obavljenog kriptiranja korisniku se ispisuje ucjenjivačka poruka koja traži određeni iznos u zamjenu za jedinstveni ključ kojim će se nakon plaćanja izvršiti dekriptiranje.

Glavna ideja rješenja je spriječiti CryptoLocker da piše preko izvornih blokova datoteka sve dok ima još prostora na disku te preusmjeravati pisanje kriptiranih blokova na drugo mjesto na disku. Pritom će postojati dvije verzije datoteka, izvorna koja je i dalje sačuvana na disku računala i kriptirana koju korisnik vidi kao trenutnu datoteku na računalu. Drugi dio programskog rješenja trebao bi omogućiti brisanje kriptirane verzije datoteke i ponovno prikazivanje izvorne verzije kao one trenutne. Kada bi taj dio rješenja bio omogućen unutar samog operacijskog sustava, CryptoLocker bi to mogao zloupotrijebiti za brisanje izvorne verzije datoteka, tako da taj dodatak mora biti implementiran na operacijskom sustavu pokrenutnom pomoću nekog vanjskog medija. Za osnovni napad CryptoLockera pretpostavlja se da on u svim datotekama ciljanog formata sve blokove napada slijedno, uzimajući pritom jedan po jedan blok datotečnog sustava, pri čemu su kriptirani blokovi jednake veličine i u istom redoslijedu kao blokovi izvorne verzije datoteke. Zato je u okviru osnovnog rješenja napada CryptoLockera dovoljno bilježiti pokazivače na izvorne verzije datoteka i odgovarajuće kriptirane verzije kako bi se dekriptiranje kasnije moglo obaviti. Osnovni napad može uključivati razna poboljšanja kao što su korištenje različitih algoritama za kriptiranje u kojima blok kriptiranja ne mora nužno biti jednak bloku datotečnog sustava, što znači da veličina izvornog nije više jednaka veličini kriptiranog bloka. U tom je slučaju potrebno bilježiti broj blokova izvorne datoteke kako bi se kasnije moglo ustanoviti gdje završavaju svi blokovi njene kriptirane verzije. Umjesto slijednog kriptiranja svih blokova pojedine datoteke naprednije rješenje može kriptirati samo neke ključne ili nasumično odabrane blokove datoteke. Pri tome bi evidencija kriptiranja, odnosno dekriptiranje izvedeno na način kao u osnovnom rješenju napada mogli dovesti do udvostručavanja, gubitka ili krivog redoslijeda blokova izvorne datoteke nakon obavljenog dekriptiranja. Zato je potrebno bilježiti pokazivač na svaki pojedini izvorni blok i odgovarajući pokazivač na njegovu kriptiranu verziju. Postoji i mogućnost dodatno unaprijeđenog napada pri kojem se nastoji pojedine blokove kriptirati više puta. Tako bi se prostor diska ispunio kriptiranim blokovima nakon čega bi se izvorni blokovi mogli neometano prepisati zbog nepostojanja slobodnog prostora na disku na kojeg bi se pisanje preusmjerilo. Za rješavanje toga napada potrebno je onemogućiti preusmjeravanje prepisivanja izvornog bloka ukoliko je to prepisivanje ranije bilo pokušano, odnosno ako za pokazivač na taj izvorni blok već postoji evidentiran odgovarajući pokazivač na kriptirani blok. Umjesto toga, izvorni će se blok kopirati na mjesto gdje je ranije bio

zapisan kriptirani, dopustit će se prepisivanje izvornoga bloka te će se unutar podatkovne strukture za evidenciju zamijeniti pokazivači na izvornu i odgovarajuću kriptiranu verziju. Pretraživanje liste postojećih parova pokazivača oduzelo bi puno vremena i učinilo rješenje neefikasnim kada bi oni bili zapisani unutar povezane liste. Radi toga je zapisivanje parova pokazivača puno bolje izvršiti pomoću algoritma raspršenog adresiranja koji bi svakom pokazivaču na izvorni blok jednoznačno pridružio pokazivač na odgovarajući kriptirani blok.

6. Literatura

[1] *Wikipedia, The Free Encyclopedia*, Zloćudni software, 2.6.2016., https://hr.wikipedia.org/wiki/Zlo%C4%87udni_softver ,10.5.2016.

[2] Repozitorij ispitivanog CryptoLockera s kraja 2014., https://GitHub.com/ytisf/theZoo/tree/master/malwares/Binaries/CryptoLocker_22Jan2014 , 5.5.2016.

[3] Najnovija verzija ispitivanog CryptoLockera, http://www.malware-traffic-analysis.net/2015/11/30/2015-11-30-Angler-EK-sends-CryptoWall-malware-and-artifacts.zip , 5.5.2016.

[4] *Wikipedia, The Free Encyclopedia*,Snapshot (computerstorage), 30.5.2016., https://en.wikipedia.org/wiki/Snapshot_%28computer_storage%29 , 15.5.2016.

[5] *Wikipedia, The Free Encyclopedia*, Copy-On-Write, 29.3.2013.,https://hr.wikipedia.org/wiki/Copy-On-Write , 1.6.2016.

[6] Magicbytes- identifying common files formats at a glance, 8.7.2013.,https://blog.netspi.com/magic-bytes-identifying-common-file-formats-at-a-glance/, 1.6.2016.

[7] JoshuaCannell, CryptoLocker ransomware: Whatyourneed to know, 8.10.2013.,https://blog.malwarebytes.org/101/2013/10/CryptoLocker-ransomware-what-you-need-to-know/, 5.6.2016.

[8] *Wikipedia, The Free Encyclopedia*, Information security, 5.6.2016.,https://en.wikipedia.org/wiki/Information_security, 15.5.2016.

[9] CryptoLocker ransomware informationand FAQ, 1.3.2015.,http://www.bleepingcomputer.com/virus-removal/CryptoLocker-ransomware-information, 1.6.2016.

[10] Dan Raywood, How to avoid being caught out by ransomware, 4.2.2016.,http://www.computerweekly.com/feature/How-to-avoid-being-caught-out-by-ransomware,6.6.2016.

[11] JoshuaCannell, How ransomware spreads and works, 29.4.2015., http://www.combofix.org/how-ucjenjivački zloćudni kod-spreads-and-works.php, 20.5.2016.

[12] *Wikipedia, The Free Encyclopedia*, Botnet, 9.6.2016., https://en.wikipedia.org/wiki/Botnet, 20.5.2016.

[13] Crytpolocker Bitcoin payment, 1.10.2013., http://blockchain.info/address/18iEz617DoDp8CNQUyyrjCcC7XCGDf5SVb, 1.6.2016.

[14] Redit, https://www.reddit.com/, 4.5.2016.

[15] WinRAR and RAR archiver downloads, http://www.rarlab.com/download.htm, 3.5.2016.

[16] Windows Sysinternals, https://technet.microsoft.com/en-us/sysinternals/bb545021.aspx, 6.5.2016.

[17] Torrentz, https://torrentz.eu/, 1.4.2016.

[18] Wireshark, https://www.wireshark.org/, 10.4.2016.

[19] Oracle VM VirtualBox, https://www.virtualbox.org/, 15.3.2016.

[20] Volatility, http://www.volatilityfoundation.org/, 17.6.2016.

[21] VMware, http://www.vmware.com/, 20.3.2016.

[22] XenServer, http://xenserver.org/, 20.3.2016.

[23] Parallels Desktop, http://www.parallels.com/eu/products/desktop/, 20.3.2016.

[24] QEMU, http://wiki.qemu.org/Main_Page, 20.3.2016.

[25] Virtual PC, https://www.microsoft.com/en-us/download/details.aspx?id=4580, 20.3.2016.

[26] *Wikipedia, The Free Encyclopedia*, Virtual machine, 13.5.2016., https://en.wikipedia.org/wiki/Virtual_machine, 20.5.2016.

[27] Biblioteka Dokan, 10.12.2015., http://dokan-dev.GitHub.io/, 15.4.2016.

[28] Biblioteka Dokany, 1.3.2016., https://GitHub.com/dokan-dev/dokany, 16.4.2016.

[29] *Wikipedia, The Free Encyclopedia*, Advanced Encryption Standard, 2.5.2016., https://en.wikipedia.org/wiki/Botnet, 23.6.2016.

Naslov, sažetak i ključne riječi

Naziv teme: Zaštita od ucjenjivačkog zloćudnog koda sprečavanjem prepisivanja podataka na datotečnom sustavu

Naziv teme (engleski): Protection from Ransomware by Prohibiting Data Overwrite on File System

Sažetak: U uvodu završnog rada objašnjeni su pojmovi zloćudnog koda i ucjenjivačkog zloćudnog koda. Glavni zadatak rada je razmotriti zabranu prepisivanja podataka na datotečnom sustavu u svrhu zaštite od zloćudnog ucjenjivačkog programa CryptoLockera. Ispitivanje načina rada CryptoLockera vrši se pomoću virtualnog stroja kako se ne bi ugrozilo fizičko računalo, a za spremanje pojedinih stanja koriste se snimke stanja operacijskog sustava. CryptoLocker svoje djelovanje prikriva spremanjem u radnu memoriju i brisanjem s diska, a za komunikaciju sa poslužiteljem koristi php skripte pohranjene na zaraženim web domenama. Osnovni način zaštite od CryptoLockera je pisanje kriptirane verzije datoteke na drugo mjesto na disku kako bi se izvorna verzija kasnije mogla oporaviti, a postoje i poboljšane verzije tog algoritma koje uzimaju u obzir kriptiranje oznaka za kraj datoteke, različite omjere veličine izvornog i kriptiranog bloka, napad na zasebne blokove i ponavljano napadanje radi zauzimanja svoj prostora na disku. Niti jedan sigurnosni sustav nije savršeno napravljen pa je najbolje način zaštite odgovornim ponašanjem izbjeći da to infekcije uopće dođe.

Sažetak (engleski): In the introduction of my final assingment concepts of malware and ransomware have been explained. The main purpose of the assingment is to consider a prohibition of overwriting data on the file system in order to protect oneself from ransomware named CryptoLocker. The way how CryptoLocker works is tested by a virtual machine to avoid threathening a host computer and snapshots are used to save certain states of the system. Cryptolocker hides its' work by loading itself into the primary memory and deleting itself from the disk, and it uses php scripts located on infected web domain to communicate with the server. The basic way of protection from CryptoLocker is writing a crypted version of a file on other location on the disk to enable a recovery of original version, but there are also some improved versions of the algorithm that consider encrypting end of file marker, different possible ratios od

plain and crypted block size, attacking individual data blocks and repeated attacks that have a purpose of using all diskspace. No security system is perfect and the best way to protect oneself is to avoid malware infection by behaving responsibly.

Ključne riječi: zlоćudni kod, ucjenjivački zlоćudni kod, CryptoLocker, iznuda, zaštita, virtualna testna okolina, zabrana prepisivanje datoteka, oporavak podataka

Ključne riječi (engleski): malware, ransomware, CryptoLocker, extortion, protection, virtual test environment, prohibiting file overwriting, data recovery