ELEKTROTEHNIŠKI VESTNIK 92(1-2): 1-12, 2025
PREGLEDNI STROKOVNI ČLANEK

Zlonamerna programska oprema – trendi, mehanizmi in primeri

Urban Sedlar

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: urban.sedlar@fe.uni-lj.si

Članek podaja pregled področja zlonamerne programske opreme (angl. malware), vključno s kratko zgodovino, značilnostmi, sodobnimi trendi in ključnimi tehnikami, ki jih uporablja. Različne vrste zlonamernega programja klasificiramo glede na namen, funkcionalnost, način vzdrževanja prisotnosti in metode širjenja, ter analiziramo, kako se posamezne kategorije preslikajo na okvir za analizo napadalnih taktik in tehnik organizacije MITRE, imenovan ATT&CK. Posebej izpostavimo popularne tipe zlonamerne programske opreme, vključno z izsiljevalsko, vohunsko in brezdatotečno programsko opremo, korenske komplete, omrežja avtomatskih robotov, viruse, črve, skrita vrata in trojanske konje. Obravnavamo tudi ključne vektorje okužbe, kot so napadi prek dobavne verige, ribarjenje, zloraba trgovin z aplikacijami in socialni inženiring, ter pojasnimo napredne tehnike prikrivanja, vključno z obfuskacijo, izogibanjem razhroščevanju in uporabo dostavljavcev. Analiza primerov podrobneje predstavi tri raznolike primere zlonamerne programske opreme: kriptočrva WannaCry, zlonamerno mobilno aplikacijo za krajo kriptovalut iz družine CherryBlos in nedavni poskus kompromitiranja dobavne verige operacijskega sistema Linux.

Ključne besede: zlonamerna programska oprema, kibernetska varnost, vektorji okužbe, vzvratni inženiring

Malware – trends, mechanisms, and examples

The paper provides an overview of the field of malware, including a brief history, characteristics, modern trends, and key techniques used. Different types of malicious software are classified based on their purpose, functionality, persistence mechanisms, and propagation methods. The analysis examines how these categories map to the MITRE ATT&CK framework which is commonly used for analyzing adversarial tactics and techniques. The paper highlights popular types of malware, including ransomware, spyware, fileless malware, rootkits, botnets, viruses, worms, backdoors, and Trojan horses. It also covers key attack vectors, such as supply chain attacks, phishing, abuse of app stores, and social engineering, while explaining advanced evasion techniques, including obfuscation, anti-debugging techniques, and the use of droppers. A case study analysis presents three distinct examples of malware in detail: the WannaCry crypto worm, mobile malware from the CherryBlos family, used for the cryptocurrency theft, and a recent attempt to compromise the Linux operating system supply chain.

Keywords: malware, cyber security, attack vectors, reverse engineering

1 UVOD

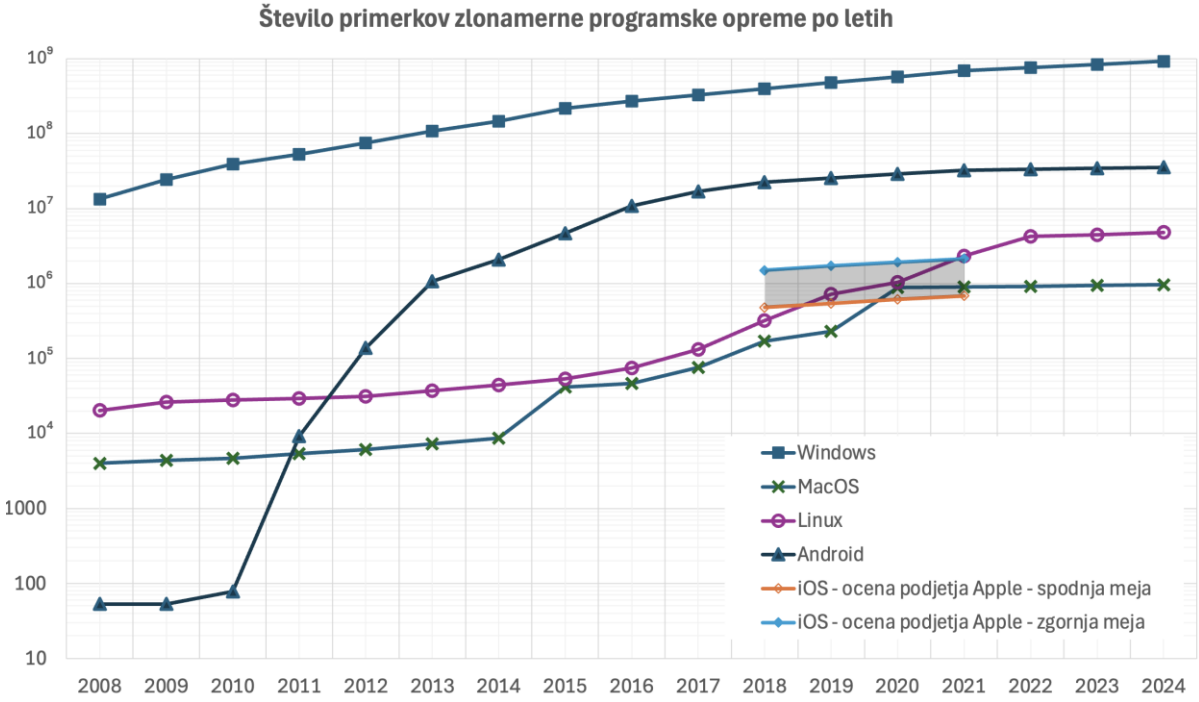
Zlonamerna programska oprema (angl. malicious software – malware) je danes ena od ključnih težav kibernetske varnosti.

Po definiciji ameriškega Narodnega urada za standarde in tehnologijo (angl. National Institute of Standards and Technology – NIST) ta izraz označuje programsko opremo, zasnovano za izvajanje nepooblaščenega procesa, ki negativno vpliva na zaupnost, celovitost ali razpoložljivost sistema [1].

Globalno je škoda, ki jo povzroči zlonamerna programska oprema, ocenjena na milijarde evrov. Že samo za izsiljevalsko programsko opremo so okvirne ocene za leto 2021 približno 20 milijard dolarjev škode in za 2024 42 milijard dolarjev [2]. Ta trend je zaskrbljujoč, a predvidljiv, saj z digitalno transformacijo gospodarstva in družbe naša življenja postajajo vse bolj odvisna od programske opreme, posledično pa ima lahko tudi zlonamerna programska oprema vedno večji negativen vpliv. Slika 1 prikazuje število identificiranih primerkov zlonamerne programske opreme po letih za posamezno platformo [3], [4].

Drugi zaskrbljujoči trend je enormen napredek na področju velikih jezikovnih modelov, ki hitro in nehote postajajo soavtorji programske kode. Medtem ko je večina obstoječe programske kode danes še človeškega izvora, bo že v bližnji v prihodnosti večina kode

Prejet: 28. februar, 2025
Odobren: 20. marec, 2025



Slika 1: Število identificiranih primerkov zlonamerne programske opreme po operacijskih sistemih skozi leta. Za skoraj dva velikostna razreda vodi operacijski sistem Windows, sledi mu Android, razvidna pa je strma rast tudi na drugih platformah. Za iOS ni natančnih podatkov zaradi omejitev pri distribuciji aplikacij (t. i. side-loading), prikazane pa so ocene podjetja Apple. Vir podatkov: [3], [4]

polavtomatsko ali avtomatsko generirana po naročilu; to bo drastično povečalo produktivnost razvijalcev, tudi zlonamernih, odprlo pa bo široka vrata za nove vektorje napadov s kompromitiranimi modeli in pozivi (npr. škodljiv model, ki v kodo podtakne skrita vrata oz. *backdoor*).

V tem članku podajamo pregled izjemno širokega področja zlonamerne programske opreme in podrobneje predstavimo pristope k njeni klasifikaciji ter tipične mehanizme, ki jih takšne aplikacije uporabljajo. V poglavju 5 predstavljene koncepte dodatno ilustriramo na treh realnih primerih – kriptočrvu WannaCry, zlonamerni mobilni aplikaciji in napadu na dobavno verigo operacijskega sistema Linux.

1.1 Zgodovina področja

Področje zlonamerne programske opreme sega v sedemdeseta leta. Prvi računalniški črv, imenovan Creeper (iz angl. plazilec), je bil razvit leta 1971, vendar po svoji funkcionalnosti ni bil škodljiv [5]. Creeper je bil razvit kot eksperiment za demonstracijo izvedljivosti programa, ki se lahko sam replicira in razširi na druge računalnike. Eden od avtorjev je pozneje napisal tudi program za njegovo odstranitev, ki ga danes štejemo za prvi protivirusni program.

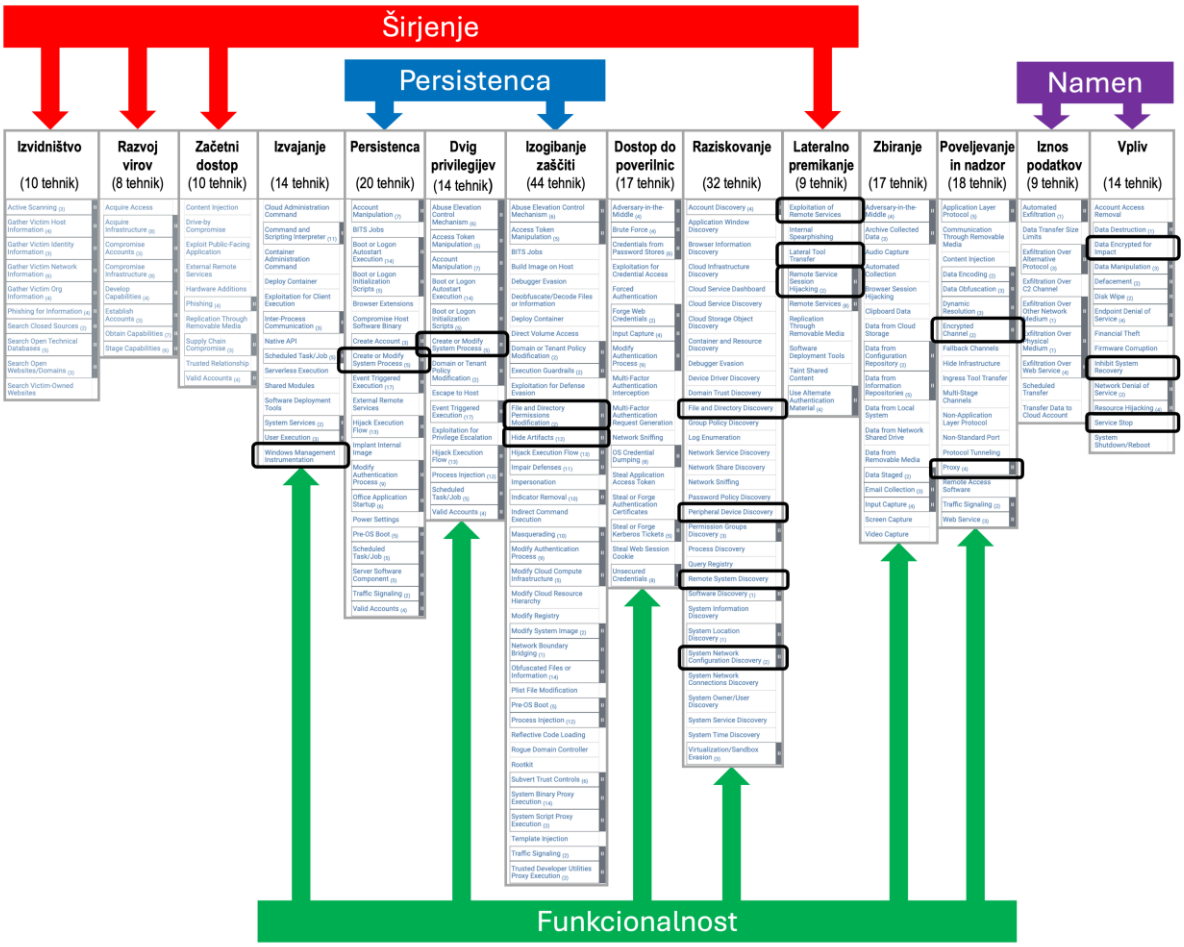
Kmalu zatem so se pojavili prvi primerki škodljive programske opreme, vendar je bila povzročena škoda v številnih primerih zgolj posledica slučajnega izčrpavanja sistemskih virov. Prvih 15 let razvoja je zaznamovalo predvsem eksperimentiranje z novimi pristopi širjenja in razkazovanja spretnosti in šaljivosti razvijalcev.

Prvi načrtno destruktivni virus Jerusalem pa je bil odkrit leta 1987; zasnovan je bil tako, da je na določen datum (petek 13.) izbrisal vse pognane programe [6]. Odkritje tega virusa je pomenilo začetek nove dobe zlonamerne programske opreme, ki se je do danes razvejala na številna pod-področja.

1.2 Motivacija razvijalcev

Pri razumevanju fenomena zlonamerne programske opreme se je smiselno vprašati tudi o motivaciji razvijalcev. Tu nam pomaga, da smo kot ustaljeni prevod besede *malware* izbrali *zlonamerno programsko opremo*, s čimer smo izključili veliko kategorijo škodljive programske opreme, ki ni nastala namerno. Primer slednjega so npr. hrošči, ki imajo prav tako lahko škodljive ali celo katastrofalne posledice. Nedaven in zloglasen primer takšnega hrošča je bil identificiran julija 2024 v programski opremi podjetja CrowdStrike in je po ocenah onesposobil 8,5 milijona računalnikov Windows po vsem svetu – številni med njimi so bili tudi v kritičnih infrastrukturah (letalski prevozniki, banke, bolnišnice ipd.) [7]. Kadar gre za podobne nenamerne napake, lahko razloge iščemo predvsem v veliki kompleksnosti sistema in nezadostnemu testiranju [8].

Pri pravi *zlonamerni* programski opremi pa lahko motivacijo za razvoj in uporabo identificiramo v več



Slika 2: Okvir MITRE ATT&CK in okvirno mapiranje kategorij glede na namen, funkcionalnost, persistenco in mehanizme širjenja zlonamerne programske opreme. S črno obrobo so obkrožene posamezne identificirane tehnike, ki jih uporablja izsiljevalska programska oprema WannaCry.

različnih smereh, pri čemer nam lahko pomagajo okvirji, kakršen je M.I.C.E. [9]:

- Denar (angl. Money) – tipično podlaga za izsiljevalsko programsko opremo, rudarjenje kriptovalut in druge neposredno monetarne učinke.
- Ideologija (angl. Ideology) – podlaga za *haktivizem*, promocijo političnih, verskih in socialnih ideologij.
- Prisila (angl. Coercion) – situacije, kjer je posameznik z grožnjo ali izsiljevanjem prisiljen k razvoju škodljive programske opreme.
- Ego – potreba po prepoznavnosti ali želja po izkazovanju sposobnosti za premagovanje tehničnih ali drugih ovir.

V praksi se izkaže, da je motivacija za večino danes aktivne zlonamerne programske opreme finančno pridobitništvo [10].

2 VRSTE ZLONAMERNE PROGRAMSKE OPREME

Klasifikacija zlonamerne programske opreme ni standardizirana in je težavna zaradi pogostega prepletanja in soodvisnosti posameznih značilnosti. Poimenovanja lahko izvirajo iz namena (oz. poslovnega modela), vgrajenih funkcionalnosti, načina persistence (vzdrževanja prisotnosti v sistemu) ali mehanizma širjenja [11], [12], [13], kot sledi.

- Namen (oz. poslovni model) zlonamerne programske opreme določa njen končni cilj. Med najpogostejše spadajo izsiljevanje (v tem primeru jo v angleščini poimenujemo *ransomware*), vohunjenje (*spyware*), kraja identitete in finančnih podatkov (*stealer*), sabotaža in uničevanje podatkov (*wiper*), rudarjenje kriptovalut (*miner*), goljufije s plačljivimi klici (*dialer*) ter oglaševalske prevare z izvajanjem lažnih klikov (*clicker*).
- Funkcionalnost se nanaša na delovanje oz. zmožnosti zlonamerne programske opreme; tipični primeri so npr. omogočanje stranskih vrat (*backdoor*) za prikrit

dostop, beleženje tipkanja (*keylogger*), snemanje zvoka in telefonskih klicev, oddaljen nadzor naprave (angl. remote access), oddaljen nadzor same zlonamerne programske opreme (angl. Command & Control), prestrezanje prometa in manipulacija omrežnega prometa, odkrivanje omrežnih virov ipd.

- Način vzdrževanja prisotnosti opisuje, kako zlonamerna programska oprema ostane prisotna, aktivna in skrita v sistemu. Primeri vključujejo modificiranje drugih programov (virusi), modificiranje sistemskih procesov (korenski kompleti oz. rootkiti) ter uporabo tehnik prikrivanja zlonamernosti: obfuskacije, terminacije neželenih procesov, zavajanja uporabnika, zakrivanja sledi ter oteževanja analize.
- Mehanizem širjenja pa opisuje, kako zlonamerna programska oprema doseže tarčo; primeri vključujejo izkoriščanje ranljivosti sistemov (črvi), tehnike socialnega inženiringa (trojanski konji), kampanje ribarjenja (angl. phishing) in napade prek dobavne verige.

Težava klasifikacije je, da se v praksi naštete kategorije pogosto prekrivajo ali pojavljajo skupaj. Zlonamerna programska oprema lahko na primer uporablja mehanizem širjenja s trojanskim konjem, da se prikrito namesti v sistem, nato pa omogoči oddaljen dostop za namen vohunjenja in kraje podatkov.

Bolj sistematična analiza je mogoča z okvirjem MITRE ATT&CK [14], ki popisuje večino znanih taktik in tehnik za izvajanje kibernetskih napadov. MITRE ATT&CK lahko tako služi za natančno opredelitev delovanja zlonamerne programske opreme, saj se lahko z analizo primerka vedno določi nabor tehnik, ki jih uporablja. Primer prikazuje Slika 2 (črni okvirčki so tehnike, ki jih uporablja kriptočrv WannaCry). Razvidno je tudi okvirno mapiranje naštetih štirih kategorij (namen, funkcionalnost, persistenca in širjenje) na 14 kategorij (stolpci) okvirja MITRE ATT&CK.

Številni formalni dokumenti (vključno z NIST SP 800-83r1: *Guide to Malware Incident Prevention and Handling for Desktops and Laptops, [15]*) pa kategorije poimenovanja obravnavajo ohlapno, zato bomo v nadaljevanju podali kratek opis nekaj najpogostejših razredov zlonamerne programske opreme.

2.1 Izsiljevalska programska oprema (ransomware)

Izsiljevalska programska oprema se je prvič pojavila leta 1989 v obliki trojanskega konja z imenom AIDS; avtor je bil evolucijski biolog in je razdelil okrog 20 tisoč okuženih disket na mednarodni konferenci na temo sindroma pridobljene imunske pomanjkljivosti (AIDS) [16]. Program je beležil število zagonov računalnika in po 90 zagonih skril imenike in šifriral imena datotek, za povrnitev dostopa pa so žrtve morale poslati 189 dolarjev na poštni predal v Panami.

Od takrat je tehnologija močno napredovala, ideja pa je ostala enaka; izsiljevalska programska oprema deluje z dvema mehanizmoma:

- preprečitev dostopa do datotek (šifriranje); in/ali
- eksfiltracija (iznos) datotek in grožnja z objavo,

oboje pa naj bi bilo s plačilom mogoče preprečiti. Danes je primaren način plačila z uporabo kriptovalut, ki so že po zasnovi psevdonimne (npr. bitcoin) ali celo anonimne (npr. monero), pri obeh grožnjah pa se pojavlja kompleksna dilema zaupanja napadalcu: ali bomo podatke res dobili nazaj, če plačamo? (Oziroma: ali bo napadalec po plačilu res izbrisal eksfiltrirane podatke?) [17], [18] Pri prvem mehanizmu (šifriranje) si lahko pomagamo z varnostnimi kopijami, pri drugem pa te ne pomagajo.

Zaupanje je pomembna komponenta poslovnega modela izsiljevalcev, zato se pogosto meni, da se bodo zaradi teorije iger obnašali pošteno, saj z nepoštenim ravnanjem pokvarijo vzdržnost poslovnega modela zase in za druge [17]: če žrtev ocenjuje, da podatkov kljub plačilu ne more dobiti, plačila sploh ne bo izvedla. Dodatno težavo povzročajo politično motivirani primeri, kakršen je bil NotPetya, ki je podatke preprosto izbrisal, potem pa zahteval odkupnino na neobstoječ bitcoin naslov – na videz znak nekompetentnosti, v resnici pa uničevalec podatkov (wiper) pod pretvezo izsiljevalske programske opreme. Podrobnejši primer izsiljevalske programske opreme predstavimo v poglavju *5.1 Kriptočrv WannaCry.*

2.2 Vohunska programska oprema (spyware)

Vohunska programska oprema (angl. spyware) se uporablja za zbiranje in eksfiltracijo podatkov o posamezniku ali organizaciji, s čimer je kompromitirana dimenzija zaupnosti. Čeprav je zbiranje podatkov o uporabniku v širšem smislu postalo priljubljen poslovni model, ki ga uporabljajo številna znana podjetja (npr. Meta, Google) in celo ponudniki protivirusne programske opreme (Avast) [19], pa vohunska programska oprema v ožjem smislu targetira veliko bolj občutljive podatke: prijavne poverilnice, številke kreditnih kartic, digitalne denarnice, zgodovino brskanja ipd. [20] Zbrani podatki se nato pogosto prodajo na črnem trgu oz. uporabijo za izvajanje goljufij. Ena resnejših posledic okužbe z vohunsko programsko opremo je kraja identitete, pri kateri napadalci zberejo zadostno količino osebnih podatkov žrtve, da se lahko v njenem imenu prijavijo v spletne storitve, odprejo bančne račune ali celo pridobijo kredite.

Kategorija vohunske programske opreme, ki se imenuje *stealerji*, predstavlja aplikacije, specializirane za krajo prijavnih poverilnic, shranjenih gesel, piškotkov in drugih podatkov, ki olajšajo nepooblaščen dostop do uporabniških računov. Kraja piškotkov lahko na primer omogoča prijavo brez vnosa gesla, s čimer lahko napadalci pridobijo dostop do uporabniških računov brez sprožitve varnostnih preverjanj, kakršna je dvofaktorska avtentikacija.

T. i. *keyloggerji* pa so vrsta vohunske programske opreme, ki beleži pritiske tipk na tipkovnici, s čimer lahko prestreže prijavne podatke, osebna sporočila, iskalne poizvedbe in celo vsebino dokumentov. Naprednejši primerki lahko zajemajo tudi posnetke zaslona, kar še dodatno povečuje potencial za zlorabo.

2.3 Korenski kompleti (rootkits)

Korenski kompleti (angl. rootkit) so zasnovani za prikrito delovanje v okuženem sistemu. Njihov glavni namen je omogočiti napadalcem trajen in neopažen dostop do kompromitiranega sistema, hkrati pa skriti svojo prisotnost pred uporabnikom in varnostnimi rešitvami. Rootkiti lahko delujejo bodisi v uporabniškem prostoru bodisi v jedru operacijskega sistema, pri čemer so slednji mnogo večja grožnja zaradi globoke integracije v sistemske procese [21], [22].

Ena izmed ključnih značilnosti rootkitov je njihova sposobnost prikrivanja svoje prisotnosti in aktivnosti drugih zlonamernih programov. To dosežejo z manipulacijo sistemskih klicev (angl. system calls), skrivanjem procesov, datotek in omrežnih povezav ter spreminjanjem izpisov diagnostičnih orodij. Rootkit lahko na primer prestreže ukazne poizvedbe (*tasklist*, *netstat*) in prikaže lažne rezultate, ki ne vsebujejo procesov, povezanih z zlonamerno programsko opremo.

2.4 Omrežja avtomatskih robotov (botnets)

Botnet je omrežje računalnikov, strežnikov, naprav interneta stvari (angl. internet of things – IoT) ali mobilnih telefonov, ki so okuženi z zlonamerno programsko opremo, ta pa je pod oddaljenim nadzorom napadalca [23].

Okužene naprave imenujemo (ro)boti ali zombi naprave, delujejo pa kot del porazdeljenega sistema, ki ga napadalec upravlja prek svojega kontrolnega strežnika (angl. Command & Control – C2).

Botneti se pogosto uporabljajo za izvajanje porazdeljenih napadov zavrnitve storitve (angl. Distributed Denial of Service – DDoS), pošiljanje neželene pošte, rudarjenje kriptovalut in podobno.

2.5 Virusi

Virusi so programi, ki se ob zagonu replicirajo, tako da okužijo gostiteljske datoteke. Najpogosteje gre za izvršljive datoteke (v binarni obliki ali kot skripte), lahko pa tudi programabilne dokumente Office (v tem primeru govorimo o makrovirusih). V primeru zagonskih virusov je gostitelj zagonski (angl. boot) sektor pogona oz. medija [24]. Virusi za širjenje potrebujejo dejanje uporabnika, torej zagon ali odpiranje okužene datoteke, in se v tem razlikujejo od črvov.

Virus običajno vsebuje dve glavni komponenti: rutino za širjenje, ki skrbi za okužbo drugih datotek ali sistemov, ter zlonameren tovor, ki izvaja škodljive aktivnosti (npr. brisanje podatkov). Ključna tehnika, ki jih virusi uporabljajo, je ugrabitev izvajanja programa (angl. execution hijacking); ko virus okuži izvršljivo datoteko, spremeni njeno strukturo tako, da se ob zagonu najprej izvede koda virusa, nato pa se lahko prenese nadzor tudi na prvotno aplikacijo, da uporabnik ne posumi na okužbo. Virus ugrabi izvajanje programa bodisi s prepisovanjem začetnega dela kode (t. i. vstopna točka – *entry point*), spremembo naslovov pogosto uporabljanih funkcij ali vrivanjem dinamičnih knjižnic [24].

Ker virusi dodajo svojo kodo v izvršljive datoteke, pustijo za sabo značilen podpis, na podlagi katerega lahko protivirusni programi detektirajo potencialno okužbo. Ti delujejo na osnovi *definicij*, to so baze znanih podpisov virusov. Za testiranje uspešnosti detekcije je definiran tudi poseben podpis, t. i. testna datoteka EICAR s spodnjo vsebino, ki mora vedno sprožiti detekcijo okužbe [25]:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-
STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Polimorfni virusi [26] se proti taki detekciji borijo s spreminjanjem inštrukcij (obfuskacija ali šifriranje), kar onemogoča enostavno detekcijo na osnovi podpisa, zato je treba v takšnih primerih uporabiti hevristike, ki iščejo pogoste vzorce (npr. uporabo dešifrirne rutine) ali analizo obnašanja programa (npr. dostop do kritičnih datotek, vzpostavljanje omrežnih povezav ipd.).

2.6 Črvi (worms)

Računalniški črvi so programi, ki se propagirajo prek omrežja [27]; to je tipično mogoče zaradi varnostnih lukenj v ciljnem računalniku – bodisi neustrezne konfiguracije (privzeta gesla), šibkih gesel, ugibanih z grobo silo, ali ranljivosti, ki omogočajo zlorabo sistema brez ustrezne avtorizacije. Ena od možnih »varnostnih lukenj« je tudi človek, posledično je priljubljen kanal za širjenje črvov tudi elektronska pošta. Zaradi takšne rekurzivne narave širjenja se črv lahko širi izjemno hitro oz. celo eksponentno.

2.7 Brezdatotečna zlonamerna programska oprema (fileless malware)

Brezdatotečna zlonamerna programska oprema je prisotna izključno v pomnilniku in na disku ne pušča klasičnih sledi v obliki izvršljivih datotek [28]. Takšen način delovanja ji omogoča, da se izogne številnim metodam detekcije, saj tradicionalni varnostni sistemi (npr. protivirusni programi) pogosto temeljijo na pregledu datotek in njihovih podpisov. Prisotnost zgolj v pomnilniku pa oteži tudi potencialno forenzično analizo, saj je za detekcijo zlonamernega programa potreben posnetek pomnilnika (angl. memory dump).

Takšna zlonamerna koda pogosto uporablja sistemska orodja, kakršna sta *PowerShell* – napredna ukazna lupina na sistemih Windows – in *PsExec*, ki omogoča oddaljeno poganjanje programov in ukazov na drugih računalnikih v omrežju brez uporabe terminalnega dostopa oz. brez storitev oddaljenega namizja (angl. remote desktop services).

Obramba pred brezdatotečno zlonamerno programsko opremo zahteva večplasten pristop, ki je sestavljen iz omejevanja pravic dostopa in uporabe problematičnih sistemskih orodij ter na drugi strani spremljanja nenavadnih aktivnosti teh orodij in uporabe vedenjske analize za zaznavanje sumljivih vzorcev delovanja. To danes omogočajo rešitve za zaznavanje in odzivanje na končnih točkah (angl. endpoint detection and response – EDR) in rešitve za razširjeno zaznavanje in odzivanje (angl. extended detection and response – XDR).

2.8 Skrita vrata (backdoors)

Skrita vrata so metoda dostopa do sistema, ki omogoča obvod vgrajenih varnostnih mehanizmov. Lahko so vgrajena namerno, na primer v primeru diagnostičnega dostopa (kot so posebno prednastavljeno geslo za diagnostiko, »telefoniranje« domov ali daljša serija korakov, ki onemogoči varnostne mehanizme), ali za namene vohunjenja. Pogosto pa se zgodi, da se *de facto* skrita vrata vzpostavijo nenamerno: privzeta gesla, ki jih uporabnik ne zamenja, odkrite varnostne luknje, ki niso pravočasno popravljene, ipd.

2.9 Trojanski konji (trojans)

Poimenovanje *trojanski konj* izvira iz starogrške mitologije, kjer so se grški vojaki skrili v velikega lesenega konja neškodljivega videza, in tako vstopili skozi mestno obzidje Troje. V informacijsko-komunikacijskih sistemih se takšna taktika uporablja za skrivanje škodljive funkcionalnosti v uporabno in na videz benigno programsko opremo. To predstavlja izjemno široka vrata, saj pri zaprtokodni programski opremi pravzaprav nikoli ne moremo vedeti, ali poleg želenih ne vsebuje tudi škodljivih elementov.

Za zaščito pred trojanskimi konji je ključna kibernetska higiena, ki vključuje prenos programske opreme le iz zaupanja vrednih virov, redne posodobitve sistema, omejevanje privilegiranega dostopa nepreverjenim programom in uporabo rešitev EDR/XDR.

3 VEKTORJI OKUŽBE

Vektor okužbe je metoda ali pot, prek katere zlonamerna programska oprema vstopi v sistem. V nadaljevanju opišemo nekaj pogostih vektorjev okužbe.

3.1 Dobavna veriga

Dobavna veriga v svetu programske opreme je analogna proizvodni dobavni verigi, le da odvisnost od fizičnega pretoka blaga zamenjajo programske odvisnosti (angl. dependencies) [29]. Danes večina kompleksnih programskih produktov nastane s pouporabo in nadgradnjo obstoječih komponent (operacijski sistemi, aplikacije, knjižnice), kar pomeni, da je velika količina aktivne kode v sodobnem projektu zunanjega izvora – v primeru binarnih datotek pa dodatno tudi brez dobrega vpogleda v njeno vsebino.

Ker so pomembni sistemi pogosto dobro zaščiteni, je napad prek dobavne verige v zadnjih letih postal izjemno privlačen, saj meri na pogosto slabše varnostne mehanizme pri samem izviru programske opreme (npr. odprtokodna programska oprema, ki sprejema popravke skupnosti); zlonamerna programska oprema pa od tam sčasoma doseže široko prisotnost v nameščenih sistemih (konkreten primer v poglavju 5.3).

Obramba zahteva strogo preverjanje dobaviteljev, zaščito procesov integracije in nameščanja ter spremljanje integritete programske opreme.

3.2 Trgovine z aplikacijami

Trgovine z aplikacijami so se iz mobilnih operacijskih sistemov razširile tudi na namizne in so danes prisotne v ekosistemih iOS, Android, Windows in macOS. Uporabniki temu distribucijskemu kanalu tipično bolj zaupajo zaradi varnostnih pregledov aplikacij pred objavo, takšne trgovine pa so v nekaterih primerih tudi edini način namestitve aplikacij (npr. na naprave z Apple iOS) [30]. Posledično je objava zlonamerne aplikacije v trgovini lukrativni cilj in razvijalci uporabljajo različne tehnike skrivanja škodljivega tovora ali namestitve v več fazah (glej poglavji 4.3 in 5.2).

3.3 Oglaševanje in povezave na družbenih omrežjih

Napadalci izkoriščajo oglaševanje in povezave na družbenih omrežjih kot vektor napada z uporabo zlonamernih oglasov (angl. *malvertising*), lažnih profilov in zavajajočih objav, ki usmerjajo žrtve na ponarejene ali okužene spletne strani, ali pa omogočajo neposreden prenos zlonamerne programske opreme. Platforme, kot so Facebook, Twitter/X, LinkedIn in Instagram, imajo širok doseg, kar olajša širjenje zlonamernih vsebin prek sponzoriranih objav ali kompromitiranih računov. Pogoste tehnike vključujejo lažne nagradne igre, prevare s kriptovalutami in aplikacije, ki izkoriščajo uporabnikovo zaupanje priljubljenim blagovnim znamkam ali znanim osebnostim.

Obramba zahteva kritično presojo povezav, preverjanje virov ter uporabo orodij za zaznavanje ribarjenja in blokiranje zlonamernih oglasov.

3.4 Ribarjenje/targetirano ribarjenje (spear phishing)

Ribarjenje ali lažno predstavljanje je način zavajanja uporabnikov, ki je namenjen pridobivanju občutljivih osebnih podatkov in gesel, prav tako pa je uporaben za širjenje zlonamerne programske opreme. Mehanizmi zavajanja so lahko povsem tehnični, npr. prenos datoteke prek povezave ali priponka v sporočilu. Na tem področju je mogočih več prevar, npr. uporaba *unicode* kontrolnih znakov za spremembo smeri besedila z leve proti desni (angl. left-to-right – LTR, tipično za latinske jezike) na desno proti levi (angl. right-to-left – RTL, tipično za npr. arabske jezike); s tem se lahko ime škodljive datoteke

`fdp.tnemukod.exe` prikaže kot `exe.dokument.pdf`.

Zavajanje pa je lahko tudi človeške narave; tak primer bi vključeval načrtno pridobivanje zaupanja in nato pošiljanje programa, ki zadeva interes prejemnika.

4 PRIKRIVANJE ZLONAMERNOSTI

Eden od pomembnih dejavnikov uspeha zlonamerne programske opreme je tudi prikrivanje zlonamernosti. V nadaljevanju opišemo nekaj pogostih tehnik.

4.1 Obfuskacija

Obfuskacija (zameglitev) je tehnika prikrivanja pravega pomena in strukture programske kode z namenom oteževanja razumevanja, zaščite intelektualne lastnine in preprečevanja vzvratnega inženiringa, v primeru zlonamernega programja pa je glavni cilj skrivanje zlonamerne kode. Obstaja več tehnik in vzorcev, s katerimi lahko inštrukcije avtomatsko prepišemo v težje razumljivo obliko.

Idejo najlažje predstavimo na primeru: namesto da bi programska koda vsebovala jasno berljivo spremenljivko

```
geslo = "tajno123",
```

bi bila obfuskirana različica lahko videti kot

```
gX1 = chr(116) + chr(97) + chr(106) +
chr(110) + chr(111) + chr(49) + chr(50) +
chr(51).
```

V tem primeru smo besedilni niz pretvorili v inštrukcije, ki ta niz sestavijo iz ASCII-kod znakov. Na tak način niza »tajno123« v kodi ne moremo več enostavno detektirati. Podoben učinek bi imelo tudi šifriranje niza.

Obfuskacija danes ni več ročni proces, saj obstaja izjemno dovršena programska oprema (npr. [31]), ki za uporabnika avtomatizira vse kompleksne procese, od šifriranja nizov in uporabljenih datotek do preprečevanja zaslonskih posnetkov in zagotovitve unikatnega prstnega odtisa.

4.2 Izogibanje razhroščevanju in vzvratnemu inženirstvu

Zlonamerna programska oprema pogosto vsebuje tudi kodo, ki zazna ali prepreči, da bi se na proces pripel razhroščevalnik. Med takšnimi tehnikami so preverjanje prisotnosti razhroščevalnika prek sistemskih klicev (npr. *IsDebuggerPresent* v Windowsih ali *ptrace* v Linuxu), branje ali nastavljanje registrov za odkrivanje nastavljenih prekinitvenih točk (angl. breakpoint) in spreminjanje toka programa, če je razhroščevalnik zaznan. V primeru zaznave razhroščevalnika lahko spremenjen tok programa močno oteži analizo in podaljša čas, ki je potreben za razumevanje delovanja programa.

4.3 Uporaba dostavljavca

Dostavljavci so majhni zlonamerni programi, zasnovani predvsem za neopazen prenos in namestitev dodatne škodljive programske opreme na sistem. Običajno uporabljajo tehnike pakiranja (obfuskacija, kompresija, šifriranje), da obidejo varnostne mehanizme in prikrijejo svojo aktivnost. Ko enkrat pridobijo začetni dostop do sistema, lahko nadaljujejo z naslednjo stopnjo napada in na sistem namestijo ciljno škodljivo programsko opremo.

4.4 Zavajanje uporabnika

Pri prikrivanju zlonamernosti imajo tehnike zavajanja uporabnika pomembno vlogo predvsem v situacijah, kjer ni nameščene nobene zaščitne programske opreme; pogost primer tega so mobilne naprave.

Zlonamerna aplikacija lahko z zavajanjem (potvarjanje ikon in vizualnih elementov) ustvari videz legitimnosti in zmanjša previdnost žrtve, kar poveča verjetnost, da jo uporabnik prostovoljno namesti oz. zažene. Po namestitvi lahko zlonamerna aplikacija s pomočjo dostavljavca namesti škodljivi tovor, skrije ikono nameščene aplikacije, pri namestitvi pa uporabi nevpadljiva imena imenikov in datotek, kar otežuje detekcijo s strani uporabnika.

5 PRIMERI

V tem poglavju podajamo kratko analizo treh raznolikih primerov zlonamerne programske opreme, s katerimi lahko bolje ilustriramo opisane tehnike.

5.1 Kriptočrv WannaCry

WannaCry sodi v kategorijo šifrirne izsiljevalske programske opreme, za širjenje pa je uporabljal mehanizem črva. Pojavil se je maja 2017, njegova glavna funkcionalnost pa je bila šifriranje datotek na žrtvinem disku, čemur je sledilo izsiljevanje s plačilom v valuti bitcoin. [32] Za plačilo odkupnine je izvajal še dodaten časovni pritisk z grožnjo po podražitvi in, končno, izbrisu datotek po nekaj dneh.

WannaCry je uporabljal ranljivost v popularnem protokolu za prenos datotek, SMB (angl. Server Message Block), ki jo je odkrila in uporabljala ameriška Nacionalna varnostna agencija (angl. National Security Agency – NSA). NSA je za izkoriščanje te ranljivosti razvila orodje, imenovano *EternalBlue*, vendar jim ga je v kibernetskem napadu marca 2017 ukradla hekerska skupina *The Shadow Brokers* [32]. Po neuspešni dražbi je skupina aprila 2017 orodje javno objavila, mesec po tem pa je že bilo uporabljeno za širjenje kriptočrva WannaCry. Transportni sistem črva je orodje *EternalBlue* uporabil, da je pridobil dostop do ranljivih računalnikov, za namestitev in izvršitev pa je uporabljal še eno orodje iz arzenala NSA, *DoublePulsar*, ki je bilo prav tako ukradeno v istem hekerskem napadu [32].

Slika 3: Pogovorno okno kriptočrva WannaCry.

Po zagonu je kriptočrv najprej preveril dostopnost določenega domenskega imena, ki je služilo kot stikalo za izklop (angl. kill switch). Če domensko ime ni obstajalo, je zašifriral podatke na računalniku, nato pa se je poskušal prek ranljivosti v protokolu SMB lateralno razširiti na računalnike na lokalnem omrežju ter na javno dostopne strežnike ter na vsakega, ki je bil ranljiv, namestiti svojo kopijo.

Podjetje Microsoft je ranljivost v operacijskem sistemu Windows odpravilo že marca 2017, dva meseca pred pojavom črva. Ranljivost je tako prizadela samo računalnike brez nameščenih popravkov. Po ocenah je bilo prizadetih več kot 200 tisoč računalnikov v več kot 150 državah, vendar pa je večji delež pripadel državam, ki so (bile) znane po uporabi nelicenčne in zastarele programske opreme; podatki kažejo na veliko število prizadetih sistemov v Rusiji, Ukrajini, Indiji in na Kitajskem. Po podatkih britanske Nacionalne zdravstvene službe je bilo tudi v britanskih bolnišnicah prizadetih več kot 70 tisoč naprav, ki so uporabljale zastarele različice operacijskega sistema Windows.

Poznejša analiza je pokazala, da napadalci niso generirali unikatnih bitcoin naslovov za vsako žrtev, kar bi jim omogočalo sledenje posameznim plačilom. Namesto tega je črv uporabljal le tri fiksne bitcoin naslove [32], kar je onemogočilo povezavo med plačili in posameznimi žrtvami ter s tem avtomatizirano dešifriranje datotek. To, skupaj z dejstvom, da se je ta izsiljevalska programska oprema širila kot črv, in ne targetirano, nakazuje na možnost, da je bil napad bolj ideološko kot finančno motiviran.

Napad je bil ustavljen samo nekaj ur po odkritju prvega primera, ko je raziskovalec v datoteki kriptočrva odkril domensko ime, ki se je uporabljalo za izklop; registracija te domene je ustavila nadaljnje aktivacije prvotne različice, vendar pa so se kmalu pojavile spremenjene različice brez tega stikala.

5.2 Mobilna zlonamerna aplikacija iz družine CherryBlos

Spodnji opis črpa informacije iz izvrstne in poglobljene analize ter vzvratnega inženirstva, ki sta nastala pod okriljem slovenskega SI-CERT [33] in sta predstavljena v okviru magistrske naloge [34] na Fakulteti za elektrotehniko.

CherryBlos je družina zlonamerne programske opreme za mobilne naprave z operacijskim sistemom Android. Razvita je bila z namenom kraje kriptovalut [34], [35], za kar uporablja dva mehanizma:

- krajo denarnic na osnovi semenske fraze (angl. seed phrase), ki jo izvede z optično prepoznavo znakov (angl. optical character recognition – OCR) na uporabnikovi galeriji fotografij;
- spreminjanje ciljnega naslova pri transakcijah s kriptovalutami, ki jih uporabnik izvede na svoji napravi.

Za dostavo programske opreme na napravo je bil CherryBlos skrit v več različnih aplikacij – podjetje TrendMicro [35] je identificiralo štiri, analiza [34] pa podrobneje obravnava eno od njih. Nekatere od identificiranih aplikacij (kot tudi več aplikacij iz sorodnih družin) so zaznali tudi v trgovini Google Play, preostale so se oglaševale prek neposrednih povezav na družbenih omrežjih. Operacijski sistem Android namreč dovoljuje tudi nameščanje nepreverjenih aplikacij, kar vsak uporabnik lahko vklopi sam [30].

Na tem mestu se najprej postavi vprašanje, kako je mogoče, da takšne zlonamerne aplikacije uspešno opravijo pregled pred objavo in dodaten pregled ob namestitvi, ki ga opravi storitev Google Play Protect. Odgovor se skriva v uporabi več prej omenjenih tehnik, kot sledi:

- Obfuskacija, ki otežuje analizo programa; CherryBlos uporablja napredno obfuskacijo s plačljivim pakirnikom Jiagu [31], zaradi česar je vsakršna podrobnejša analiza izjemno zamudna.
- Uporaba večfaznega postopka namestitve, kjer v trgovini objavljena aplikacija služi samo kot dostavljavka za zlonamerno aplikacijo; s tem je glavnina škodljivih funkcij prestavljena v šifriran tovor, ki v času pregleda ni sumljiv ali še ni na voljo.
- Uporaba drugih pomanjkljivosti v varnostnih mehanizmih; analiza [34] opisuje primere aplikacij, ki uporabljajo neveljaven ZIP-format za paket Android, ki onemogoči razpakiranje aplikacije s standardnim postopkom, hkrati pa zaradi posebnosti pri implementaciji razpakirnika na operacijskem sistemu Android dovoljuje namestitev.

Kljub velikim vložkom podjetij v varnost mobilnih platform in trgovin z aplikacijami lahko z uporabo naštetih (in sorodnih) tehnik zlonamerne aplikacije v določenih primerih uspešno zaobidejo varnostne mehanizme in se na platformi distribuirajo vse do odkritja, ko je aplikacija odstranjena.

Vseeno pa zlonamerna programska oprema zaradi zagona v restriktivnem okolju peskovnika (angl. sandbox) nima pravice dostopa do drugih delov operacijskega sistema. Za povzročitev škode si mora najprej ustrezno dvigniti privilegije. To je s tehničnega vidika relativno težko, saj bi bilo potrebno poznavanje večjih neodpravljenih ranljivosti v operacijskem sistemu. Vendar pa obstaja tudi mnogo lažji način – aplikacija lahko za privilegije enostavno prosi uporabnika.

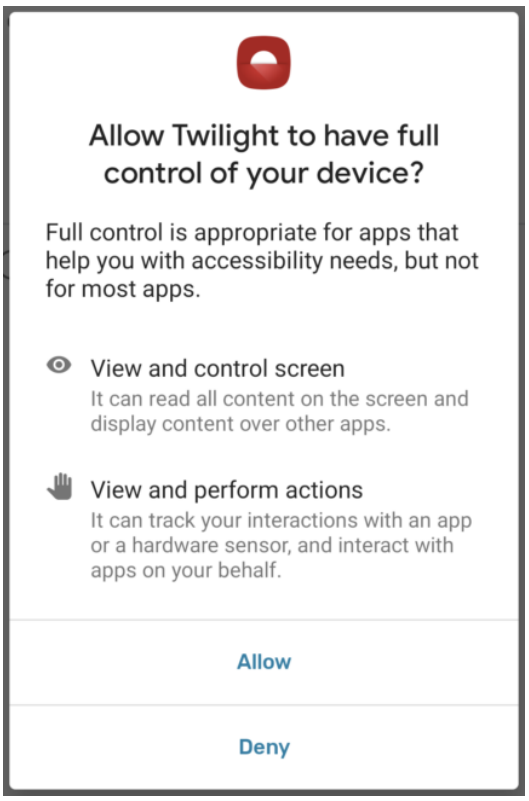
Na operacijskem sistemu Android (podobno velja tudi za iOS) je namreč na voljo funkcionalnost *dostopnosti* (angl. accessibility), ki aplikaciji omogoči programski dostop do vsebine zaslona ter ji da možnost vnosa besedila in dotikov, popoln nadzor navigacije in celo prebujanja telefona iz spanja in terminacije procesov. Slika 4 prikazuje takšen dialog za dodelitev pravic dostopnosti. Funkcija dostopnosti je namenjena predvsem ljudem s posebnimi potrebami in uporabi z bralniki zaslona ali posebnimi vhodnimi napravami. Ta funkcionalnost je zaradi svoje problematične narave strogo nadzorovana, in aplikacije, ki jo zahtevajo, so podvržene tudi poglobljenemu pregledu pred objavo na platformi Google Play. S premikom zahteve po dostopnosti v neobljavljeno aplikacijo (bodisi nameščene neposredno s spleta bodisi prek druge aplikacije – dostavljavca) pa se razvijalci izognejo problematičnemu pregledu.

Aplikacija ob pridobitvi pravic za funkcionalnost dostopnosti pridobi poln nadzor nad napravo. To je v primeru zlonamernih aplikacij pogosto ireverzibilno, saj se poleg izvajanja škodljivih funkcionalnosti aplikacija tudi aktivno bori proti odstranitvi teh pravic. Na primer:

- Ob odprtju nastavitev aplikacija sama zapre nastavitve in se vrne na domači zaslon telefona.
- Ob detekciji poziva za odstranitev aplikacije, ki ga prikaže varnostna storitev Google Play Protect, aplikacija sama zapre obvestilo in tako prekliče svojo odstranitev.
- Aplikacija preprečuje prehod telefona v stanje spanja, pri katerem bi operacijski sistem aplikacijo lahko ustavil.

Ves čas delovanja pa aplikacija izvaja tudi naslednje zlonamerne funkcionalnosti:

- Pregled galerije fotografij in zaslonskih posnetkov, na katerih naredi optično branje in detektira mnemonične fraze za odklepanje denarnice s kriptovalutami. Operacijski sistem Android sicer zahteva uporabniško potrditev za dostop do galerije fotografij, vendar pa lahko aplikacija z uporabo prej omenjene funkcije dostopnosti sama pritisne na gumb »Dovoli«.
- Spremlja vsebino zaslona, in če zazna transakcijo s kriptovalutami v priljubljeni aplikaciji Binance, zamenja ciljni naslov transakcije z naslovom prevarantov, potem pa z uporabo funkcij dostopnosti čez uporabniški vmesnik aplikacije prikaže lažno okno s pravilnim naslovom.
- Za večje število priljubljenih denarnic s kriptovalutami (BitKeep, Metamask, imToken, Trust wallet, Onto) pri odprtju prikaže lažen uporabniški vmesnik (t. i. aktivnost), ki posnema denarnico, ter zahteva vnos mnemonične fraze. Slednjo ob vnosu pošlje na strežnik.



Slika 4: Dialog za pravice dostopnosti popularne aplikacije Twlight [36]; ko uporabnik izbere »Allow«, aplikaciji dodeli možnost polnega nadzora nad napravo.

Iz zapisanega je razvidno, da je ključna ranljivost, ki jo ta aplikacija izkorišča, človeška – en pritisk na gumb »Dovoli« za dodelitev vseh pravic takšni aplikaciji je dovolj, da se ireverzibilno in z visokimi privilegiji namesti na napravo.

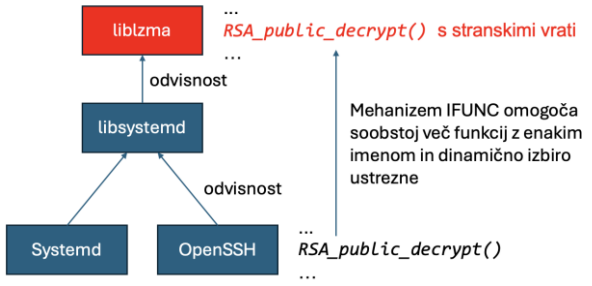
5.3 Kompromitiranje dobavne verige operacijskega sistema Linux

Kot že omenjeno, je dobavna veriga izjemno močan vektor vnosa zlonamerne kode v potencialno veliko množico sistemov. Kot študijo primera si oglejmo nedavni spodleteli poskus, ki je bil marca 2024 slučajno odkrit in uspešno ustavljen. Poskus je prek knjižnice *liblzma* targetiral širok nabor distribucij operacijskega sistema Linux, kaže pa na to, kako relativno malo truda – čeprav je napad precej sofisticiran – je potrebnega za potencialno velik izkupiček.

Linux je odprtokodni operacijski sistem, ki je zaradi svoje zanesljivosti, prilagodljivosti in brezplačnosti dosegel izjemno širok razmah. Danes po ocenah podjetij

poganja več kot 63 odstotkov oblačne infrastrukture [37], s čimer je postal tudi integralen del kritičnih infrastruktur, prisoten pa je tudi na več kot 72 odstotkih mobilnih naprav (Android) [38]. S takšnim tržnim deležem je postala dobavna veriga operacijskega sistema izjemno lukrativna tarča.

Kompleksen sistem, kakršen je Linux, je mogoče dobro obvladovati samo z modularnostjo. Eden od takšnih popularnih in široko uporabljanih modulov je že omenjena knjižnica *liblzma*, ki implementira kompresijski postopek, imenovan *XZ*. Slednjega je med letoma 2005 in 2008 razvil avtor Lasse Collin, ki še vedno vzdržuje to knjižnico. Postopek *XZ* je v preteklih 15 letih zaradi dobrega kompresijskega razmerja postal vseprisoten, drugi programi pa ga lahko enostavno uporabijo tako, da se povežejo s knjižnico *liblzma*, ki pride nameščena kot del operacijskega sistema. Knjižnico *liblzma* uporablja tudi programski paket *systemd* (sicer posredno, prek knjižnice *libsystemd*), ki skrbi za zagon in upravljanje operacijskega sistema Linux. Knjižnico *libsystemd* pa je zaradi komunikacije s *systemd* uporabljala tudi varna lupina *OpenSSH*, ki omogoča oddaljen terminalni dostop do operacijskega sistema. Razmerje prikazuje Slika 5.



Slika 5: Mehanizem vključitve zlonamerne kode v proces OpenSSH

Zaradi odprtokodne narave tako rekoč vseh komponent sistema lahko kdorkoli prispeva izboljšave zanje. Oktobra 2021 se je pojavil dotlej neznani kontributor s psevdonimom *Jia Tan*. Ta avtor je na e-poštni seznam projekta *XZ* poslal več legitimnih izboljšav, ki pa jih je skrbnik (Collin) v večji meri ignoriral. Ko se je nabralo več konstruktivnih predlogov, so drugi kontributorji začeli izvajati pritisk na skrbnika, naj jih upošteva. Ta je priznal, da je projekt dal na stranski tir zaradi duševne stiske.

Pritisk na skrbnika se je nato še stopnjeval z več lažnimi profili, s ciljem, da tudi Jia Tan pridobi status skrbnika, ki bo lahko sam dodajal svoje popravke v repozitorij. Septembra 2022, po manj kot enem letu aktivnosti, je Tan uradno postal eden od skrbnikov projekta, in je marca 2023 že sam izdal svojo prvo različico *XZ* v5.4.2.

Napad z zlonamerno kodo se je začel leto zatem, februarja 2024, ko je Tan v projekt dodal več binarnih testnih datotek za kompresijski postopek. Takšne datoteke so v kontekstu kompresorjev nekaj običajnega (izvajanje testov enot), v tem primeru pa so vsebovale kodo za stranska vrata, ki je bila obfuskirana in razporejena med dve od dodanih datotek. V enem od naslednjih korakov je Tan dodal tudi skripto za pripravo namestitvenega paketa, ki je med postopkom pakiranja iz binarnih datotek sestavila datoteko s škodljivo objektno kodo in jo uporabila pri procesu prevajanja; tako zlonamerna koda ni bila neposredno vidna nikjer v repozitoriju.

Temu je sledilo še več majhnih sprememb; ena izmed njih je bila onesposobitev varnostnega mehanizma, ki procesom omejuje privilegije (angl. landlocking). Ko je ta varnostni mehanizem aktiven, proces efektivno teče v peskovniku in ne more povzročiti škode. Mehanizem pa se v program doda šele med prevajanjem – vendar le, če je na voljo ustrezna podpora. Tan je v funkcijo za ugotavljanje podpore dodal nevpadljivo ločilo (piko), ki je pokvarilo izvajanje celotne funkcije, in posledično se med prevajanjem varnostni mehanizem ni dodal.

Mehanizem delovanja ranljivosti je naslednji: OpenSSH za zagotovitev večje varnosti podpira avtentikacijo z javnim ključem (angl. pubkey), za kar proces uporablja lastno funkcijo `RSA_public_decrypt()`, s katero preverja digitalni podpis odjemalca. Tanova koda, ki je v proces OpenSSH vključena samo posredno (*OpenSSH* uporablja *libsystemd*, ki uporablja *liblzma*), z mehanizmom indirektne funkcije (t. i. GNU_IFUNC) registrira novo – škodljivo – različico te funkcije, ki nadomesti prvotno (Slika 5). Posledično lahko tako napadalec že med postopkom avtentikacije z ustrezno oblikovanim tovorom na strežniku sproži poganjanje svojih inštrukcij z visokimi privilegiji.

Zlonamerna različica programa *XZ* v5.6.1 je bila uspešno vključena v testne različice več distribucij, med drugim Debian, Fedora, openSuse, Arch in Alpine; če ne bi bila odkrita, bi s tem dosegla večinski tržni delež distribucij Linux. Neposredni učinek zlonamerne kode bi bil zagotovitev stranskih vrat v tako rekoč vsak strežnik Linux, ki uporablja varno lupino OpenSSH.

Zlonamerno kodo je odkril razvijalec na Microsoftu, ki je opazil, da redni testi njegove kode na novi različici sistema Linux trajajo dlje in bolj obremenjujejo procesor. Pri podrobnejšem pregledu je odkril, da je vir težave knjižnica *liblzma*. Po odkritju so ranljivo kodo v nekaj dneh odstranile vse distribucije, projekt OpenSSH pa je iz varnostnih razlogov odstranil tudi povezavo s knjižnico *libsystemd* in za komunikacijo s *systemd* eksplicitno implementiral le nujno potrebne metode.

Strokovnjaki za kibernetsko varnost menijo, da bi v primeru uspeha to postala ena od najučinkovitejših stranskih vrat, saj bi avtorjem dala dostop do kateregakoli od več sto milijonov strežnikov Linux z aktivnim procesom OpenSSH [39].

Po načinu delovanja je ta napad primerljiv z drugimi zloglasnimi napadi na osnovi dobavne verige (HeartBleed, Log4j), razlikuje pa se po nameri, saj pri *XZ* ne gre za hrošča, temveč za namerno dejanje. Seveda pa

nimamo podatka o tem, koliko podobnih primerov, bodisi namernih bodisi nenamernih, se v kodi tega in preostalih operacijskih sistemov še skriva.

6 ZAKLJUČEK

Zlonamerna programska oprema se je iz preprostih eksperimentov razvila v eno najbolj kompleksnih in škodljivih orodij kibernetskega kriminala, ki ima daljnosežne posledice za posameznike, organizacije in kritično infrastrukturo. Z napredkom tehnologije so se razvile sofisticirane metode širjenja, vzdrževanja prisotnosti in prikrivanja zlonamernosti, naša vedno večja odvisnost od informacijsko-komunikacijskih tehnologij pa še povečuje ranljivo površino in količino sistemov, kjer se takšna programska oprema lahko skrije. Primeri, kot je opisan *CherryBlos*, kažejo, da zlonamerna programska oprema ni več omejena le na klasične računalniške sisteme, ampak meri tudi na mobilne platforme in digitalne finančne sisteme. Prav tako incident z *liblzma* opozarja na izjemno ranljivost dobavne verige, kar odpira pomembna vprašanja o zaupanju v odprtokodno programsko opremo ter kaže na potrebo po strožjih zagotovilih in mehanizmih vpogleda.

V prihodnosti bo razvoj umetne inteligence in avtomatizacije igral ključno vlogo pri oblikovanju novih varnostnih izzivov in rešitev. Po eni strani lahko pričakujemo vedno več uporabe takšnih tehnik za generiranje prilagojenega zlonamernega programja, ki se bo učinkoviteje izogibalo detekciji; po drugi strani pa bodo inteligentne rešitve pomagale tudi pri obrambi. Predvidevamo lahko, da se bo obseg napadov zaradi digitalizacije, vedno večjega števila povezanih naprav in vedno večje količine programske opreme le še povečeval, kar bo zahtevalo večplastne varnostne strategije, strožje regulacije in boljše ozaveščanje uporabnikov. Brez ustreznih varnostnih standardov in proaktivne zaščite se tveganja ne bodo le ohranila, ampak bodo postala še bolj nepredvidljiva in z možnimi sistemskimi posledicami na globalni ravni.

ZAHVALA

Raziskavo so delno podprli Javna agencija za raziskovalno dejavnost Republike Slovenije, Urad vlade Republike Slovenije za informacijsko varnost (URSIV) in Ministrstvo za obrambo Republike Slovenije prek projekta »Kibernetska varnost obrambnih sistemov in kritičnih infrastruktur« (V2-2378) ter raziskovalnega programa »Decentralizirane rešitve za digitalizacijo industrije ter pametnih mest in skupnosti« (P2-045).

LITERATURA

[1] A. Johnson, K. Dempsey, R. Ross, S. Gupta, in D. Bailey, „Guide for security-focused configuration management of information systems“, National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800–128, okt. 2019. doi: 10.6028/NIST.SP.800–128.

[2] S. Morgan, „Global Ransomware Damage Costs Predicted To Exceed $265 Billion By 2031“, Cybercrime Magazine. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/

[3] „AV-ATLAS - Malware & PUA“, AV-ATLAS - Malware & PUA. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://portal.av-atlas.org/malware

[4] Apple, Inc., „A Threat Analysis of Sideloading“.

[5] J. Parikka, *Digital Contagions: A Media Archaeology of Computer Viruses*. Peter Lang, 2007.

[6] S. Singh, P. K. Khare, in P. Mor, „Malware Detection and Removal Techniques“, *International Journal of Electronics and Computer Science Engineering*, let. 2, št. 1, str. 273–280, 2013.

[7] I. Naseer, „The crowdstrike incident: Analysis and unveiling the intricacies of modern cybersecurity breaches“, *International Journal of Information Management*, let. Vol. 11, str. 4, okt. 2024.

[8] G. Rodríguez-Pérez, G. Robles, A. Serebrenik, A. Zaidman, D. M. Germán, in J. M. Gonzalez-Barahona, „How bugs are born: a model to identify how bugs are introduced in software components“, *Empir Software Eng*, let. 25, št. 2, str. 1294–1340, mar. 2020, doi: 10.1007/s10664-019-09781-y.

[9] A. A. Joshua, „Fraud Theories and Global Cybercrime During the COVID-19 Pandemic: the Extended S.C.C.O.R.E model“, *International Journal of Research Science and Management*, let. 10, št. 10, Art. št. 10, okt. 2023.

[10] Verizon, „2024 Data Breach Investigations Report“. Pridobljeno: 27. februar 2025. [Na spletu]. Dostopno na: https://www.verizon.com/business/resources/reports/2024-dbir-executive-summary.pdf

[11] „Malware Classification Based on Dynamic Behavior“, predstavljeno na 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), sep. 2016. doi: 10.1109/SYNASC.2016.057.

[12] D. V. Bontchev, „Current Status of the CARO Malware Naming Scheme“, predstavljeno na 15th Virus Bulletin Int. Conf., Dublin, Ireland, okt. 2005.

[13] O. Or-Meir, N. Nissim, Y. Elovici, in L. Rokach, „Dynamic Malware Analysis in the Modern Era—A State of the Art Survey“, *ACM Comput. Surv.*, let. 52, št. 5, str. 1–48, sep. 2020, doi: 10.1145/3329786.

[14] „MITRE ATT&CK®“. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://attack.mitre.org/

[15] M. Souppaya in K. Scarfone, „Guide to Malware Incident Prevention and Handling for Desktops and Laptops“, National Institute of Standards and Technology, NIST Special Publication (SP) 800-83 Rev. 1, jul. 2013. doi: 10.6028/NIST.SP.800-83r1.

[16] S. Razaulla *idr.*, „The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions“, *IEEE Access*, let. 11, str. 40698–40723, 2023, doi: 10.1109/ACCESS.2023.3268535.

[17] E. Cartwright, J. Hernandez Castro, in A. Cartwright, „To pay or not: game theoretic models of ransomware“, *Journal of Cybersecurity*, let. 5, št. 1, str. 1–12, jan. 2019, doi: 10.1093/cybsec/tyz009.

[18] C. Everett, „Ransomware: to pay or not to pay?“, *Computer Fraud & Security*, let. 2016, št. 4, str. 8–12, apr. 2016, doi: 10.1016/S1361-3723(16)30036-7.

[19] S. Khandekar, „Big Brother Browser: Who’s Watching You Online?“, 2023.

[20] D. Harkin, A. Molnar, in E. Vowles, „The commodification of mobile phone surveillance: An analysis of the consumer spyware industry“, *Crime, Media, Culture*, let. 16, št. 1, str. 33–60, mar. 2020, doi: 10.1177/1741659018820562.

[21] J. Joy, A. John, in J. Joy, „Rootkit Detection Mechanism: A Survey“, v *Advances in Parallel Distributed Computing*, let. 203, D. Nagamalai, E. Renault, in M. Dhanuskodi, Ur., v Communications in Computer and Information Science, vol. 203. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, str. 366–374. doi: 10.1007/978-3-642-24037-9_36.

[22] S. Kim, J. Park, K. Lee, I. You, in K. Yim, „A Brief Survey on Rootkit Techniques in Malicious Codes.“, *J. Internet Serv. Inf. Secur.*, let. 2, št. 3/4, str. 134–147, 2012.

[23] M. Feily, A. Shahrestani, in S. Ramadass, „A Survey of Botnet and Botnet Detection“, v *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, jun. 2009, str. 268–273. doi: 10.1109/SECURWARE.2009.48.

[24] P. Szor, *The Art of Computer Virus Research and Defense*. Pearson Education, 2005.

[25] D. Harley, L. Myers, in E. Willems, „Test Files and Product Evaluation: the Case for and against Malware Simulation“, v *AVAR Conference*, 2011.

[26] X. Li, P. K. K. Loh, in F. Tan, „Mechanisms of Polymorphic and Metamorphic Viruses“, v *2011 European Intelligence and Security Informatics Conference*, sep. 2011, str. 149–154. doi: 10.1109/EISIC.2011.77.

[27] C. Zhang, S. Zhou, in B. M. Chain, „Hybrid Epidemics—A Case Study on Computer Worm Conficker“, *PLoS ONE*, let. 10, št. 5, str. e0127478, maj 2015, doi: 10.1371/journal.pone.0127478.

[28] Sudhakar in S. Kumar, „An emerging threat Fileless malware: a survey and research challenges“, *Cybersecur*, let. 3, št. 1, str. 1, dec. 2020, doi: 10.1186/s42400-019-0043-x.

[29] V. Hassija, V. Chamola, V. Gupta, S. Jain, in N. Guizani, „A Survey on Supply Chain Security: Application Areas, Security Threats, and Solution Architectures“, *IEEE Internet of Things Journal*, let. 8, št. 8, str. 6222–6246, apr. 2021, doi: 10.1109/JIOT.2020.3025775.

[30] S. Garg in N. Baliyan, „Comparative analysis of Android and iOS from security viewpoint“, *Computer Science Review*, let. 40, str. 100372, maj 2021, doi: 10.1016/j.cosrev.2021.100372.

[31] „360JIAGU“. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://jiagu.360.com/#/global/index

[32] M. Akbanov, V. G. Vassilakis, in M. D. Logothetis, „WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms“, *Journal of Telecommunications and Information Technology*, št. 1, str. 113–124, 2019.

[33] „Nacionalni odzivni center za kibernetsko varnost“, SI CERT. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://www.cert.si/

[34] U. Vidergar, „Analiza zlonamernih mobilnih aplikacij s tehnikami vzvratnega inženirstva“, thesis, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2024. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://repozitorij.uni-lj.si/IzpisGradiva.php?id=159629

[35] „Related CherryBlos and FakeTrade Android Malware Involved in Scam Campaigns“, Trend Micro. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://www.trendmicro.com/en_us/research/23/g/cherryblos-and-faketrade-android-malware-involved-in-scam-campai.html

[36] „Is Twilight’s Accessibility service a threat to my privacy? – Twilight“. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://twilight.urbandroid.org/is-twilights-accessibility-service-a-thread-to-my-privacy/

[37] Fortune Business Insights, „Server Operating System Market Share Analysis“. Pridobljeno: 28. februar 2025. [Na spletu]. Dostopno na: https://www.fortunebusinessinsights.com/server-operating-system-market-106601

[38] „Mobile OS market share worldwide 2009-2024“, Statista. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/

[39] K. Roose, „Did One Guy Just Stop a Huge Cyberattack?“, *The New York Times*, 3. april 2024. Pridobljeno: 26. februar 2025. [Na spletu]. Dostopno na: https://www.nytimes.com/2024/04/03/technology/prevent-cyberattack-linux.html

Urban Sedlar je leta 2010 doktoriral na Fakulteti za elektrotehniko Univerze v Ljubljani, kjer je trenutno zaposlen kot izredni profesor in vodja projektov v Laboratoriju za telekomunikacije. Raziskovalno je aktiven na področju modeliranja napadalcev in ocene kibernetskih groženj z uporabo kibernetskih pasti. Sodeloval je v več raziskovalnih in razvojnih projektih Evropske komisije in nacionalnih projektih na temo kibernetske varnosti, računalništva v oblaku, interneta stvari in sistemov za odzivanje na izredne razmere.